

ROBOTICS

Basic Electronics

Electricity, components, circuits, motors and microcontrollers: the foundation for robotics.

11 Chapters

Philip Osztromok

Generated with Claude

BASIC ELECTRONICS

Contents

Chapter 1	Electricity Basics	2
Chapter 2	Circuits and Measurement	13
Chapter 3	Resistors and Dividers	25
Chapter 4	Capacitors and Inductors	41
Chapter 5	Diodes and LEDs	59
Chapter 6	Transistors as Switches	80
Chapter 7	Analogue Signals and Op-Amps	102
Chapter 8	Digital Logic Basics	128
Chapter 9	Motors, Relays and Power	161
Chapter 10	Microcontrollers in Circuits	194
Chapter 11	Choosing a Board and Going Further	215

Basic Electronics

Course 1 · Chapter 1 · Electricity Basics

Every robot, from a line-following buggy to a walking quadruped, is a collection of circuits: sensors that turn the world into voltages, logic that decides, and drivers that push current into motors. This course builds that picture one component at a time. This first chapter gives you the four ideas everything else rests on (charge, voltage, current and resistance), the one equation that ties them together (Ohm's law), and ends with a circuit you can build in ten minutes: a battery, a resistor and an LED.

No background is assumed. You will need a calculator and, ideally, a cheap multimeter, but you can do every exercise in this chapter in a free browser simulator if you have no hardware yet.

1. The four quantities

Charge and current

Matter contains charged particles. In a metal wire, some electrons are free to drift between atoms. **Current** is the rate at which charge flows past a point. It is measured in **amperes** (A), and one ampere is one **coulomb** of charge per second:

```
I = Q / t        current (A) = charge (C) / time (s)
Example: 0.5 C passes in 2 s  ->  I = 0.5 / 2 = 0.25 A = 250 mA
```

One coulomb is about 6.24×10^{18} electrons, so even a tiny current is an enormous number of moving electrons.

Conventional current

Early scientists guessed that current flowed from the positive terminal of a battery to the negative. We now know that in wires it is electrons moving the other way, but the convention stuck. **Conventional current** flows from + to −, and every schematic, datasheet and arrow symbol (such as the one inside a diode) uses it. The maths comes out identical, so use it and ignore the electron direction.

Voltage

Voltage (potential difference) is the energy given to each coulomb of charge as it moves between two points. It is measured in **volts** (V): one volt is one joule per coulomb.

```
V = E / Q        voltage (V) = energy (J) / charge (C)
```

Voltage is always measured *between two points*. A battery marked 9 V has 9 V between its terminals; there is no such thing as "the voltage at one point" without a reference. In circuits that reference is called **ground** (0 V), usually the battery's negative terminal.

Resistance

Resistance measures how much a component opposes current. It is measured in **ohms** (Ω). A thick copper wire has almost none, rubber has almost infinite resistance, and a *resistor* is a part built to have an exact value.

> **THE WATER ANALOGY**
>
> Think of a water system. Voltage is the water pressure, current is the flow rate, and resistance is a narrow pipe. More pressure gives more flow; a narrower pipe gives less. The analogy breaks down in places, but it is the best first mental model.

Units and prefixes

Electronics constantly uses very small and very large numbers, so you must be fluent in the prefixes:

PREFIX	SYMBOL	MULTIPLY BY	EXAMPLE
mega	M	1,000,000	1 MΩ = 1,000,000 Ω
kilo	k	1,000	4.7 kΩ = 4,700 Ω
(none)		1	9 V, 220 Ω, 2 A
milli	m	0.001	20 mA = 0.020 A
micro	µ	0.000001	150 µA = 0.000150 A

The most common beginner error is forgetting to convert. If you put 10 mA into a formula, write 0.010 A, or keep everything in volts, kilohms and milliamps (the three are consistent: V = kΩ × mA).

2. Ohm's law

In 1827 Georg Ohm found that for most conductors the current is proportional to the voltage across them. Doubling the voltage doubles the current. The constant of proportionality is 1/R, so:

```
I = V / R        current = voltage / resistance
V = I × R      voltage = current x resistance
R = V / I        resistance = voltage / current

Memory triangle (cover the one you want):
        V
      -----
      I | R
```

Worked example. A 1 kΩ resistor has 5 V across it. Then I = 5 / 1000 = 0.005 A = **5 mA**. Put 10 V across the same resistor and the current doubles to 10 mA.

> **OHM'S LAW IS NOT TRUE FOR EVERYTHING**
>
> It holds for resistors and wires. It does *not* hold for LEDs, diodes and transistors, whose current rises far faster than the voltage. That is exactly why an LED needs a resistor in series: the resistor does the limiting, and Ohm's law applies to the resistor alone. You will see how diodes behave in Basic Electronics 5.

3. Power

Power is the rate at which energy is converted, measured in **watts** (W). Since voltage is energy per coulomb and current is coulombs per second, multiplying them gives joules per second:

```
P = V × I

Substitute V = I x R:   P = I x I x R  =  I^2 x R
Substitute I = V / R:   P = V x V / R  =  V^2 / R
```

All three forms are the same law; choose whichever uses the two numbers you already know.

Worked example: why power ratings matter. Suppose you wire a 100 Ω resistor straight across a 9 V battery. I = 9 / 100 = 0.09 A (90 mA). P = V × I = 9 × 0.09 = **0.81 W**. A common resistor is rated 0.25 W (1/4 W). It is dissipating more than three times its rating, so it will get too hot to touch and can burn. Always check P against the rating, and leave margin: aim for at most half the rating.

4. DC and AC

Direct current (DC) flows one way at a steady value: batteries, USB chargers and the supply rails inside every robot. **Alternating current (AC)** reverses direction continuously. UK mains is AC at 230 V RMS and 50 Hz, so the voltage swings 50 times per second between about +325 V and −325 V.

The "230 V" is an *RMS* value, defined as the DC voltage that would heat a resistor equally. For a sine wave, $V_{peak} = V_{RMS} \times \sqrt{2}$, giving 230 × 1.414 ≈ 325 V. Robot electronics works almost entirely in DC; AC appears in this course in power supplies (rectifiers, Basic Electronics 5) and in signals such as audio and motor drive waveforms.

	DC	AC
Direction	Constant	Reverses periodically
Typical sources	Batteries, USB, solar cells, regulators	Mains sockets, generators, signal sources
Described by	A single value (e.g. 5 V)	RMS (or peak) value and frequency
In this course	Nearly everything	Filters, rectifiers, signals

5. Safety: what is safe to touch

The danger to your body is current through it, but current is driven by voltage, so voltage is the practical guide. Dry skin has a resistance of tens of kilohms or more, so low voltages cannot push a dangerous current through you.

SOURCE	VOLTAGE	RISK
AA/AAA cells, coin cells, USB, Arduino/Pico/Pi boards	1.5 to 5 V	Safe to touch. The risk is to the components, and to wires if you short a battery.

SOURCE	VOLTAGE	RISK
9 V PP3 battery, 12 V lead-acid, hobby 3S LiPo (11.1 V)	9 to 12 V	Safe to touch, but *can deliver large currents* if shorted: hot wires, burns. LiPo cells can catch fire.
Mains outlets and anything plugged into them	230 V AC (UK)	**Can kill.** Do not open, modify or probe mains equipment, and do not build mains circuits on a breadboard.

RULES FOR THIS ENTIRE COURSE

- Only build from batteries or a low-voltage (5 V to 12 V) supply. Never from the mains.
- Never connect a battery's + terminal directly to its − terminal. A short circuit makes the wire and battery hot within seconds.
- Disconnect the battery before changing the wiring.
- Do not charge or puncture LiPo or lithium cells without a proper charger and a fireproof bag. This chapter uses neither.
- Wear eye protection when trimming component legs: the offcuts fly off at speed.
- Hot resistors and soldering irons burn. Check with the back of a finger first.

6. First circuit: battery, resistor, LED

An LED (light-emitting diode) lights when current flows through it in one direction only. It has a roughly fixed voltage drop across it when lit (its **forward voltage**, V_F) and no resistance worth speaking of, so connected straight to a battery it would try to draw as much current as the battery could give, and burn out in a moment. A series resistor sets the current.

Parts list

PART	VALUE	NOTES
Battery	9 V (PP3) with snap clip	Or a 3 x AA pack (4.5 V) with a different resistor: see Challenge 1
R1	820 Ω, 1/4 W	Colour code grey-red-brown (an E12 series value)
D1	Standard 5 mm red LED	V_F about 2.0 V at 10 mA; absolute maximum about 20 to 30 mA
Breadboard	Half size or larger	Plus 3 jumper wires
Multimeter	Any digital	Optional but strongly recommended

Choosing R1: worked calculation

Aim for a comfortable 10 mA: bright for an indicator LED and safely below its limit. In a series loop, the battery voltage is shared between the resistor and the LED, so the resistor gets whatever the LED does not:

```
Step 1: voltage across the resistor
V_R = V_supply - V_F = 9.0 - 2.0 = 7.0 V

Step 2: Ohm's law for the resistor, aiming for I = 10 mA
R = V_R / I = 7.0 / 0.010 = 700 ohm

Step 3: pick a real value. Resistors come in the E12 series:
10 12 15 18 22 27 33 39 47 56 68 82 (x 1, 10, 100, 1k ...).
700 ohm is not E12; the neighbours are 680 and 820.
Choose the higher (820) to stay on the safe side.

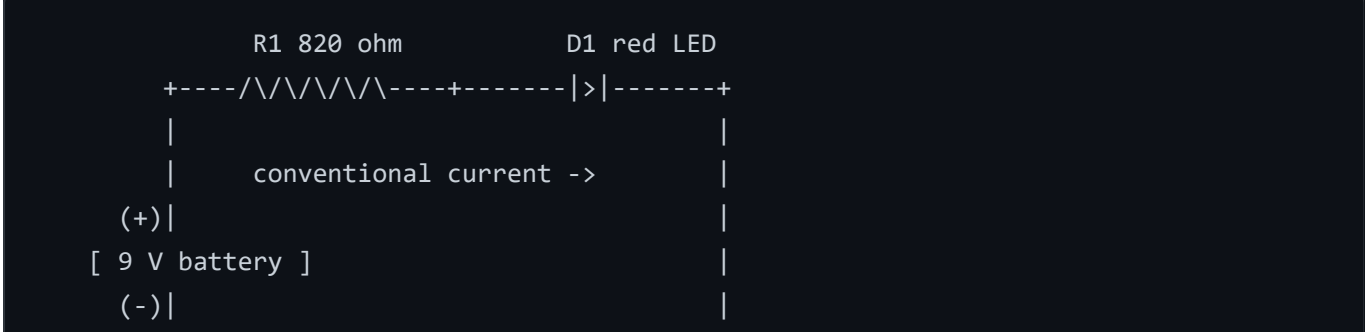
Step 4: actual current with 820 ohm
I = 7.0 / 820 = 0.00854 A = 8.5 mA

Step 5: power check
P_R   = V_R x I = 7.0 x 0.00854 = 0.060 W = 60 mW   (rating 250 mW: fine)
P_LED = V_F x I = 2.0 x 0.00854 = 0.017 W = 17 mW
P_bat = 9.0 x 0.00854 = 0.077 W = 77 mW             (= 60 + 17: energy is conserved)
```

With 680 Ω the current would be 10.3 mA, also fine; 820 Ω just gives a little more margin and the brightness difference is barely visible. Round the resistor *up* when in doubt: a dimmer LED is harmless, a hot one is not.

Schematic

A schematic shows how parts connect, not where they sit physically. The resistor is the zigzag (or a rectangle); the LED is a triangle pointing in the direction of conventional current, with a bar at the cathode.



```
        |                                   |
        +-----------------------------------+
                        0 V (ground)

  Current path: battery (+) -> R1 -> LED anode -> LED cathode -> battery (-)
  Measured voltages: across battery 9 V = across R1 7 V + across LED 2 V
```

Breadboard version

A breadboard has short rows of five connected holes (a to e connected together, f to j connected together, with a gap down the middle) and long power rails down the sides. Basic Electronics 2 covers the layout in detail; for now follow this recipe, using one numbered column of the board for each connection.

1. Disconnect the battery. Place the LED with its **longer leg (anode, +)** in column 10 and its shorter leg (cathode, next to the flat edge of the plastic rim) in column 12, both in rows a to e.
2. Place R1 with one leg in column 5 and the other in column 10, so it shares column 10 with the LED anode.
3. Run a jumper wire from the LED's cathode column (12) to the board's − rail (blue line).
4. Run a jumper from column 5 (R1's free end) to the + rail (red line).
5. Clip the battery on: red lead to the + rail, black lead to the − rail. The LED lights.

> **LED DOES NOT LIGHT?**
>
> Check, in order: the LED is the right way round (swap it; 9 V through 820 Ω will not harm it), both rails are connected to the right battery wires, nothing bridges two columns accidentally, and the battery is fresh.

Simulator version (no hardware needed)

Falstad Circuit Simulator (falstad.com/circuit, runs in the browser):

1. Right-click the canvas and add a *Resistor* (Passive Components), an *LED* (Active Components, under Diode) and a *DC Voltage Source* (Inputs and Sources).
2. Wire the three in a loop, battery to resistor to LED to battery, with the LED's anode (the triangle's flat base) towards the resistor.
3. Double-click the voltage source and set it to 9 V; double-click the resistor and set it to 820 Ω.

4. Hover over or click a component to see its voltage and current. The moving dots show current direction (+ to −).

Tinkercad Circuits (tinkercad.com, free account): create a new circuit and drag in a 9 V battery, a resistor, a red LED and a multimeter. Set the resistor to 820 Ω in its panel and wire the same loop. Click *Start Simulation*. Place the multimeter across (in parallel with) a part to read volts, or in series to read current, and select the mode in its panel.

Expected meter readings

MEASUREMENT	CALCULATED	REAL CIRCUIT	SIMULATOR
Battery terminals (DC V)	9.0 V	9.3 to 9.6 V fresh; 8 V or less when tired	The value you set
Across R1 (DC V)	7.0 V	7.2 to 7.6 V	About 7.0 V (depends on the LED model)
Across LED (DC V)	2.0 V	1.8 to 2.1 V (red)	About 1.8 to 2.1 V
Loop current (DC mA, meter in series)	8.5 mA	About 8 to 9 mA	About 8 to 9 mA

Expect real values to differ by a few percent: the battery voltage is not exactly 9 V, E12 resistors are typically ±5% (or ±1% for metal-film types), and each LED's V_F varies. What matters is that the two voltages **add up to the battery voltage** (7.4 + 2.0 = 9.4 V, say) and that R1's voltage divided by 820 gives the measured current. That is Kirchhoff's voltage law, covered in Basic Electronics 2.

> **MEASURING CURRENT: THE METER GOES IN SERIES**
>
> To measure current, break the loop and put the meter's probes in the gap, with the red probe in the mA or A socket and the dial on DC current. Never put an ammeter directly across the battery or another voltage source: that is a short circuit through the meter and will blow its fuse. For voltages, probe in parallel with the part and use the V dial. Full multimeter technique is the subject of Basic Electronics 2.

7. Series and parallel resistance

There are two ways to connect resistors, and the rules for them will be used constantly.

Series: one path, same current

The same current passes through each resistor, and the voltages across them add up to the supply. Using V = IR for each:

```
V = V1 + V2 = I x R1 + I x R2 = I x (R1 + R2)
R_series = R1 + R2 (+ R3 ...)

Example: 1 k + 2.2 k = 3.2 k ohm
```

Parallel: several paths, same voltage

The same voltage is across each branch, and the branch currents add. Using I = V/R for each:

```
I = I1 + I2 = V / R1 + V / R2 = V x (1/R1 + 1/R2)
1 / R_parallel = 1/R1 + 1/R2 (+ 1/R3 ...)

Two resistors only (product over sum):
R_parallel = (R1 x R2) / (R1 + R2)

Example: 330 || 470 = (330 x 470) / (330 + 470)
                    = 155,100 / 800 = 193.9 ohm
Two equal resistors: R / 2.  1 k || 1 k = 500 ohm.
```

Adding a branch in parallel always *lowers* the total resistance (there are more paths for current), and the result is always smaller than the smallest branch (193.9 Ω is below 330 Ω). That is a quick sanity check on any calculation. Voltage dividers (series pairs used to scale a voltage) are the subject of Basic Electronics 3.

	SERIES	PARALLEL
Current	Same through each	Splits between branches
Voltage	Splits between parts	Same across each branch
Total resistance	Sum (larger than any one)	Reciprocal sum (smaller than any one)
If one part fails open	Everything stops	Other branches keep working

8. Challenges

CHALLENGE 1: SIZE A RESISTOR

A green LED (V_F about 2.1 V, maximum 20 mA) is to be powered from 5 V (a USB supply). Calculate the resistor needed for about 10 mA, pick the nearest safe E12 value, then work out the actual current and the power dissipated in the resistor. Repeat with the 4.5 V (3 x AA) pack and decide whether 220 Ω is acceptable.

View solution

CHALLENGE 2: SERIES AND PARALLEL

A 12 V supply is connected across a 1 kΩ resistor and a 2.2 kΩ resistor in series. Find the current, the voltage across each, and the power in each. Then rewire them in parallel across the same 12 V and find each branch current, the total current, and the equivalent resistance. Check that the answers satisfy the sanity rules in section 7.

View solution

CHALLENGE 3: BUILD IT AND MEASURE IT

Build the battery-resistor-LED circuit (breadboard or simulator). Record the battery voltage, the voltage across R1, the voltage across the LED and the loop current, and compare them with the table in section 6. Then swap R1 for 220 Ω and predict, before you do it, the current and the resistor power; decide whether that is safe for the LED and the resistor.

View solution

QUICK REFERENCE

- **Current:** $I = Q / t$ (amperes); conventional current flows + to −
- **Voltage:** $V = E / Q$ (volts); always measured between two points
- **Ohm's law:** $V = I \times R$, so $I = V / R$ and $R = V / I$
- **Power:** $P = V \times I = I^2R = V^2/R$; keep P below half the rating
- **Series:** $R = R1 + R2$, same current. **Parallel:** $R = R1R2 / (R1 + R2)$, same voltage
- **LED resistor:** $R = (V_{supply} - V_F) / I$, rounded up to the next E12 value

- **Safety:** low-voltage batteries only; never short a battery; never touch mains

Next, Basic Electronics 2 turns to circuits and measurement: Kirchhoff's laws, the breadboard layout in full, and how to use the multimeter properly to check what you have built.

Basic Electronics

Course 1 · Chapter 2 · Circuits and Measurement

Chapter 1 gave you Ohm's law and the series and parallel rules for a single loop. Real circuits are networks: several loops sharing parts, with voltages and currents you cannot see until you measure them. This chapter gives you three tools for that: the conventions of the **schematic** (how to read and draw a circuit), **Kirchhoff's two laws** (the bookkeeping rules that every circuit obeys), and the **multimeter** (how to check what is really happening, and how the meter itself can mislead you). It also explains the breadboard layout properly, so the circuits in later chapters go together on the first attempt.

1. Reading a schematic

A schematic is a map of connections, not a picture of the board. Part positions on the page mean nothing; only which terminals are joined matters. By convention, the highest voltage (the + supply) is drawn at the top, ground at the bottom, and signals flow left to right.

```
Common symbols (ASCII versions)

 Resistor            ----/\/\/\----         or   ----[____]----
 Battery / cell       ---|i---                   (long line = +, short line = -)
 DC supply           (+) o----                  (labelled +5V, +9V, VCC ...)
 Ground (0 V)         ----+
                          |
                         ---   (three bars, getting shorter)
 LED                  ----|>|----              (current flows toward the bar)
 Switch               ----o  o----              (open) / ----o/ o---- (drawn open)
 Capacitor            ----| |----              (see Basic Electronics 4)

 Wires:
   crossing, NOT connected        crossing WITH a dot = connected
          |                              |
       ---+---   (avoid this)         ---(*)---
          |                              |
   A junction is always a dot. Prefer "T" junctions, which are unambiguous.
```

Each part also has a **reference designator**: a letter plus number. R is a resistor, C a capacitor, D a diode or LED, Q a transistor, U an integrated circuit, S or SW a switch, B or BT a battery. R1, R2 and R3 are three different resistors; the values are written next to them (`1k`, `2.2k`, `470`). A value such as `4k7` means 4.7 kΩ: the letter replaces the decimal point, so it cannot be lost when a drawing is photocopied.

Nodes

A **node** is any set of points joined by wire. Everything on a node is at the same voltage, because wire has (almost) no resistance. This is the key reading skill: trace each wire until it reaches a part, and mark every stretch that is one node. In the circuit below there are only *three* nodes: the + rail, node X and ground, even though the drawing has many wire segments.

2. Kirchhoff's laws

Ohm's law describes one part. Kirchhoff's laws (Gustav Kirchhoff, 1845) describe how parts connect, and together with Ohm's law they solve every resistor network. Both are simple statements of conservation.

Kirchhoff's current law (KCL): what flows in must flow out

Charge is not created or stored at a junction (a wire holds no charge, to the precision we care about), so the total current into any node equals the total current out of it.

```
KCL:  sum of currents into a node = sum of currents out of it

            I1 = 5 mA -->   (node)   --> I3 = ?
                              |
            I2 = 2 mA -->-----+

   Into the node: 5 + 2 = 7 mA.   So I3 = 7 mA.

Parallel rule from KCL: I = I1 + I2 at the split, and with V the same
across both branches, I = V/R1 + V/R2, giving 1/Rp = 1/R1 + 1/R2 as in Chapter 1.
```

Kirchhoff's voltage law (KVL): round a loop, the voltages sum to zero

Voltage is energy per coulomb. A charge that goes round a closed loop returns to where it started, so the energy it gained from the sources equals the energy it gave up in the parts. In

practice: **the rises equal the drops**.

```
KVL:  going round any closed loop, (rises) = (drops)

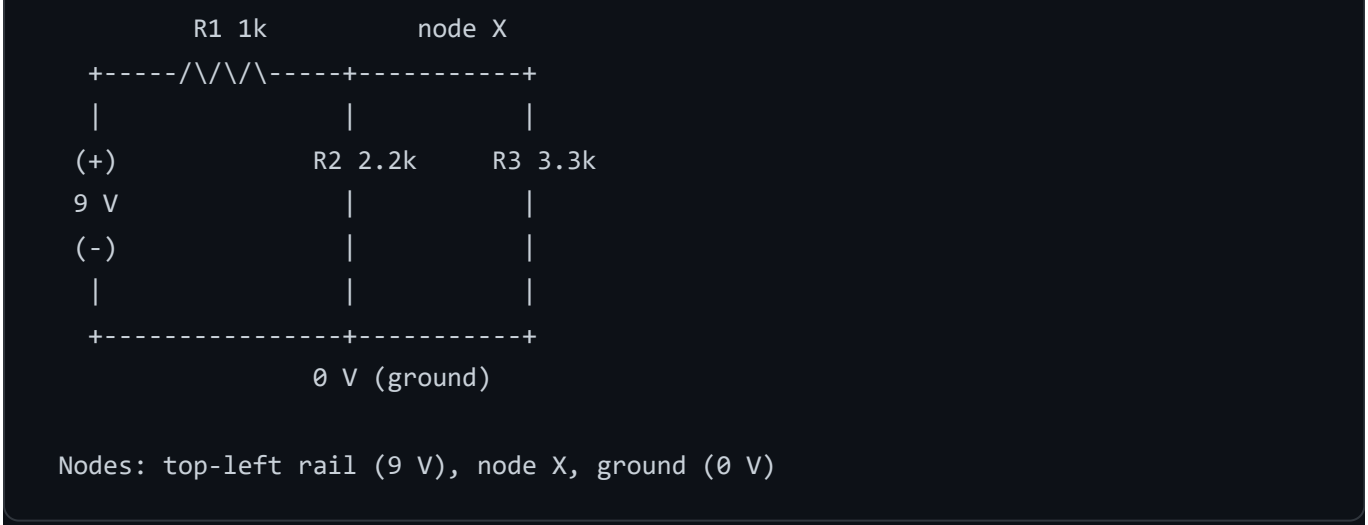
  Chapter 1 loop:  9 V battery  =  V(R1) + V(LED)  =  7 V + 2 V

  Series rule from KVL: V = I x R1 + I x R2 = I x (R1 + R2).
```

> **SIGN CONVENTION IN ONE SENTENCE**
>
> Pick a direction for every current and keep it. Current flowing *through a resistor* makes the end it enters higher than the end it leaves, by I × R. If your final answer for a current is negative, nothing is wrong: the real current simply flows the opposite way to the direction you assumed.

Worked example 1: a series-parallel network (the chapter's circuit)

Nine volts feeds R1 = 1 kΩ in series with a parallel pair, R2 = 2.2 kΩ and R3 = 3.3 kΩ. We want every current and voltage.




```
Step 1: collapse the parallel pair (product over sum)
R23 = (2200 x 3300) / (2200 + 3300) = 7,260,000 / 5,500 = 1,320 ohm

Step 2: add the series resistor
R_total = 1000 + 1320 = 2,320 ohm

Step 3: total current (Ohm's law)
I = 9 / 2320 = 0.003879 A = 3.879 mA
```

```
Step 4: voltages (KVL: the two drops must add to 9 V)
V(R1)  = 3.879 mA x 1 k    = 3.879 V
V(X)   = 3.879 mA x 1.32 k = 5.121 V
Check: 3.879 + 5.121 = 9.000 V

Step 5: branch currents (same 5.121 V across each)
I2 = 5.121 / 2200 = 2.328 mA
I3 = 5.121 / 3300 = 1.552 mA
KCL check at node X: 2.328 + 1.552 = 3.880 mA = I (rounding)

Shortcut, the current divider: each branch takes a share inversely
proportional to its resistance. For two branches:
I2 = I x R3 / (R2 + R3) = 3.879 x 3300 / 5500 = 2.327 mA
(the SMALLER resistor takes the BIGGER share.)
```

Worked example 2: two sources, one unknown (node method)

Series and parallel reduction fails when two sources feed the same network. Kirchhoff handles it by choosing one unknown, the voltage V_A at the shared node, and applying KCL there.

```
 9 V (+) ----[ R1 1k ]----+
                          |
                          |   I1 and I2 flow into node A
 3 V (+) ----[ R2 1k ]--+----[ R3 1k ]----+
                        A      I3 -->     |
                                          0 V (ground)

 Both sources' (-) terminals are also connected to 0 V (ground).
 Node A is the junction of R1, R2 and R3.
```

```
Assume I1 and I2 flow INTO node A, and I3 flows OUT through R3.
Each branch current is (voltage difference) / R:
I1 = (9 - Va) / 1000
I2 = (3 - Va) / 1000
I3 = Va / 1000

KCL at A: I1 + I2 = I3
(9 - Va)/1000 + (3 - Va)/1000 = Va/1000
9 - Va + 3 - Va = Va
12 = 3 Va
Va = 4.0 V
```

```
Back-substitute
I1 = (9 - 4)/1000 = +5.0 mA
I2 = (3 - 4)/1000 = -1.0 mA     (negative: current really flows OUT of node A into the 3
I3 = 4/1000       = +4.0 mA

KCL:  5.0 + (-1.0) = 4.0 mA.  Good.
KVL, loop 9 V -> R1 -> R3:  I1 x 1k + I3 x 1k = 5 + 4 = 9 V.  Good.
KVL, loop 3 V -> R2 -> R3:  I2 x 1k + I3 x 1k = -1 + 4 = 3 V.  Good.
```

Why the negative sign matters: the larger source (9 V) overpowers the smaller one and pushes current backwards through it. A 3 V source being pushed backwards is, in effect, being charged. Keep that in mind whenever two supplies meet at a node, a situation you will meet again with battery packs and regulators in Basic Electronics 9.

3. The multimeter

A digital multimeter (DMM) has a dial, a display and three sockets. It measures voltage, current and is all this course needs.

SETTING	PROBES IN SOCKETS	CONNECT IT	NOTES
DC volts (V with a straight line, or V⎓)	Black in COM, red in VΩ	**Across** the part (in parallel)	Circuit powered. Reverse leads give a negative reading: harmless.
AC volts (V~)	Same	Across	Not used in this course's low-voltage circuits.
DC current (mA or A)	Black in COM, red in mA (or 10A for large currents)	**In series**: break the circuit and put the meter in the gap	Powered. Never across a source (short circuit).
Resistance (Ω)	Black in COM, red in VΩ	Across the part, **circuit unpowered**	The meter supplies its own tiny current.
Continuity (diode / beep symbol)	Same	Across a wire or joint, unpowered	Beeps if resistance is below roughly 50 Ω.

How to measure each quantity

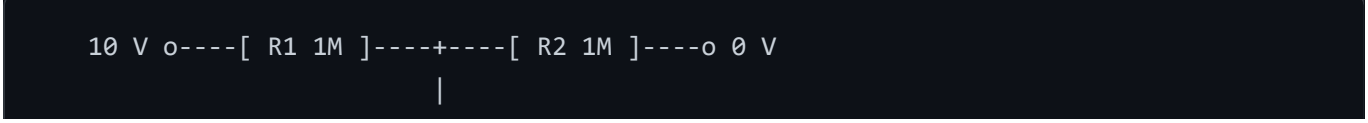
1. **Voltage:** set the dial to DC V at a range above the expected voltage (or auto-range). Touch the black probe to the lower-voltage side (usually ground) and the red probe to the point of interest. The meter reads the potential difference between the two probe tips. To measure the drop across R1 in the chapter circuit, put one probe on each leg of R1.
2. **Current:** turn the power off, *break* the loop at the point you want, set the dial to DC mA, move the red lead to the mA socket, and complete the loop through the meter. The same current now flows through the meter. Afterwards **move the red lead back to VΩ**: leaving it in the current socket and then measuring a voltage is the classic way to blow the meter's fuse and short the circuit.
3. **Resistance:** power off, and ideally take one leg of the part out of the circuit. If you measure in place, every other path between the two points is in parallel with the part you wanted, and the reading is lower than the part's value.
4. **Continuity:** test a jumper wire (should beep), the two ends of a breadboard rail (does the rail run the full length?), or two points that should *not* be connected (must not beep). It is the fastest fault-finding tool you own.

> **THREE HABITS THAT SAVE METERS AND COMPONENTS**
>
> - Ammeter: **series only**. Voltmeter: **parallel only**. A current setting across a battery is a short circuit.
> - Never measure resistance on a powered circuit, and never on a charged capacitor.
> - After any current measurement, return the red lead to the VΩ socket and the dial to a safe voltage range.

The meter is part of the circuit: loading

Measuring changes the thing you measure. A **voltmeter** has a high input resistance (typically 10 MΩ, as little as 1 MΩ on a cheap or analogue meter) and draws a little current through it. An **ammeter** has a small resistance (its *burden*, a few ohms on the mA range) added into the loop. Both change the circuit slightly; the effect matters when the circuit resistance is comparable with the meter's.

Worked example: voltmeter loading. Two 1 MΩ resistors divide 10 V. The midpoint should be 5 V. A meter with 10 MΩ input is placed across the lower resistor, so the meter's resistance is in parallel with it:



```
                      (midpoint)  <--- probe here
                          |
                   [ meter 10M ] to 0 V

  Lower leg with the meter attached: R2 || Rmeter
  R2 || Rm = (1M x 10M) / (1M + 10M) = 0.9091 M ohm

  Divider (Basic Electronics 3 derives this properly):
  Vmid = Vin x R_lower / (R_upper + R_lower)
  Vmid = 10 x 0.9091 / (1 + 0.9091) = 10 x 0.4762 = 4.76 V     (true value 5.00 V: -4.8 %)

  Same circuit, meter of 1 M ohm:
  R2 || Rm = 0.5 M ohm;  Vmid = 10 x 0.5 / 1.5 = 3.33 V        (-33 %!)

  Same meter (10M), but 1 k resistors instead of 1 M:
  R2 || Rm = 999.9 ohm;  Vmid = 10 x 999.9 / 1999.9 = 5.000 V   (no visible error)
```

Worked example: ammeter burden. A 5 V supply across a 100 Ω resistor should carry I = 5 / 100 = 50 mA. If the meter's mA range has 10 Ω of internal resistance, the loop becomes 110 Ω and I = 5 / 110 = 45.5 mA, a 9 % error. In the 9 V, 2.32 kΩ circuit of section 2 the same meter changes the current by only 9 / 2330 versus 9 / 2320, a 0.4 % effect.

> **RULE OF THUMB**
>
> A voltmeter's resistance should be at least 10 to 100 times the resistance of the circuit at the point you probe. Be suspicious of any reading in a megohm-range circuit; remember also that a voltmeter that reads 0 V on a floating or high-resistance point may only be loading it.

4. The breadboard

A solderless breadboard is a plastic block with spring-clip contacts hidden under a grid of holes. You push component legs (up to about 0.8 mm thick) and solid jumper wires into the holes. Nothing is soldered, so circuits can be changed in seconds, but only if you know which holes are connected underneath.

```
  Top view of a half-size breadboard (columns 1 to 30, only part drawn)

     + rail  (+) o o o o o o o o o o o o o ...   <-- connected along the whole rail
     - rail  (-) o o o o o o o o o o o o o ...   <-- connected along the whole rail
```

```
            column:  1  2  3  4  5  ...
            row a    o  o  o  o  o          \
            row b    o  o  o  o  o           |  a-e of one column are
            row c    o  o  o  o  o           |  connected together
            row d    o  o  o  o  o           |
            row e    o  o  o  o  o          /
          ------- centre gap (not connected across) -------
            row f    o  o  o  o  o          \
            row g    o  o  o  o  o           |  f-j of one column are
            row h    o  o  o  o  o           |  connected together
            row i    o  o  o  o  o           |
            row j    o  o  o  o  o          /

  + rail  (+) o o o o o o o o o o o o o ...
  - rail  (-) o o o o o o o o o o o o o ...
```

- Each **column** is two groups of five: a to e together, and f to j together, with **no connection across the central gap**. The gap is sized to straddle a standard 0.3-inch (7.62 mm) DIP integrated circuit, so each pin gets its own isolated column of four spare holes.
- The **power rails** (the long strips at the edges) run along the length of the board. On a full-size board some rails are *split in the middle*, so check with the multimeter's continuity setting before assuming a rail is live at both ends.
- Red conventionally marks +, blue or black marks −. Use **one rail for + and one for ground** and keep jumper wires short and flat, which makes the layout readable and easy to debug.
- Breadboard contacts add a little resistance and are not for high current. A rule of thumb is to keep to under about 1 A in total, which is plenty for sensors, LEDs and logic, and not for motors (Basic Electronics 9).

5. Build it: the chapter circuit

Build the series-parallel network from section 2 (Worked example 1) and check your calculations against meter readings.

Parts list

PART	VALUE	COLOUR CODE (4-BAND)
Battery	9 V (PP3) with snap clip	
R1	1 kΩ, 1/4 W	brown-black-red
R2	2.2 kΩ, 1/4 W	red-red-red
R3	3.3 kΩ, 1/4 W	orange-orange-red
Breadboard	Half size or larger, plus 2 jumper wires	
Multimeter	Any digital (needed for the real build)	

Real breadboard layout

Each connection below is one column. Numbers refer to the column; the letters are the rows used.

```
Hole list (one column per connection):
   J1:  + rail  -> column 5, row b
   R1:  column 5 row a  -> column 10 row a
   R2:  column 10 row b -> column 20 row b
   R3:  column 10 row c -> column 20 row c
   J2:  column 20 row d -> - rail
   Battery red -> + rail,  black -> - rail  (connect LAST)
```

1. Wire everything with the battery disconnected, as listed above.
2. Before connecting power, switch the multimeter to resistance (Ω) and put the probes on the + and − rails. You should read about 2.32 kΩ. A very low reading means a short somewhere (do not connect the battery), and an open-circuit reading means a missing connection.
3. Connect the battery (red to +, black to −) and switch to DC volts.
4. Measure the battery voltage first, then the voltage across R1, then between node X (column 10) and the − rail.
5. Break the circuit at R1 (lift one leg out of its hole) and put the meter, on DC mA, between the loose leg and the hole: that is the total current. Repeat by lifting the R2 leg at column 20 for R2's branch current.

Simulator version

Falstad Circuit Simulator (falstad.com/circuit): add a DC voltage source set to 9 V and three resistors set to 1 k, 2.2 k and 3.3 k (double-click a part to change its value), wired as in the schematic, with a ground symbol on the lower rail. Hover over any component to see its voltage and current; use the right-click menu (Add Ammeter, or the Scope option) for a persistent display.

Tinkercad Circuits (tinkercad.com): place a 9 V battery and three resistors on a breadboard, wire per the hole list above, set each resistor's value in its panel, then add a multimeter. Click it to choose the mode: connect *across* a part for volts, *in series* for amps, and across the rails with the power disconnected for ohms. Click Start Simulation to take readings.

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
Rail to rail resistance (battery off)	2.32 kΩ	2.2 to 2.4 kΩ	2.32 kΩ
Battery voltage	9.00 V	9.3 to 9.6 V fresh	9.00 V
Across R1	3.88 V	3.9 to 4.1 V	3.88 V
Node X to ground	5.12 V	5.2 to 5.4 V	5.12 V
Total current (meter in series with R1)	3.88 mA	3.9 to 4.1 mA	3.88 mA
Current in R2	2.33 mA	2.3 to 2.5 mA	2.33 mA
Current in R3	1.55 mA	1.5 to 1.7 mA	1.55 mA

The real readings scale with the real battery voltage: a fresh 9 V battery that measures 9.4 V gives 9.4 / 2320 = 4.05 mA, for instance. Resistor tolerance (±5 % is typical for carbon-film resistors) adds a few percent. The checks that always hold, to within meter accuracy: **the two voltages add to the battery voltage** (KVL), and **the two branch currents add to the total current** (KCL).

Build: the two-source circuit

For Worked example 2, wire a 9 V battery, a pack of two AA cells (3 V), and three 1 kΩ resistors with all the negative terminals joined to the same rail: R1 from the 9 V + terminal to node A, R2 from the 3 V + terminal to node A, R3 from node A to the rail. Expected (calculated, then real within a few percent): V_A = 4.0 V; voltage across R1 = 5.0 V; across R3 = 4.0 V; R1 current 5.0

mA, R3 current 4.0 mA, and the R2 current reads **−1.0 mA** (the meter shows a minus sign when current flows against the direction its leads define, here out of node A towards the AA pack).

DO NOT LEAVE A PRIMARY CELL BEING PUSHED BACKWARDS

One milliamp through alkaline AAs for a few minutes is harmless, but ordinary alkaline cells must not be charged. Take the 2 x AA pack out of the circuit as soon as you have your readings. In the simulator the same circuit is completely safe.

6. Challenges

CHALLENGE 1: SERIES-PARALLEL NETWORK

A 12 V supply feeds R1 = 470 Ω in series with R2 = 680 Ω in parallel with R3 = 1.2 kΩ. Find the equivalent resistance, the total current, the voltage across R1 and across the parallel pair, the current in R2 and R3, and the power in each of the three resistors. Show that the powers add to the power from the supply.

View solution

CHALLENGE 2: KIRCHHOFF'S LAWS

A 10 V source feeds node A through R1 = 1 kΩ. A 4 V source feeds the same node through R2 = 2 kΩ. R3 = 2 kΩ joins node A to ground. Use KCL at node A to find its voltage, then all three currents, and check your answer with KVL around both loops and with a power balance. Which way does current flow through the 4 V source?

View solution

CHALLENGE 3: BUILD, MEASURE AND TEST THE METER

Build (or simulate) the section 5 circuit and record all seven readings in the table. Then calculate what a 10 MΩ voltmeter and a 1 MΩ voltmeter would read at the midpoint of two 470 kΩ resistors across 9 V, and say by what percentage each is wrong. If you have a meter, try it.

View solution

QUICK REFERENCE

- **Schematic:** parts drawn by connection, not position; dot = junction; +V at top, ground at bottom; `4k7` = 4.7 kΩ
- **KCL:** current into a node = current out. **KVL:** going round a loop, rises = drops
- **Current divider (two branches):** I2 = I × R3 / (R2 + R3)
- **Node method:** name the unknown node voltage, write each branch current as (ΔV)/R, apply KCL, solve, verify with KVL
- **Meter:** volts across (parallel), amps through (series), ohms with power off; red lead back to VΩ afterwards
- **Loading:** voltmeter 10 MΩ in parallel with the part, ammeter a few Ω in series; keep circuit resistance well below/above these
- **Breadboard:** a to e joined, f to j joined, gap between; rails run lengthwise (may split in the middle)

Next, Basic Electronics 3 looks at resistors in depth (colour codes, power ratings), then uses the series rule to build voltage dividers, pull-up and pull-down resistors and a light-sensor circuit.

Basic Electronics

Course 1 · Chapter 3 · Resistors and Voltage Dividers

The resistor is the most common part in any circuit, and the voltage divider (two resistors in series, read from the middle) is the most useful thing you can build with it. Almost every sensor on a robot, from a light sensor to a button to a battery-level monitor, is read through a divider of some kind. This chapter covers how to read and choose resistors (colour codes, standard values, tolerance, power rating), then derives the divider formula, shows what happens when you connect a load to it, and uses it for pull-up and pull-down resistors, potentiometers and a light-sensitive circuit. It builds directly on Ohm's law from Basic Electronics 1 and the series, parallel and meter rules from Basic Electronics 2.

1. Choosing and reading resistors

Types

TYPE	TYPICAL TOLERANCE	TYPICAL POWER	WHERE YOU MEET IT
Carbon film	±5 %	1/8 W, 1/4 W, 1/2 W	Cheap hobby kits; fine for LEDs, pull-ups, general use
Metal film	±1 % (or better)	1/4 W, 1/2 W	Dividers, sensor circuits, anywhere the value matters
Wirewound / power	±1 to 5 %	1 W to 100 W	Motor, heater and load circuits; they run hot
Surface mount (SMD)	±1 to 5 %	1/16 W to 1/2 W	Every circuit board; value printed as digits (`472` = 47 × 10^2 = 4.7 kΩ)
Variable (pot, trimmer)	±20 %	0.1 to 0.5 W	Section 5

The colour code

Through-hole resistors are too small to print numbers on, so the value is a set of coloured bands. Hold the resistor with the tolerance band (gold or brown, set slightly apart) on the right and read from the left. In a **4-band** resistor the first two bands are digits, the third is the

multiplier (a power of ten) and the fourth is the tolerance. A **5-band** resistor (usual for 1 % metal film) has three digit bands, then the multiplier, then the tolerance.

COLOUR	DIGIT	MULTIPLIER	TOLERANCE
Black	0	× 1	
Brown	1	× 10	±1 %
Red	2	× 100	±2 %
Orange	3	× 1 k	
Yellow	4	× 10 k	
Green	5	× 100 k	±0.5 %
Blue	6	× 1 M	±0.25 %
Violet	7	× 10 M	±0.1 %
Grey	8		
White	9		
Gold		× 0.1	±5 %
Silver		× 0.01	±10 %

```
4-band examples (digit, digit, multiplier, tolerance)
 brown  black  red     gold   =  1 0 x 100   =  1,000 ohm  = 1 k    +/-5 %
 yellow violet red     gold   =  4 7 x 100   =  4,700 ohm  = 4.7 k  +/-5 %
 brown  black  orange  gold   =  1 0 x 1000  = 10,000 ohm  = 10 k   +/-5 %
 orange orange brown   gold   =  3 3 x 10    =    330 ohm           +/-5 %

5-band example (digit, digit, digit, multiplier, tolerance)
 yellow white  white   brown  brown  =  4 9 9 x 10  = 4,990 ohm = 4.99 k  +/-1 %

Tolerance turns one value into a range
 4.7 k +/-5 %  :  4,700 x 0.95 = 4,465 ohm   to   4,700 x 1.05 = 4,935 ohm
```

READING TIP

If you cannot tell which end is which, find the end whose band is nearest the edge: that is the first digit. When in doubt, **measure it** with the multimeter (Basic Electronics 2), with the resistor out of the

circuit. The code tells you what the maker intended; the meter tells you what you have.

Why 4.7 k and not 4.5 k? The E-series

Resistors are made in a fixed set of *preferred values*, called the E-series, spaced so that each value is a fixed ratio above the last. The E12 series has 12 values per decade, so the ratio is $10^{1/12} = 1.212$. Rounded to two digits it gives:

```
E12 (+/-10 %):  12 values per decade, ratio 1.212 between neighbours
 10  12  15  18  22  27  33  39  47  56  68  82      (then 100, 120, 150 ...)

Calculated: 10 x 1.212^n  =  10, 12.1, 14.7, 17.8, 21.5, 26.1, 31.6, 38.3,
 46.4, 56.2, 68.1, 82.5  -->  rounded to the table above

E24 (+/-5 %): adds 11 13 16 20 24 30 36 43 51 62 75 91
E96 (+/-1 %): 96 values per decade (e.g. 10.0, 10.2, 10.5, 10.7 ... 4.99, 5.11)

Why this spacing: with +/-10 % tolerance, 10 spans 9 to 11 and 12 spans
10.8 to 13.2. The ranges just overlap, so any wanted value is within about
10 % of something you can buy, and no value is made that another already covers.
```

In practice a hobby kit holds E12 or E24 values (1 k, 2.2 k, 4.7 k, 10 k, 22 k ...). When your calculation gives an awkward value, pick the nearest standard value and recalculate what the circuit does with it, as the divider design in section 3 shows.

2. Power rating and heat

A resistor turns all of the electrical power it receives into heat. The power is found from Ohm's law three equivalent ways:

```
P = V x I      (definition: volts x amps = watts)
Substitute I = V/R:   P = V^2 / R
Substitute V = I x R: P = I^2 x R

Worked example 1: 5 V across 100 ohm
I = 5 / 100 = 50 mA
P = V x I = 5 x 0.050 = 0.25 W        (or 5^2/100 = 0.25 W, or 0.05^2 x 100 = 0.25 W)
A 1/4 W resistor is exactly at its limit: too hot to touch, with a short life.
```

```
Worked example 2: 12 V across 470 ohm
P = 12^2 / 470 = 144 / 470 = 0.306 W
This exceeds a 1/4 W (0.25 W) rating, so the resistor would burn out.
```

Safety margin and derating

The rated power is what the part can dissipate at a stated ambient temperature (usually 70 °C) and still last. The manufacturer's **derating curve** is a straight line falling to zero power at the maximum temperature (typically 155 °C). So the hotter the surroundings, the less the resistor may dissipate:

```
Allowed power at ambient T (for T between 70 C and 155 C):
 P_allowed = P_rated x (155 - T) / (155 - 70) = P_rated x (155 - T) / 85

1/4 W resistor in a warm enclosure at 100 C:
 P_allowed = 0.25 x (155 - 100) / 85 = 0.25 x 55 / 85 = 0.162 W

Rule of thumb: choose a rating of at least 2 x the actual power.
Example 2 again: need P_rating >= 2 x 0.306 = 0.613 W  -->  choose a 1 W resistor
(a 1/2 W part would run at 61 % of its rating, which is too hot to be comfortable).
```

> **HOT RESISTORS**
>
> A resistor running at its full rating has a surface temperature that can exceed 100 °C, enough to burn skin and to melt the plastic of a breadboard or wire insulation. If a resistor is hot enough that you notice, it is under-rated. Switch off and recalculate.

3. The voltage divider

Two resistors in series across a supply share its voltage in proportion to their sizes. Take the voltage from the junction between them, and you have a lower voltage than the supply. This is a **voltage divider**.

```
 Vin o----+
          |
        [ R1 ]       (upper resistor)
          |
          +-------o Vout
```

```
          |
        [ R2 ]       (lower resistor)
          |
 0 V  o---+---------o 0 V (ground)
```

Derivation

```
The two resistors are in series (Basic Electronics 1), so one current flows:
 I = Vin / (R1 + R2)

Vout is the voltage across R2 alone (Ohm's law):
 Vout = I x R2 = Vin x R2 / (R1 + R2)

So the output is the fraction R2/(R1+R2) of the input, always between 0 and Vin.
```

Worked example 1: 5 V down to 3.3 V

This is a real task: a 5 V sensor output going to a 3.3 V logic pin. Try R1 = 1 kΩ and R2 = 2 kΩ.

```
Vout, using the formula
Vout = 5 x 2000 / (1000 + 2000) = 5 x 0.6667 = 3.333 V

Divider current and power (the price of a divider is that it wastes current)
I  = 5 / 3000 = 1.667 mA
P1 = I^2 R1 = 0.001667^2 x 1000 = 2.78 mW
P2 = I^2 R2 = 0.001667^2 x 2000 = 5.56 mW      (total 8.33 mW = 5 V x 1.667 mA)

Design in reverse: wanted 3.3 V from 5 V. Choose R2 = 10 k, solve for R1:
 Vout/Vin = R2/(R1+R2)   -->   R1 = R2 x (Vin/Vout - 1) = 10 k x (5/3.3 - 1) = 5.15 k
Nearest E24 value is 5.1 k:
Vout = 5 x 10 / (5.1 + 10) = 50 / 15.1 = 3.311 V    (0.3 % from target: good)
```

Tolerance: how accurate is a divider?

Take the 1 k / 2 k divider above. The worst high output comes when R2 is 5 % high and R1 5 % low, the worst low output the other way round:

```
5 % resistors
High:  R1 = 950, R2 = 2100:   Vout = 5 x 2100 / 3050 = 3.443 V
Low:   R1 = 1050, R2 = 1900:  Vout = 5 x 1900 / 2950 = 3.220 V
Output is 3.33 V +3.3 % / -3.4 %
```

```
1 % resistors
High:  R1 = 990, R2 = 2020:   Vout = 5 x 2020 / 3010 = 3.356 V
Low:   R1 = 1010, R2 = 1980:  Vout = 5 x 1980 / 2990 = 3.311 V
Output is 3.33 V +0.7 % / -0.7 %
```

The divider ratio is a little better than the resistors alone suggest, because errors in the two resistors partly cancel, but 1 % metal-film resistors are well worth the extra pennies when the voltage matters (a battery monitor, a reference level).

Loading: what happens when you connect something

The formula assumes nothing is connected to Vout. A load (an input pin, another circuit) of resistance R_L connects in *parallel* with R2, which makes the lower leg smaller and the output lower. This is exactly the meter-loading effect of Basic Electronics 2.

```
Same divider (5 V, R1 = 1 k, R2 = 2 k), now with a load RL across R2
Lower leg = R2 || RL = (R2 x RL) / (R2 + RL)

RL = 2 k:   2k || 2k  = 1.000 k    Vout = 5 x 1.000 / (1 + 1.000)  = 2.500 V
RL = 10 k:  2k || 10k = 1.667 k    Vout = 5 x 1.667 / (1 + 1.667)  = 3.125 V
RL = 20 k:  2k || 20k = 1.818 k    Vout = 5 x 1.818 / (1 + 1.818)  = 3.226 V   (-3.2 %)
RL = 1 M:   2k || 1M  = 1.996 k    Vout = 5 x 1.996 / (1 + 1.996)  = 3.331 V   (no visib

Rule of thumb: the load should be at least 10 x R2 (3 % error or less here;
100 x R2 for under 0.5 %). Or make R1 and R2 smaller, at the cost of more divider curren
```

The Thevenin view of a loaded divider

There is a tidy way to see this. From the output terminal, a divider looks like *one* source in series with *one* resistor (Thevenin's theorem, 1883). The source voltage V_{th} is what the output reads with nothing connected. The resistor R_{th} is what you measure between the output and ground with the supply replaced by a wire (so R1 and R2 are in parallel). Then the load simply forms a second divider with R_{th}:

```
Vth = Vin x R2 / (R1 + R2)            (unloaded output)
Rth = R1 || R2 = R1 x R2 / (R1 + R2)    (output resistance)

Vout(loaded) = Vth x RL / (Rth + RL)
```

```
Check with R1 = 1 k, R2 = 2 k, RL = 10 k, Vin = 5 V:
 Vth = 3.333 V
 Rth = 1000 x 2000 / 3000 = 666.7 ohm
 Vout = 3.333 x 10000 / (666.7 + 10000) = 33,333 / 10,666.7 = 3.125 V     (matches the d

Load error is small when Rth is much less than RL. Use this to ask "how stiff
is my divider?": a low Rth (small resistors) is stiff, a high Rth is weak.
```

4. Pull-up and pull-down resistors

A digital input pin reads 1 if it sees a high voltage and 0 if it sees a low one. If nothing is connected to the pin (it **floats**), it picks up stray electric fields from your hand, the mains wiring and neighbouring wires, and reads randomly. A button that only connects the pin to one rail leaves it floating whenever it is not pressed. The fix is a resistor that gently holds the pin to the other rail until something stronger takes over.



Each is a voltage divider where one leg is the button (0 Ω closed, infinite open) and the other is the pull resistor:

- **Button open:** the input pin draws almost no current (a CMOS input leaks about 1 µA at most). The resistor carries that tiny current, so the voltage across it is I × R = 1 µA × 10 kΩ = 10 mV. The pin sits within 10 mV of the rail it is tied to: a clean logic level.
- **Button pressed:** the button shorts the pin to the opposite rail, and the resistor now carries a current of V/R from supply to ground: 3.3 / 10 kΩ = 0.33 mA, which is small and only flows while the button is held.

Choosing the value

```
Lower limit: pressed-button current and wasted power
 I_pressed = Vcc / R   Want I_pressed to be at most about 0.2 mA
 R >= 3.3 V / 0.2 mA = 16.5 k          (10 k = 0.33 mA is common and acceptable)

Upper limit 1: leakage must not drag the level out of range
 A logic high must be at least about 0.7 x Vcc = 2.31 V on a 3.3 V part, so the
 resistor may drop at most 3.3 - 2.31 = 0.99 V.
 With 1 uA leakage:  R <= 0.99 V / 1 uA = 990 k   (never the real limit)

Upper limit 2: speed and noise. The pin and wiring have capacitance C (say 20 pF).
 Rising edge takes about 3 time constants:   tau = R x C
 10 k x 20 pF = 0.2 us   (fast)       100 k x 20 pF = 2 us (still fine for a button)
 A high resistance also makes the pin easier for noise to disturb.

Typical choice: 4.7 k to 47 k;  10 k is the default for a button.
```

INTERNAL PULL-UPS

Most microcontrollers have optional internal pull-up resistors (tens of kilo-ohms, for example 20 to 50 kΩ on an Arduino Uno) that you switch on in software. That saves a part for a simple button, but an external resistor gives a known value, which matters for long wires or noisy places. Basic Electronics 8 returns to pull-ups for open-drain buses, and Basic Electronics 10 shows the code. Pull-up or pull-down is a design choice; with a pull-up the button reads *inverted* (0 means pressed), which is the common convention.

5. The potentiometer: a divider you can turn

A potentiometer (pot) is a resistive track with a sliding contact called the **wiper**. It has three terminals: the two ends of the track, and the wiper. The resistance between the ends is fixed (say 10 kΩ). The wiper splits the track into two parts, which makes a voltage divider whose ratio you set with the knob.

```
Vin o----+
         |
       [ track upper part ]   resistance (1 - x) x R
         |
         +--------- wiper ---o Vout
         |
       [ track lower part ]   resistance x R
         |
0 V  o---+

x = fraction of the way from the ground end to the supply end (0 to 1)
Unloaded:  Vout = Vin x (x R) / R = x x Vin     (the output is simply a fraction of the
```

```
10 k pot on 5 V, no load:
 x = 0.25:  Vout = 1.25 V        x = 0.5:  Vout = 2.50 V        x = 1.0:  Vout = 5.00 V

Loaded with RL = 10 k at the wiper (x = 0.5):
 Lower leg = 5 k || 10 k = 3.333 k;   upper leg = 5 k
 Vout = 5 x 3.333 / (5 + 3.333) = 16.667 / 8.333 = 2.00 V      (not 2.50 V: a 10 k load

Rule of thumb: the load should be at least 10 x the pot value.
10 k pot with a 100 k load, x = 0.5: 5 k || 100 k = 4.762 k;  Vout = 5 x 4.762 / 9.762 =
```

- **Linear (B) and logarithmic (A) taper.** A linear pot's resistance changes evenly with angle, which is what you want for position and sensor inputs. A logarithmic pot is for audio volume controls, matching the ear's sense of loudness. For robotics use **linear**.
- **As a variable resistor (rheostat):** use only the wiper and one end; the resistance between them goes from 0 to R. Tie the unused end to the wiper to stop the track going open-circuit if the wiper lifts.
- **Trimmers** are small pots for one-off adjustment with a screwdriver. Servo and joystick position sensors are often pots too.

- A pot on an ADC input (Basic Electronics 10) reads a number from 0 to 1023 on a 10-bit converter: at 2.50 V on a 5 V reference the reading is about 2.50 / 5 × 1023 = 512.

6. A light sensor from a divider

A **light-dependent resistor** (LDR, or photoresistor) is a resistor whose value falls as light on it increases. It is a perfect partner for the divider because a microcontroller reads voltage, not resistance, and the divider turns resistance into voltage.

Typical figures for a small cadmium-sulphide cell (they vary a lot between parts, so measure yours with the meter):

LIGHT	LDR RESISTANCE (ILLUSTRATIVE)
Darkness	about 1 MΩ or more
Dim room	about 20 kΩ
Bright room	about 5 kΩ
Bright sunlight	about 500 Ω

Place the LDR as the *upper* resistor (R1) and a fixed 10 kΩ resistor as the lower one (R2), so the output **rises with light**:

```
  +5 V o----+
           |
         [ LDR ]        (R1: falls with light)
           |
           +--------o Vout   (to an ADC or comparator)
           |
         [ 10 k ]       (R2)
           |
  0 V   o---+

 Vout = 5 x 10 k / (R_LDR + 10 k)
  Dark         R = 1 M:     Vout = 5 x 10 / 1010  = 0.050 V
  Dim room     R = 20 k:    Vout = 5 x 10 / 30    = 1.667 V
  Bright room  R = 5 k:     Vout = 5 x 10 / 15    = 3.333 V
  Bright sun   R = 500:     Vout = 5 x 10 / 10.5  = 4.762 V
```

```
Swap the two parts (LDR at the bottom) and the output FALLS as it gets brighter.
```

Choosing the fixed resistor. At Vout = half supply, a given change in the LDR resistance gives the biggest change in output. The divider is at half supply when R2 equals the LDR resistance, so choose **R2 equal to the LDR resistance at the brightness you most care about**: about 20 kΩ for a dim-room detector, 5 kΩ for a daylight detector, 10 kΩ if you want a good compromise over the whole range. Because the LDR changes over several decades, the output is naturally compressed, which is good for an ADC: it gets a useful signal from a vast range of light.

> **WHAT NEXT?**
>
> The divider's output is an analogue voltage. To *decide* "is it dark?" you compare it with a threshold using a comparator or a microcontroller input; Basic Electronics 7 builds the comparator, and Basic Electronics 10 reads the same divider with an ADC. LDRs are also slow (tens of milliseconds), which is fine for ambient light but useless for detecting flicker or fast pulses; a phototransistor is the answer there.

7. Build it: three dividers

Build the three circuits one at a time, taking each apart before the next. All use the same supply: **5 V** from a USB power adapter, a breadboard power module, or the 5 V pin of an Arduino or similar board (use the board only as a power supply for now). If you use 3 AA cells (4.5 V), multiply every voltage reading below by 0.9.

Parts list

PART	VALUE	COLOUR CODE (4-BAND)
R1 (stage A)	1 kΩ, 1/4 W	brown-black-red
R2 (stage A)	2 kΩ (or 2.2 kΩ, with new readings), 1/4 W	red-black-red
RL (load)	10 kΩ, 1/4 W	brown-black-orange
Potentiometer (stage B)	10 kΩ linear	
LDR (stage C)	Small CdS photoresistor	

PART	VALUE	COLOUR CODE (4-BAND)
R (stage C)	10 kΩ, 1/4 W	brown-black-orange
Supply	5 V (see above)	
Breadboard, 4 jumper wires, multimeter		

Measure each resistor with the meter and write the value down before you fit it: you will use the measured values to check your calculations.

Stage A: fixed divider and loading

```
Hole list (one column per connection):
   J1:  + rail  -> column 5, row b
   R1:  column 5 row a   -> column 10 row a      (1 k, upper)
   R2:  column 10 row b  -> column 20 row b      (2 k, lower)
   J2:  column 20 row c  -> - rail
   RL:  column 10 row c  -> - rail               (10 k load; fit last, remove to compare
   Supply + to + rail, supply - to - rail          (connect LAST)
   Measure: red probe on column 10 row e, black probe on the - rail
```

1. Wire with the supply disconnected. Check with the meter on Ω between the rails: with RL not yet fitted you should read about 3 kΩ.
2. Connect the supply and read the supply voltage first (it is rarely exactly 5.00 V).
3. Read Vout without RL (about 3.33 V), then fit RL and read again (about 3.13 V). Then swap RL for a 2 kΩ resistor (about 2.50 V).

Stage B: potentiometer

```
   Pot legs in columns 25, 26, 27 (row a), left to right, wiper = middle leg
   J1:  column 25 row b -> + rail
   J2:  column 27 row b -> - rail
   Measure: red probe on column 26 row c (wiper), black probe on the - rail
   Load test: 10 k resistor from column 26 row d -> - rail
```

Turn the knob from one end to the other, watching the reading sweep from 0 V to 5 V. Set it to about half way (read 2.50 V) and add the 10 kΩ load: the reading should fall to about 2.00 V.

Stage C: light sensor

```
Use the stage A layout with R1 = LDR and R2 = 10 k:
J1:   + rail -> column 5 row b
LDR:  column 5 row a  -> column 10 row a
R2:   column 10 row b -> column 20 row b      (10 k)
J2:   column 20 row c -> - rail
Measure: column 10 row e to the - rail
```

Before applying power, measure the LDR's resistance in room light, in the shade of your hand, and with a torch on it, and compare your Vout readings against $5 \times 10\text{ k} / (R_{LDR} + 10\text{ k})$ using those resistances. Cover the sensor and the output should fall towards 0 V; shine a torch on it and it should rise.

Simulator version

Falstad Circuit Simulator (falstad.com/circuit): add a DC voltage source and set it to 5 V, resistors set to 1 k, 2 k and 10 k (double-click a part to edit its value), and a ground symbol. Hover over the middle node or use the right-click Scope option to see the voltage. For stage B use the *Potentiometer* from the Passive Components menu (its slider sets the wiper position). If your version has no photoresistor, a second potentiometer or a resistor you edit by hand stands in for the LDR.

Tinkercad Circuits (tinkercad.com): place a breadboard, resistors (their value is set in the panel that opens when you click a part), a *Potentiometer* and a *Photoresistor* (click it to move the light slider), a 5 V *Power Supply* and a *Multimeter*. Wire per the hole lists above, set the multimeter to volts, and press Start Simulation. To compare against the formula, set the multimeter to Ω, with the supply disconnected, to read the photoresistor's resistance at the slider's current position, then calculate the expected voltage.

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
A: rail to rail resistance (supply off, no load)	3.00 kΩ	2.8 to 3.2 kΩ	3.00 kΩ

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
A: Vout, no load	3.33 V	3.2 to 3.45 V (scales with supply)	3.33 V
A: Vout, RL = 10 k	3.13 V	3.0 to 3.25 V	3.13 V
A: Vout, RL = 2 k	2.50 V	2.4 to 2.6 V	2.50 V
B: wiper at half, no load	2.50 V	depends on the knob position	2.50 V
B: wiper at half, RL = 10 k	2.00 V	1.9 to 2.1 V	2.00 V
C: LDR 20 k (dim)	1.67 V	use your measured resistance	formula with the shown resistance
C: LDR 5 k (bright)	3.33 V	use your measured resistance	formula with the shown resistance

The real readings scale with the real supply voltage (a supply reading 5.10 V gives 5.10 × 2/3 = 3.40 V in stage A), and resistor tolerance adds a few percent. The result that holds whatever the part tolerances is the **ratio**: Vout divided by the supply voltage should be 0.667 in stage A before loading, and 0.50 in stage B at half travel.

8. Challenges

CHALLENGE 1: DECODE, ENCODE AND POWER

(a) Give the value and the tolerance range of four resistors: brown-black-orange-gold; yellow-violet-red-gold; green-blue-brown-brown; and 5-band red-red-black-red-brown. (b) Write the colour bands for 330 Ω 5 %, 6.8 kΩ 5 % and 100 kΩ 1 % (5-band). (c) A 1/4 W resistor (rated at 70 °C, zero at 155 °C) carries 0.15 W in a box at 100 °C: is that within its derated limit? (d) 12 V is applied across 470 Ω: find the power and the smallest standard rating (1/4, 1/2, 1 or 2 W) that keeps it at or below 50 % of the rating.

View solution

CHALLENGE 2: DESIGN AND LOAD A DIVIDER

Design a divider from 9 V that gives about 2.5 V with about 1 mA flowing in it, choosing two E24 values. Find the actual Vout, then the Vout when a 10 kΩ load is connected (use the Thevenin method and check it directly). By what percentage does loading change the output? What load resistance keeps the error under 2 %? Finally, rescale both resistors to a tenth of their value and say what you gain and what you pay.

View solution

CHALLENGE 3: LIGHT SENSOR AND PULL-UP

An LDR measures 500 kΩ in the dark, 15 kΩ in room light and 2 kΩ in bright light, and is the upper resistor of a divider on 5 V. (a) Find Vout for each light level with a 10 kΩ lower resistor. (b) Choose the lower resistor so that Vout is 2.5 V in room light, and find the other two outputs. (c) Choose a pull-up resistor for a push button on a 3.3 V logic input, requiring a pressed current under 0.2 mA, and give the pressed current, the power in the resistor, and the rise time constant with 20 pF on the pin. If you have the parts or a simulator, build (a) and record the readings.

View solution

QUICK REFERENCE

- **Colour code:** digits 0 to 9 = black, brown, red, orange, yellow, green, blue, violet, grey, white; multiplier is the power of ten; gold = 5 %, brown = 1 %
- **E12:** 10 12 15 18 22 27 33 39 47 56 68 82 (× powers of ten); E24 and E96 for 5 % and 1 %
- **Power:** $P = VI = V^2/R = I^2R$; choose a rating of at least 2 × P; derate above 70 °C: $P \times (155 - T)/85$
- **Divider:** $V_{out} = V_{in} \times R2 / (R1 + R2)$; $R1 = R2 \times (V_{in}/V_{out} - 1)$
- **Loading:** replace R2 by R2 || RL, or use $V_{th} = V_{in} \times R2/(R1+R2)$, $R_{th} = R1 \parallel R2$, $V_{out} = V_{th} \times RL/(R_{th} + RL)$; keep RL ≥ 10 × R2
- **Pull-up / pull-down:** 4.7 k to 47 k, 10 k typical; pull-up button reads 0 when pressed
- **Pot:** Vout = x × Vin unloaded; load at least 10 × the pot value; linear taper for sensors
- **LDR divider:** LDR on top, Vout rises with light; make the fixed resistor equal to the LDR at the light level that matters

Next, Basic Electronics 4 moves from parts that only resist to parts that store energy: capacitors and inductors, the RC time constant, and a simple filter and button-debounce circuit.

Basic Electronics

Course 1 · Chapter 4 · Capacitors and Inductors

Resistors only ever turn electrical energy into heat. Capacitors and inductors are different: they *store* energy, a capacitor in an electric field and an inductor in a magnetic field, and give it back later. That one difference makes time appear in your circuits. A capacitor takes time to fill, an inductor takes time to build its current, and a pair of simple resistor-and-capacitor circuits can delay a signal, smooth a supply, filter out noise or clean up a bouncing switch. It also explains why a motor or relay coil can destroy the transistor that switches it, which Basic Electronics 5, Basic Electronics 6 and Basic Electronics 9 will solve. This chapter builds on Ohm's law (Basic Electronics 1), series and parallel rules (Basic Electronics 2) and the divider (Basic Electronics 3).

1. Capacitors

What a capacitor is

A capacitor is two conductors (plates) separated by an insulator (the dielectric). No current flows *through* it, but when a voltage is applied, charge piles up on one plate and an equal charge leaves the other. The more charge it holds per volt, the bigger its **capacitance** C, measured in farads (F):

```
Definition: charge stored per volt
 Q = C x V          (Q in coulombs, C in farads, V in volts)

Practical units (a farad is enormous)
 1 mF = 0.001 F        1 uF = 0.000001 F        1 nF = 0.000000001 F        1 pF = 10^-1
 1 uF = 1000 nF = 1,000,000 pF

Worked example: 100 uF charged to 5 V
 Q = 100e-6 x 5 = 500e-6 C = 500 uC = 0.5 mC

Why practical capacitors are built the way they are: parallel plates, C = e x A / d
 (e = permittivity of the dielectric, A = plate area, d = gap). With air (e0 = 8.85 pF/m
 plates of 1 cm^2 (1e-4 m^2) and a 1 mm gap:
 C = 8.85e-12 x 1e-4 / 1e-3 = 8.85e-13 F = 0.885 pF
To get microfarads, makers roll very large areas of metal foil with micron-thin
 dielectric, or use dielectrics with a permittivity thousands of times that of air.
```

Energy stored

Moving a small extra charge dq onto a capacitor that already holds voltage v = q/C takes energy dE = v × dq. Adding that up from empty (0) to full charge Q:

```
 E = integral from 0 to Q of (q / C) dq = Q^2 / (2C)
 and since Q = C x V:

 E = 1/2 x C x V^2

Worked example: 100 uF at 5 V
 E = 0.5 x 100e-6 x 5^2 = 0.5 x 100e-6 x 25 = 1.25e-3 J = 1.25 mJ
That is tiny: an AA cell holds about 1.5 V x 2.5 Ah x 3600 = 13,500 J, roughly ten
 million times more. Capacitors are for fast, small bursts, not for bulk energy.

The energy goes as the SQUARE of the voltage: doubling V gives 4 x the energy.
 1000 uF at 50 V holds 0.5 x 1e-3 x 2500 = 1.25 J: enough to feel or to blow a thin wire
```

CHARGED CAPACITORS

A large capacitor stays charged after the power is off. At 5 V that is harmless, but never open a mains-powered supply or touch a high-voltage capacitor (tens of volts and up) without discharging it through a resistor first. Never short a charged capacitor with a screwdriver: the spark can weld metal and throw it at you. Use a resistor of 100 Ω or more.

Capacitors in parallel and series

```
PARALLEL: same voltage V across both, charges add (Q = Q1 + Q2 = C1 V + C2 V)
 C_total = C1 + C2 + C3 ...        (more plate area, so more capacitance)
 10 uF || 22 uF = 32 uF

SERIES: the same charge Q on each, voltages add (V = Q/C1 + Q/C2 = Q (1/C1 + 1/C2))
 1 / C_total = 1/C1 + 1/C2 + ...    (note: the opposite of resistors)
 Two only:  C_total = C1 x C2 / (C1 + C2)

 10 uF in series with 10 uF = 100 / 20 = 5 uF
 10 uF in series with 47 uF = 470 / 57 = 8.246 uF       (always smaller than the smalles

Voltage sharing in series: V across C1 = Vtotal x C2 / (C1 + C2)
 (the SMALLER capacitor gets the LARGER share, because the same Q gives V = Q/C)
 10 uF and 47 uF across 12 V:
```

```
    V(10 uF) = 12 x 47 / 57 = 9.895 V
    V(47 uF) = 12 x 10 / 57 = 2.105 V          (sum 12.000 V)
```

Capacitors combine the opposite way round to resistors: parallel adds, series reduces. Because the voltage shares unevenly, putting capacitors in series to survive a higher voltage is risky unless you add balancing resistors; the sensible fix is one capacitor with a higher voltage rating.

Types you will meet

TYPE	TYPICAL RANGE	TOLERANCE	POLARISED?	USE
Ceramic C0G / NP0	1 pF to 100 nF	±1 to 5 %	No	Precise, stable: filters, timing, radio
Ceramic X7R / X5R	100 pF to 10 µF and up	±10 to 20 %	No	Decoupling (the 100 nF "104" on every board); capacitance falls with applied DC voltage
Film (polyester, polypropylene)	1 nF to 10 µF	±5 to 10 %	No	Audio, stable timing, mains-rated types; physically large
Aluminium electrolytic	1 µF to 10,000 µF and up	±20 % (often −20 / +80 %)	**Yes**	Smoothing, bulk decoupling; leaky, ages, has series resistance (ESR)
Tantalum	0.1 µF to 470 µF	±10 to 20 %	**Yes**	Compact bulk decoupling; can fail short and burn if reversed or over-volted
Supercapacitor	0.1 F to 1000 F	−20 / +80 %	Yes	Memory back-up; about 2.7 V per cell

Reading the value

```
Small ceramics use the same code as SMD resistors, but the answer is in picofarads
 104  =  10 x 10^4 pF  = 100,000 pF = 100 nF = 0.1 uF     (the standard decoupling cap)
 103  =  10 x 10^3 pF  =  10 nF
 473  =  47 x 10^3 pF  =  47 nF
 224  =  22 x 10^4 pF  = 220 nF
 105  =  10 x 10^5 pF  = 1 uF
A letter after the digits is the tolerance: J = 5 %, K = 10 %, M = 20 %.
```

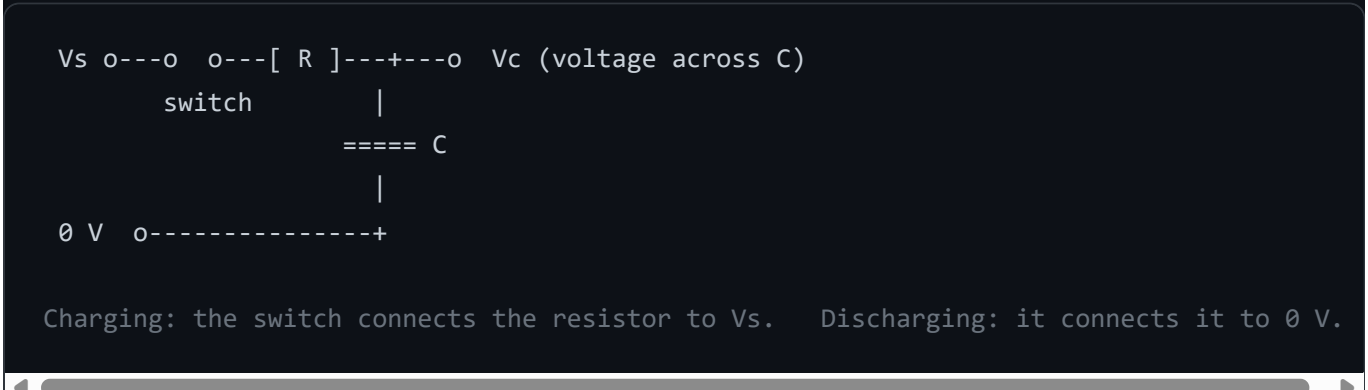
```
Electrolytics have the value, the voltage and the polarity printed on the can:
 220 uF  16 V     a stripe with minus signs marks the NEGATIVE leg; the new, long leg is
```

Voltage rating, polarity and tolerance

- **Voltage rating** is the most the capacitor may see continuously. Choose at least 1.5 to 2 × the working voltage for an electrolytic (a 10 V part on a 5 V rail, a 25 V part on 12 V). Ceramics also *lose capacitance* as the DC voltage across them rises ("DC bias derating"): a 10 µF X5R ceramic rated 6.3 V can deliver well under half of 10 µF at 5 V, depending on the part, so read the datasheet curve for a smoothing capacitor.
- **Polarity.** An electrolytic connected backwards leaks heavily, heats up and can vent or burst with a pop and a puff of electrolyte. Check the stripe before you power up. Ceramic and film capacitors have no polarity.
- **Tolerance** is wide: a 100 µF ±20 % electrolytic is anywhere from 80 to 120 µF, so a timing circuit built with it is only accurate to about that. Use film or C0G where the time or frequency really matters.

2. Charging and discharging through a resistor

Connect a resistor R in series with a capacitor C and a supply V_s through a switch. The moment the switch closes the capacitor is empty, so it looks like a short circuit and the current is V_s/R. As charge builds, the capacitor's voltage rises, the voltage left across the resistor falls, and the current dwindles. It is the same shape as water filling a bucket through a narrow pipe: fast at first, ever slower as it fills.



Derivation

```
1. The resistor drops what the capacitor does not: Vs - v = i x R
2. The capacitor's current is the rate of change of its charge: i = dQ/dt = C dv/dt
   (so a steady v means no current: a capacitor blocks DC once charged)
3. Combine: Vs - v = R C dv/dt        so   dv / (Vs - v) = dt / (R C)
4. Integrate, with v = 0 at t = 0:  -ln(Vs - v) + ln(Vs) = t / (R C)

 Charging:     v(t) = Vs x (1 - e^(-t / RC))
 Current:      i(t) = (Vs / R) x e^(-t / RC)

Discharging (C starts at V0 and the resistor goes to 0 V): same method with Vs = 0:
 v(t) = V0 x e^(-t / RC)

The product tau = R x C has units of time (ohm x farad = second) and is called the
 TIME CONSTANT. Everything is a function of t / tau.
```

The 63 % and 99 % points

t / tau	charging: v / Vs = 1 - e^-(t/tau)	discharging: v / V0 = e^-(t/tau)
0.5	39.3 %	60.7 %
1	63.2 %	36.8 %
2	86.5 %	13.5 %
3	95.0 %	5.0 %
4	98.2 %	1.8 %
4.6	99.0 %	1.0 %
5	99.3 %	0.7 %

Memory aid: after ONE time constant a charging capacitor is about 63 % full; after FIVE it is, for practical purposes, full (99.3 %).

Time to reach a chosen fraction f of the supply (rearranging the charging formula):

$t = -R\,C \times \ln(1 - f)$

half way ($f = 0.5$): $t = 0.693\ \tau$ 90 % ($f = 0.9$): $t = 2.303\ \tau$

Worked example: 10 kΩ and 220 µF from 5 V

$\tau = R \times C = 10{,}000 \times 220\text{e-}6 = 2.2$ s

Charging from 0 V:

$t = 2.2$ s (1 tau): $v = 5 \times (1 - e^{-1}) = 5 \times 0.632 = 3.16$ V

$t = 4.4$ s (2 tau): $v = 5 \times (1 - e^{-2}) = 5 \times 0.865 = 4.32$ V

$t = 6.6$ s (3 tau): $v = 5 \times 0.950 = 4.75$ V

```
 t = 10.1 s  (4.6 tau): v = 5 x 0.990                    = 4.95 V
 Time to reach 2.5 V:   t = 0.693 x 2.2 = 1.52 s

Current: it starts at Vs / R and decays with the same time constant
 i(0) = 5 / 10,000 = 0.500 mA      i(2.2 s) = 0.5 x 0.368 = 0.184 mA

Discharging from 5 V through the same resistor:
 t = 2.2 s:  v = 5 x 0.368 = 1.84 V      t = 4.4 s:  v = 0.677 V      t = 6.6 s:  v = 0.

Energy: the capacitor ends up storing 1/2 C V^2 = 0.5 x 220e-6 x 25 = 2.75 mJ.
 The supply delivered Q x V = (220e-6 x 5) x 5 = 5.5 mJ. Exactly HALF is lost as heat in
 whatever the value of R.

Tolerance: C +/-20 % and R +/-5 % give tau from 0.8 x 0.95 x 2.2 = 1.7 s to 1.2 x 1.05 x
```

THE SAME LAW EVERYWHERE

Any quantity that changes at a rate proportional to how far it is from its target follows this exponential curve: a hot drink cooling, a rising battery-voltage reading through a filter, a motor's speed after a throttle change. The RC circuit is the easiest place to see it and measure it.

3. Capacitors and AC: reactance and RC filters

Capacitive reactance

A capacitor passes a changing voltage but not a steady one. The faster the voltage changes, the more current it passes. Starting from $i = C\,dv/dt$ with a sine wave $v = V_p \sin(\omega t)$, where $\omega = 2\pi f$:

```
 i = C dv/dt = C x Vp x w x cos(w t)        so the peak current is  Ip = w C Vp

 The ratio of peak voltage to peak current acts like a resistance, called REACTANCE:

 Xc = Vp / Ip = 1 / (w C) = 1 / (2 x pi x f x C)        (in ohms; falls as f or C rises

Worked examples
 100 nF at 1 kHz:   Xc = 1 / (2 x pi x 1000 x 100e-9) = 1 / 6.283e-4 = 1,592 ohm
 100 nF at 100 kHz: Xc = 15.9 ohm
 100 nF at 50 Hz:   Xc = 31.8 k ohm
```

```
220 uF at 50 Hz:   Xc = 1 / (2 x pi x 50 x 220e-6) = 1 / 0.06912 = 14.5 ohm
At DC (f = 0):     Xc = infinite  (blocks DC)
```

A capacitor in series with a resistor is a frequency-dependent voltage divider. Because the capacitor's current leads its voltage by a quarter of a cycle (90°), the reactance and the resistance do not add directly; they add at right angles, like the sides of a triangle:

The RC low-pass filter

```
Vin o----[ R ]----+----o Vout        LOW-PASS: lets low frequencies through, cuts high
                  |
                =====  C
                  |
0 V  o------------+------o 0 V

Impedance of R and C in series: Z = sqrt(R^2 + Xc^2)
Divider (reading across C):     |Vout / Vin| = Xc / sqrt(R^2 + Xc^2) = 1 / sqrt(1 + (w

Special frequency: where Xc = R, i.e.  w R C = 1:

fc = 1 / (2 x pi x R x C)        the cut-off (corner, -3 dB) frequency

At fc:  |Vout / Vin| = 1 / sqrt(2) = 0.707     (the output is at 70.7 % of the input
                                               amplitude: the power is HALF, called -3 d
        phase of the output lags the input by 45 degrees
Well below fc: gain close to 1.   Well above fc: gain falls as fc / f (10 x the frequen
Link to the time constant:  fc = 1 / (2 x pi x tau)
```

```
Worked example: R = 1.6 k (E24 value), C = 100 nF
 tau = 1600 x 100e-9 = 160 us
 fc  = 1 / (2 x pi x 1600 x 100e-9) = 1 / 1.0053e-3 = 994.7 Hz     (about 1 kHz)

Gain at various frequencies, 1 V peak input: gain = 1 / sqrt(1 + (f/fc)^2)
 100 Hz:   f/fc = 0.1005   gain = 1 / sqrt(1.0101) = 0.995     out = 0.995 V
 995 Hz:   f/fc = 1.000    gain = 0.707                        out = 0.707 V (phase -45
 1 kHz:    w R C = 1.0053  gain = 1 / sqrt(2.0106) = 0.705     out = 0.705 V
 10 kHz:   f/fc = 10.05    gain = 1 / sqrt(102.0)  = 0.099     out = 0.099 V (-20 dB)
```

```
Use: a 1 kHz low-pass removes hum and hiss above 1 kHz from a sensor line, or turns a
 PWM waveform (Basic Electronics 10) into a steady average voltage.
```

The RC high-pass filter

Swap the two parts and the capacitor is in the signal path: it passes fast changes and blocks slow ones, including DC. Taking the output across the resistor,

```
 Vin o-----||-----+----o Vout          HIGH-PASS: blocks DC and low frequencies
          C       |
                [ R ]
                  |
 0 V  o-----------+------o 0 V

 |Vout / Vin| = w R C / sqrt(1 + (w R C)^2)       same fc = 1 / (2 x pi x R x C)

Same R = 1.6 k and C = 100 nF (fc = 995 Hz):
 100 Hz:  gain 0.100      1 kHz:  gain 0.709 (about -3 dB)      10 kHz:  gain 0.995
Use: couple an audio or sensor signal into the next stage while blocking its DC level,
 or detect a sudden edge on an otherwise steady line.
```

FILTER	WHERE C GOES	PASSES	BLOCKS	TYPICAL ROBOT USE
Low-pass	Across the output (to ground)	Slow signals, DC	Fast noise	Smooth a sensor reading before the ADC; average a PWM signal
High-pass	In series with the signal	Fast changes	DC, drift	AC-couple a microphone; detect an edge

4. Decoupling and smoothing

Decoupling: a tiny local reservoir

A microcontroller or motor driver draws current in sudden bursts as its transistors switch, for example 50 mA in 10 ns. The supply wires back to the battery are not perfect: they have inductance (roughly 1 nH per millimetre, so about 100 nH for 10 cm). A rapidly changing current through an inductance produces a voltage, v = L di/dt (section 5):

```
Without a capacitor: the supply droops while the wire resists the change
 v = L x di/dt = 100e-9 x (0.050 / 10e-9) = 100e-9 x 5e6 = 0.5 V

With 100 nF right at the chip, the burst is supplied by the capacitor's own charge:
 dV = I x dt / C = 0.050 x 10e-9 / 100e-9 = 5 mV
(from the definition Q = C V: removing charge I dt lowers V by I dt / C)
```

- **100 nF ceramic** as close as possible to each IC's power pins, with short leads (that is the purpose of the "104" caps on every board).
- **10 to 100 µF electrolytic** (bulk) near a motor driver, regulator or the point where power enters a board: slower, bigger bursts.
- A small and a large capacitor in parallel is useful: each covers a different range of speeds.

Smoothing a rectified supply

A rectifier (Basic Electronics 5) turns AC into a series of humps. A big capacitor fills at each peak and feeds the load between peaks. While feeding the load with a steady current I for one gap of T = 1/f seconds, the voltage falls by dV = I × T / C, so:

```
 ripple  dV = I / (f x C)       (a slight over-estimate, as the capacitor is topped up p

UK mains is 50 Hz; a full-wave rectifier gives humps at 100 Hz. Load 100 mA:
 C = 1000 uF:   dV = 0.1 / (100 x 1e-3) = 1.0 V
 For 0.1 V ripple:  C = I / (f x dV) = 0.1 / (100 x 0.1) = 0.01 F = 10,000 uF

Why switching supplies run at 100 kHz or more: same 100 mA, f = 100 kHz, C = 100 uF:
 dV = 0.1 / (1e5 x 100e-6) = 0.1 / 10 = 0.01 V
A hundred times smaller capacitor does a hundred times better job.
```

5. Inductors

An inductor is a coil of wire, often wound on an iron or ferrite core. Current through it makes a magnetic field; if the current changes, the field changes, and a changing field induces a voltage in the coil that **opposes the change** (Lenz's law). The inductance L is measured in henries (H; practical sizes are µH to mH, up to a few H for relay and motor coils).

```
Definition: the voltage across an inductor follows how fast its current changes
 v = L x di/dt           (a steady current = no voltage: at DC an inductor is just a wir
                          apart from the resistance of the copper)

Energy stored: power = v x i = L i di/dt, so the energy added while the current
 rises from 0 to I is the integral of L i di:
 E = 1/2 x L x I^2
 100 mH carrying 50 mA:  E = 0.5 x 0.1 x 0.05^2 = 1.25e-4 J = 125 uJ

Reactance (same method as for the capacitor, i = Ip sin(w t), v = w L Ip cos(w t)):
 XL = w L = 2 x pi x f x L            (RISES with frequency: the opposite of a capacitor
 10 mH at 50 Hz:  XL = 2 x pi x 50 x 0.01  = 3.14 ohm
 10 mH at 1 kHz:  XL = 62.8 ohm           10 mH at 100 kHz:  6.28 k ohm

Combining (when the coils are far apart, with no magnetic coupling): the OPPOSITE of
 capacitors, the SAME as resistors
 series: L = L1 + L2          parallel: 1/L = 1/L1 + 1/L2
```

	CAPACITOR	INDUCTOR
Stores energy in	Electric field	Magnetic field
Cannot change instantly	Voltage (a sudden step needs infinite current)	Current (a sudden stop needs infinite voltage)
Law	i = C dv/dt	v = L di/dt
Energy	½ C V^2	½ L I^2
At DC (steady state)	Open circuit	Short circuit (a wire)
Reactance	1/(2πfC), falls with f	2πfL, rises with f
Time constant with R	τ = R C	τ = L / R

The RL time constant

```
  Vs o---o  o---[ R ]---( L )---o 0 V       (a coil always has some wire resistance R)

Derivation: Vs = i R + L di/dt  -->  di / (Vs/R - i) = (R/L) dt. Integrating, as for the
 i(t) = (Vs / R) x (1 - e^(-t R / L))        time constant  tau = L / R

Worked example: 5 V, R = 100 ohm, L = 100 mH
```

```
tau = 0.1 / 100 = 1 ms          final current = 5 / 100 = 50 mA
i(1 ms) = 50 x 0.632 = 31.6 mA     i(3 ms) = 50 x 0.950 = 47.5 mA     (95 % after 3 tau

Notice: the inductor makes the current RISE slowly, and a larger R makes it faster
 (the opposite behaviour to a capacitor).
```

Back-EMF and why coils need protection

The law v = L di/dt has a dangerous consequence. When current through a coil is switched off, di/dt is large and *negative*, so the coil produces a large voltage in the direction that tries to keep the current flowing. This is the **back-EMF** (or inductive kick). Think of a heavy wheel: you cannot stop it instantly, and the faster you try, the more force you feel.

```
A 12 V relay coil: R = 240 ohm, L = 240 mH. Running current = 12 / 240 = 50 mA.
 A transistor (Basic Electronics 6) cuts the current to zero in 2 us:
 v = L x di/dt = 0.24 x (0.050 / 2e-6) = 0.24 x 25,000 = 6,000 V      (in theory)

In reality something gives way first: the switch contacts spark, or the transistor
 junction breaks down at tens of volts and is destroyed. The stored energy
 1/2 L I^2 = 0.5 x 0.24 x 0.05^2 = 300 uJ has to go somewhere.

THE FIX: a FLYBACK diode across the coil (arrow pointing at the supply side).
 Normally reverse-biased, so it does nothing; when the switch opens, the coil's current
 can circulate through the diode and fade away harmlessly:

        +12 V o-------+-----------+
                      |           |
                ( L ) coil     ---   flyback diode: cathode (bar) to the +12 V side
                [ R ]          /_\   anode to the switched side
                      |           |
                      +-----------+------o to the switch (transistor) and then 0 V

 With the diode: the switched node goes only about 0.7 V ABOVE +12 V (to 12.7 V), and th
 current decays with tau = L / R = 1 ms, from 50 mA to 5 mA in about 2 ms.
```

ALWAYS PROTECT INDUCTIVE LOADS

Relays, solenoids, motors and any coil switched by a transistor or microcontroller pin need a flyback (freewheel) diode or an equivalent clamp. Without one, the spike can kill the switching transistor in a single operation, or travel back down the supply and reset the microcontroller. The diode itself is

> introduced in Basic Electronics 5, wired up to a transistor switch in Basic Electronics 6, and used with motors, H-bridges and relays in Basic Electronics 9.

A spinning motor adds a second, related effect: it also generates a voltage proportional to its speed, the motor's own back-EMF, which limits its speed and current. Basic Electronics 9 covers both kinds.

6. A switch debounce circuit

A mechanical switch does not close cleanly. For a millisecond or several, the springy contacts bounce, making and breaking the circuit dozens of times. A microcontroller reads every bounce as a separate press, so one press can count as five. The fix is to make the signal change slowly enough that the bounces are smoothed away. An RC circuit does it with two parts. The pull-up resistor from Basic Electronics 3 charges a capacitor on the input pin.

```
+3.3 V o------+
              |
          [ 10 k ]    Rp (pull-up)
              |
              +--------+-----o  to the input pin (reads 1 normally, 0 when pressed)
              |        |
            ===== C  [ 100 ]  Rs        C = 1 uF, Rs = 100 ohm
              |        |
              |      o/ o     button
              |        |
0 V   o-------+--------+

(C is from the pin node to 0 V; Rs and the button are a second path from the pin node t
```

```
Button released: C charges through the pull-up. tau_up = Rp x C = 10,000 x 1e-6 = 10 ms
 The pin must reach the logic-high threshold, about 0.7 x Vcc (2.31 V on 3.3 V):
 t = -tau x ln(1 - 0.7) = 1.204 x 10 ms = 12.0 ms          (a short, harmless delay on r

Button pressed: C discharges through Rs (100 ohm) and the closed contacts. tau_down = 10
 Pin falls to the logic-low threshold, 0.3 x Vcc (0.99 V), in t = 1.204 x 0.1 ms = 0.12 m
 Peak discharge current = 3.3 / 100 = 33 mA (Rs protects the contacts and the capacitor)
 Pressed, steady: pin = 3.3 x 100 / (10,000 + 100) = 0.033 V   (a clear 0)
```

```
How long a bounce gap does it absorb? During a gap (contacts open) C charges for that ti
 from about 0 V. The pin must stay below 0.3 x Vcc:
 t_gap = -tau_up x ln(1 - 0.3) = 0.357 x 10 ms = 3.6 ms
Gaps shorter than 3.6 ms are completely hidden. For longer bounces, increase C:
 Rp = 10 k, C = 2.2 uF gives tau = 22 ms and absorbs gaps up to 0.357 x 22 = 7.85 ms.
```

EVEN BETTER: A SCHMITT-TRIGGER INPUT

A normal input can chatter if the slow RC edge hangs around the threshold. A Schmitt-trigger input (the 74HC14 inverter, and the input of many microcontroller pins) has two thresholds, one for rising and one for falling, so a slow edge produces one clean transition. RC plus Schmitt trigger is the classic hardware debounce. Software debouncing (ignoring changes for about 20 ms after the first one) is the alternative, and Basic Electronics 10 shows both. Basic Electronics 8 covers Schmitt triggers and logic levels properly.

7. Build it: charge, filter, coil

Three stages. Stages A and B use only a multimeter and a stopwatch (a phone is fine) on a breadboard. Stage C, the filters and the inductor, shows things that need an oscilloscope on a real bench (a scope reads signals changing in microseconds and milliseconds; a multimeter cannot follow them), so do those in a simulator. If you own a signal generator and scope, build the filter from the parts list and compare with the table.

Parts list

PART	VALUE	USED IN
R	10 kΩ, 1/4 W (brown-black-orange)	A, B
C1	220 µF electrolytic, 10 V or more	A
C2	100 µF electrolytic, 10 V or more (and 1 µF for the simulator)	B
Rs	100 Ω, 1/4 W (brown-black-brown)	A (discharge), B
Push button	Tactile switch, or two wire ends to touch	B

PART	VALUE	USED IN
Filter parts (simulator, or bench with a signal generator)	1.6 kΩ, 100 nF	C
Coil (simulator)	240 mH, 240 Ω, 12 V; a real relay coil with the same figures	C
Supply	5 V (USB adapter, power module or a board's 5 V pin; 3 AA cells gives 4.5 V and every voltage below scales by 0.9)	A, B
Breadboard, jumper wires, multimeter, stopwatch		A, B

CHECK THE POLARITY

The electrolytic's stripe side is negative and goes to ground (the − rail). Wire it backwards and it will heat and may burst. The positive leg is the longer one.

Stage A: charge and discharge, measured

```
Hole list (one column per connection):
   R:   column 5 row a   -> column 10 row a      (10 k)
   C1:  + leg (long) in column 10 row b, - leg (stripe) in column 12 row b
   J2:  column 12 row c  -> - rail
   J1:  column 5 row b   -> + rail              (the "switch": pull it out and push it i
   Supply + to + rail, supply - to - rail         (connect LAST)
   Measure: red probe on column 10 row e, black probe on the - rail (meter on volts DC)
   Discharge: touch the 100 ohm resistor across the capacitor's legs until the meter rea
```

1. With J1 out, discharge C1 with the 100 Ω resistor and confirm the meter reads about 0 V. Check the supply voltage too.
2. **Charge:** push J1 into the + rail and start the stopwatch at the same moment. Write down the voltage at 1.5 s, 2.2 s, 4.4 s, 6.6 s and 10 s (a phone video of the meter and stopwatch makes this easy).
3. **Discharge:** once at full voltage, move J1 from the + rail to the − rail, restart the stopwatch, and record the voltage at the same times.
4. Estimate the time constant: it is the time when the charge reading reaches 63 % of the supply (3.16 V on 5 V). Divide by R and you have your capacitor's real value.

Stage B: debounce circuit in slow motion

Build the debounce schematic from section 6 with the *100 µF* capacitor, which stretches the 10 ms time constant to 1 s so the meter can follow it. The real-speed version (1 µF) is in the simulator.

```
Hole list:
   J1:   + rail -> column 5 row b
   Rp:   column 5 row a  -> column 10 row a      (10 k, pull-up)
   C2:   + leg column 10 row b, - leg column 12 row b;  J2: column 12 row c -> - rail
   Rs:   column 10 row c -> column 15 row a      (100 ohm)
   Button: one terminal to column 15 row b, the other terminal to the - rail
   Measure: red probe on column 10 row e (the pin node), black probe on the - rail
```

1. Power up and wait about 5 s: the pin node reads the supply (about 5 V).
2. Press and hold the button: the reading drops at once to about 0.05 V (5 × 100 / 10,100).
3. Release and start the stopwatch: the reading climbs back, reaching 63 % (3.16 V) at 1 s and 70 % (3.5 V) at 1.2 s. That 70 % point is where a 5 V logic input would see a "1".

Stage C: simulator only (filters and coil)

Falstad Circuit Simulator (falstad.com/circuit). Use *Draw* to add parts, double-click a part to edit its value, and right-click a node to send it to the scope.

- **RC charging (compare with Stage A):** a 5 V DC source, a switch, a 10 kΩ resistor and a 220 µF capacitor to ground. Scope the capacitor node and click the switch. The 63 % mark must fall at 2.2 s (you may need to slow or speed the simulation with its speed slider).
- **Low-pass and high-pass:** an AC voltage source (set the amplitude to 1 V, frequency to 100 Hz, 1 kHz, then 10 kHz), a 1.6 kΩ resistor and a 100 nF capacitor. Scope the source and the output together and read the output amplitude. Then swap R and C for the high-pass.
- **Coil current:** a 12 V DC source, a 240 Ω resistor and a 240 mH inductor (set the inductance in its edit box) in series, then a switch to ground. Right-click the resistor and choose the scope to show its current. Close the switch: the current should climb towards 50 mA with a 1 ms time constant.
- **Flyback:** in the same circuit, open the switch and look at the voltage at the node between the coil and the switch. Then add a diode with its anode at ground and its cathode at the +12 V end of the coil (across the coil, as in the section 5 schematic), and repeat. Without the diode the node jumps to a very large voltage; with it, the node rises only to about 12.7 V.

Tinkercad Circuits (tinkercad.com): the breadboard, resistor, electrolytic capacitor (it shows its polarity), push button, power supply and multimeter cover stages A and B with the same hole lists. The function generator and oscilloscope components give the filters (set the generator to a sine wave of 1 V amplitude and use the scope's cursors to read the output). Tinkercad has no ideal inductor in its basic parts, so use Falstad for stage C's coil.

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
A: time constant 10 k × 220 µF	2.2 s	1.7 to 2.8 s	2.2 s
A: charging at 2.2 s	3.16 V	2.7 to 3.6 V	3.16 V
A: charging at 4.4 s	4.32 V	4.0 to 4.6 V	4.32 V
A: charging at 10 s	4.95 V	4.8 to 5.0 V (leakage)	4.95 V
A: time to reach 2.5 V	1.52 s	1.2 to 1.9 s	1.52 s
A: discharging at 2.2 s	1.84 V	1.4 to 2.3 V	1.84 V
A: discharging at 6.6 s	0.25 V	0.1 to 0.5 V	0.25 V
B: button pressed (5 V)	0.05 V	0.03 to 0.15 V	0.05 V
B: 1 s after release (tau = 1 s)	3.16 V	2.7 to 3.6 V	3.16 V
B: 1.2 s after release	3.50 V (70 %)	3.0 to 3.9 V	3.50 V
C: low-pass, 100 Hz / 1 kHz / 10 kHz out (1 V in)	0.995 / 0.705 / 0.099 V	Within a few %	0.995 / 0.705 / 0.099 V
C: high-pass, 100 Hz / 1 kHz / 10 kHz out (1 V in)	0.100 / 0.709 / 0.995 V	Within a few %	0.100 / 0.709 / 0.995 V
C: coil current at 1 / 2 / 3 ms (12 V, tau = 1 ms)	31.6 / 43.2 / 47.5 mA	(needs a scope)	31.6 / 43.2 / 47.5 mA
C: coil, final current	50 mA	(measure with the meter, 12 V)	50 mA
C: switched node after opening, with flyback diode	about 12.7 V (0.7 V above the supply)	(needs a scope)	about 12.7 V; without the diode a huge spike

The real readings spread because of tolerance: the capacitor is ±20 % (and electrolytics change with age) and the resistor is ±5 %. The numbers that do not depend on tolerance are the *fractions*: the reading is always 63 % of the supply after one measured time constant, and 99 % after 4.6.

8. Challenges

CHALLENGE 1: CAPACITOR SUMS

(a) A 47 µF capacitor is charged to 12 V: find the charge and the stored energy. (b) A 22 µF and a 47 µF capacitor are connected (i) in parallel and (ii) in series: find each total capacitance. With the series pair across 24 V, find the voltage across each, and say whether the 22 µF part is safe if it is rated at 16 V. (c) A 220 µF capacitor charged to 9 V discharges through 1 kΩ: find the time constant, the starting current, the energy stored, and the time for the voltage to fall to 1 V. Also give its reactance at 50 Hz.

View solution

CHALLENGE 2: DESIGN AN RC LOW-PASS FILTER

Design a low-pass filter with a cut-off close to 500 Hz using a 47 nF capacitor. (a) Calculate the exact resistance, then choose the nearest E24 value and find the actual cut-off. (b) Find the gain (as a ratio of output to input amplitude) at 50 Hz, 500 Hz and 5 kHz. (c) A 5 V step is applied: find the time constant and the time for the output to reach 90 % of its final value, and check it against rise time = 0.35 / fc. If you can, simulate it in Falstad and compare.

View solution

CHALLENGE 3: DEBOUNCE AND THE RELAY COIL

(a) A switch on a 3.3 V input bounces for up to 6 ms. With a 10 kΩ pull-up and the capacitor on the pin as in section 6, choose an E6 capacitor (1, 1.5, 2.2, 3.3, 4.7 µF) so that gaps of 6 ms are hidden (pin stays below 0.3 × Vcc). Give the release delay (to 0.7 × Vcc), and the peak discharge current and time constant with a 100 Ω series resistor. (b) A 12 V relay coil has R = 240 Ω and L = 240 mH. Find the final current, the time constant, the time to reach 90 % of final current, and the stored energy. If the transistor cuts the current in 2 µs, what voltage does v = L di/dt predict? With a flyback diode (ignore its 0.7 V drop), find how long the current takes to fall to 10 %.

View solution

QUICK REFERENCE

- **Capacitor:** Q = C V; E = ½ C V^2; parallel C = C1 + C2; series 1/C = 1/C1 + 1/C2 (smaller capacitor takes the larger voltage share)
- **RC:** τ = R C; charging v = Vs (1 − $e^{-t/\tau}$), discharging v = V0 $e^{-t/\tau}$; 63 % at 1 τ, 95 % at 3 τ, 99 % at 4.6 τ; t = −τ ln(1 − f)
- **Reactance:** Xc = 1/(2πfC); XL = 2πfL; capacitors block DC and pass AC, inductors the opposite
- **RC filter:** fc = 1/(2πRC); gain 0.707 (−3 dB) at fc; low-pass has C to ground, high-pass has C in the signal path
- **Decoupling:** 100 nF ceramic at each IC supply pin, 10 to 100 µF bulk near drivers and regulators; ripple dV = I/(fC)
- **Capacitor care:** electrolytic stripe is negative; rating at least 1.5 to 2 × the working voltage; ±20 % tolerance; "104" = 100 nF
- **Inductor:** v = L di/dt; E = ½ L I^2; τ = L/R, i = (V/R)(1 − $e^{-t/\tau}$)
- **Back-EMF:** switching off a coil makes a big voltage spike; put a flyback diode across every relay, solenoid or motor coil (cathode to the supply side)
- **Debounce:** 10 k pull-up plus 1 µF to ground plus 100 Ω series resistor to the button; hides gaps up to 0.357 × R C

Next, Basic Electronics 5 introduces diodes and LEDs: one-way current, forward voltage, sizing an LED resistor, the flyback and protection diodes promised above, and rectifier basics.

Basic Electronics

Course 1 · Chapter 5 · Diodes and LEDs

Every component so far (resistors, capacitors, inductors) treats current the same way in both directions. A **diode** does not: it lets current through one way and blocks it the other, like a non-return valve in a water pipe. That one trick lets you light an LED, protect a circuit from a battery put in backwards, absorb the spike from a relay coil (the problem raised at the end of Basic Electronics 4), hold a voltage steady, and turn the alternating current from a transformer into the direct current a robot's electronics need. This chapter builds on Ohm's law (Basic Electronics 1), series and parallel rules (Basic Electronics 2), the voltage divider (Basic Electronics 3) and the capacitor and inductor laws (Basic Electronics 4).

1. The diode

One-way current

A diode has two legs: the **anode** (A) and the **cathode** (K). Conventional current flows easily from anode to cathode, which is the way the triangle in the symbol points. The bar is the cathode, and on a real diode a painted band marks that end.

```
Symbol and the real part:

          anode              cathode
    A o-----|>|-----o K           current flows A to K (arrow direction)

     ( A )==[=====|]==( K )      1N4001: the band is on the CATHODE end

  Forward biased (anode more positive than cathode by about 0.6 to 0.7 V or more):
   it conducts, and drops about 0.7 V whatever the current.
   Reverse biased (anode more negative): it blocks, apart from a leakage current
   of nanoamps. Beyond its reverse-voltage rating it breaks down and is destroyed
   (a Zener diode is built to survive this, section 3).

   Water analogy: a spring-loaded check valve. It needs a small push (about 0.7 V) to open,
   then lets everything through; pushed backwards it stays shut.
```

What is inside: the PN junction (briefly)

A silicon diode is two layers of silicon joined together. One layer (P) is doped to have mobile positive "holes"; the other (N) has mobile electrons. Where they meet, electrons and holes cancel and leave a thin **depletion region** with no free carriers, and an electric field across it that acts as a built-in barrier of about 0.6 to 0.7 V for silicon. A forward voltage larger than the barrier pushes carriers across; a reverse voltage widens the barrier and nothing flows. You do not need more detail than this to use diodes; the exact current is described by one equation.

The Shockley equation

```
 I = Is x ( e^( V / (n x Vt) ) - 1 )

 I   diode current (positive = forward)       V   voltage across it, anode minus cathode
 Is  saturation (leakage) current: depends on the part; about 1e-14 A for a small
     silicon signal diode (used below as an illustration; real parts vary from 1e-15 to
 n   ideality factor, 1 to 2 (use 1 here)
 Vt  thermal voltage = k T / q = 1.381e-23 x 300 / 1.602e-19 = 25.85 mV at 300 K (about

Forward, V much greater than Vt: the -1 can be ignored, so  I = Is x e^(V / Vt)
 Reverse, V much less than -Vt: e^(...) goes to 0, so  I = -Is   (just the tiny leakage)

Solve it for the voltage needed to carry a chosen current (take logs):
 V = n x Vt x ln( I / Is )

 I = 1 uA:   V = 0.02585 x ln(1e-6  / 1e-14) = 0.02585 x 18.42 = 0.476 V
 I = 10 uA:  V = 0.536 V
 I = 100 uA: V = 0.595 V
 I = 1 mA:   V = 0.02585 x ln(1e11) = 0.02585 x 25.33 = 0.655 V
 I = 10 mA:  V = 0.714 V
 I = 100 mA: V = 0.774 V

Every TEN-fold increase in current adds only Vt x ln(10) = 59.5 mV. So over a range
 of 100,000 to 1 in current the voltage only moves from 0.48 V to 0.77 V. That is why
 "a silicon diode drops about 0.7 V" is a good rule: the current can change enormously
 while the voltage barely moves.

Small-signal (dynamic) resistance at a given current: rd = n x Vt / I
 at 1 mA: rd = 0.02585 / 0.001 = 25.9 ohm        at 10 mA: 2.6 ohm        at 100 uA: 259
Temperature: the barrier falls about 2 mV for each degree C of warming.
 From 25 C to 75 C (+50 C), the drop falls by about 100 mV.
```

A diode in a circuit: the load-line idea

The diode and its resistor share the supply, and the same current flows through both. You can solve it two ways. The quick way assumes the drop is a fixed V_f (0.7 V). The exact way is repeated guessing against the equation.

```
 5 V o----[ 1 k ]----|>|---- 0 V        1N4148-type diode, forward biased

Quick method (constant drop 0.7 V):
 I = (5 - 0.7) / 1000 = 4.30 mA

Exact method: the resistor says I = (5 - V)/1000, the diode says V = 0.02585 x ln(I / 1e
 Guess V, find I from the resistor, find the V that I needs, repeat:
 guess V = 0.70:  I = 4.300 mA  ->  V = 0.02585 x ln(4.3e-3 / 1e-14) = 0.6925 V
 guess V = 0.6925: I = 4.3075 mA ->  V = 0.6925 V          (settled)

 Answer: V = 0.692 V, I = 4.31 mA. The 0.7 V rule is within 1 % of the current.

Reverse the diode: no current flows, so there is no voltage across the resistor,
 and the whole 5 V appears across the diode (the meter shows 5.00 V across it).
```

Common diodes

PART	KIND	FORWARD DROP (TYPICAL)	MAX CURRENT	MAX REVERSE VOLTAGE	USE
1N4148	Small-signal silicon	0.62 V at 5 mA	200 mA	100 V	Logic, signals, protecting a signal pin; fast
1N4001 to 1N4007	Rectifier silicon	0.9 V at 1 A	1 A	50 V (1N4001) to 1000 V (1N4007)	Power rectifiers, flyback across a relay or small motor
1N5819	Schottky	0.45 V at 1 A	1 A	40 V	Low-loss rectifier, polarity protection, flyback across a motor
BZX55C5V1 (and similar)	Zener	0.7 V forward; 5.1 V in reverse	500 mW	(breakdown is the point)	Voltage reference or clamp (section 3)

PART	KIND	FORWARD DROP (TYPICAL)	MAX CURRENT	MAX REVERSE VOLTAGE	USE
LED	Light-emitting	1.8 to 3.3 V by colour	20 mA	about 5 V only	Indicator (section 2)

THREE RATINGS TO READ ON EVERY DATASHEET

Maximum forward current (average, and a higher surge value), **maximum reverse voltage** (written V_R or PIV, the peak inverse voltage), and the **forward drop** at your working current. Power lost in the diode is $P = V_f \times I$, so a 1 A silicon diode at 0.9 V wastes 0.9 W and gets warm.

2. LEDs and their resistors

A light-emitting diode is a diode whose junction gives out light when current flows. It follows the same exponential curve, but its forward drop is higher (the barrier depends on the semiconductor material, which also sets the colour). The brightness follows the *current*, not the voltage, and the current rises steeply with a tiny voltage change, so an LED must never be connected straight to a supply. A series resistor sets the current.

Forward voltage by colour

COLOUR	TYPICAL V_F AT 10 TO 20 MA	TYPICAL WORKING CURRENT	NOTES
Infrared (remote control)	1.2 V	20 to 100 mA	Not visible; a phone camera shows it
Red	1.8 to 2.0 V	5 to 20 mA	The classic indicator
Yellow / amber	2.0 to 2.1 V	5 to 20 mA	
Green	2.0 to 2.2 V (3.0+ V for the bright "pure" type)	5 to 20 mA	Check the datasheet
Blue, white	3.0 to 3.3 V	5 to 20 mA	Need more than 3.3 V of supply to run properly

The resistor calculation

The resistor is in series with the LED, so it carries the same current I. Kirchhoff's voltage law (Basic Electronics 2) says the supply V_s is shared between the LED drop and the resistor drop; the resistor gets what the LED does not use:

```
 Vs o---[ R ]---|>|---o 0 V           (the LED's triangle points towards 0 V; anode on

 Vs = V(R) + Vf = I x R + Vf      so      R = (Vs - Vf) / I

Then round R UP to the next standard value (E12: 10, 12, 15, 18, 22, 27, 33, 39, 47, 56,
 A slightly higher R gives a slightly lower current, which never exceeds the LED rating.

Example 1: red LED (Vf = 2.0 V), 5 V supply, 10 mA wanted
 R = (5 - 2.0) / 0.010 = 300 ohm  ->  next E12 value: 330 ohm
 Actual current: I = (5 - 2.0) / 330 = 9.09 mA
 Resistor power: P = I^2 R = 0.00909^2 x 330 = 27.3 mW      (a 1/4 W resistor is plenty)
 LED power: 2.0 x 0.00909 = 18.2 mW

Example 2: the same red LED from a 3.3 V logic pin, 8 mA wanted
 R = (3.3 - 2.0) / 0.008 = 162.5 ohm  ->  next E12 value: 180 ohm
 Actual current: 1.3 / 180 = 7.22 mA          (if you choose 150 ohm, I = 8.67 mA: also

Example 3: blue LED (Vf = 3.2 V), 5 V supply, 15 mA wanted
 R = (5 - 3.2) / 0.015 = 120 ohm  (exactly an E12 value)   Actual current 15.0 mA
 On a 3.3 V supply the same LED has only 0.1 V across the resistor. It will be dim or
 off and its brightness will depend on the exact Vf. Use 5 V, or a boost converter.

Why the voltage drop spread matters: the LED's Vf varies +/-0.2 V between samples.
 A SMALL resistor (little headroom) gives a large current error; a bigger headroom
 gives a steady current. At 5 V and 330 ohm: Vf 1.8 V gives 9.7 mA, Vf 2.2 V gives 8.5 m
 At 3.3 V and 180 ohm: the same two LEDs give 8.3 mA and 6.1 mA: a larger spread.
```

```
Same 330 ohm resistor on 5 V, different colours (typical Vf):
   Colour    Vf     Current = (5 - Vf)/330     Brightness
   Red       2.0 V       9.09 mA               good
   Yellow    2.1 V       8.79 mA               good
   Green     2.1 V       8.79 mA               good
   Blue      3.2 V       5.45 mA               fine (blue LEDs are very bright)
Typical "5 mm" indicators are bright at 2 to 5 mA; 20 mA is the maximum and is rarely ne
```

NEVER POWER AN LED WITHOUT A RESISTOR

Without a series resistor the LED's tiny dynamic resistance lets the current rise until the LED, or the pin driving it, burns out. A microcontroller pin typically supplies only 8 to 40 mA in total depending on the board. An LED also blocks only about 5 V in reverse: connect it backwards to more than that and it can fail. The long leg is the anode (+); the flat edge on the case is the cathode (−).

Several LEDs: series or parallel?

```
SERIES: the same current flows through all, the drops add. One resistor serves the chain

 9 V o---[ R ]---|>|---|>|---|>|---o 0 V        3 red LEDs in a chain

 R = (9 - 3 x 2.0) / 0.015 = 3.0 / 0.015 = 200 ohm  -> next E12: 220 ohm
 Actual current: 3.0 / 220 = 13.6 mA. Resistor power 13.6 mA^2 x 220 = 41 mW. Battery dr
 The supply must exceed the total drops: three blue LEDs need 3 x 3.2 = 9.6 V, so 9 V is
 If any LED fails open, the whole chain goes out.

PARALLEL: the same voltage across each. GIVE EACH LED ITS OWN RESISTOR.

 5 V o----+-------+-------+
          |       |       |
        [330]   [330]   [330]
          |       |       |
         |>|     |>|     |>|
          |       |       |
 0 V o----+-------+-------+

 Each branch: (5 - 2.0) / 330 = 9.09 mA.   Total supply current = 3 x 9.09 = 27.3 mA.
 Power from the supply: 5 x 27.3 mA = 136 mW. The LEDs take 3 x 2.0 x 9.09 mA = 54.5 mW
 the rest is lost as heat in the resistors. In the series chain above, the LEDs take 3 x
 9 V (67 %), which is why series is the more efficient layout when the supply is high en

Why not share ONE resistor across parallel LEDs? Their Vf differ slightly. The LED with
 the lowest Vf takes more than its share, heats up, its Vf falls further (the temperatur
 and it hogs still more current, a runaway. Separate resistors force each branch to its
```

LAYOUT	SUPPLY VOLTAGE NEEDED	SUPPLY CURRENT	RESISTORS	BEST FOR
Series	Sum of the Vf values plus headroom	One LED's current	One	High supply voltage (9 V, 12 V), saving battery current
Parallel (own resistors)	One Vf plus headroom	N times one LED's current	One per LED	Low supply voltage (3.3 V, 5 V), independent switching

Multiplexing: lighting many LEDs with few pins

Your eye averages light over about 10 ms. If you switch LEDs on and off faster than about 100 times a second, they look steadily lit, with a brightness proportional to the **average** current. Multiplexing uses this to share the driver pins among several LEDs by turning on just one group at a time and cycling through them quickly. A 4-digit seven-segment display, for example, joins the same segment (a to g) of all four digits together, with one common line per digit.

```
  Four digits take turns, each on for one quarter of the frame:

    time:      0--2.5 ms   2.5--5 ms   5--7.5 ms   7.5--10 ms   (then repeat; 100 frames/
    digit 1:      ON          off         off          off
    digit 2:      off         ON          off          off
    digit 3:      off         off         ON           off
    digit 4:      off         off         off          ON

  duty cycle = 1/4 = 25 %          average current = peak current x duty cycle

 To get the same brightness as a steady 5 mA, each pulse must carry 4 times as much:
  peak = 5 mA / 0.25 = 20 mA      (exactly the LED maximum: do not push higher)
  resistor for a 5 V supply and red LEDs: R = (5 - 2.0) / 0.020 = 150 ohm        (E12 valu
  Perceived brightness is close to the average current, so equal to a steady 5 mA LED.

 Pins saved: 4 digits x 8 segments = 32 LEDs. Driven directly that needs 32 pins; multipl
  it needs 8 segment lines + 4 digit lines = 12.
  One digit's pulse carries up to 8 segments x 20 mA = 160 mA, which is more than a micro
  can supply, so each digit line needs a transistor switch (Basic Electronics 6).
  Basic Electronics 10 drives displays from a microcontroller using this method.
```

3. Zener diodes

If you apply a large enough *reverse* voltage to an ordinary diode it suddenly conducts, and usually dies. A **Zener diode** is designed so that this reverse breakdown happens at a precise voltage (3.3 V, 5.1 V, 9.1 V and so on, written V_Z) and is survivable. Forward, it behaves like an ordinary diode; reversed, it holds its terminals at about V_Z while current flows, as long as you limit that current with a series resistor so that the power stays below the rating.

```
Zener symbol: the diode symbol with a bent bar, wired REVERSED in use:

  cathode (K, bar) on the more positive side, anode (A) towards 0 V

Behaviour at a glance (for a 5.1 V part):
  forward  (A positive):      like any diode, drops about 0.7 V
  reverse, below 5.1 V:       almost no current (a little leakage)
  reverse, at about 5.1 V:    conducts, and the voltage stays near 5.1 V while the
                              current rises from a few mA to tens of mA
So it holds its terminals near Vz, provided a series resistor keeps the current
inside its power rating: Iz(max) = P(rated) / Vz = 0.5 / 5.1 = 98 mA.
```

The shunt regulator

```
Vin o-----[ R ]-----+-----------+-----o Vout (about Vz)
                    |           |
                  K |           |
                    ---         [ RL ]   the load
                   /_\ A        |
                    |           |
0 V  o--------------+-----------+-----o 0 V

Wire the Zener with its cathode (bar) on the +Vout node and its anode on 0 V.
Currents:  I(R) = (Vin - Vz) / R  =  Iz + IL        (the Zener takes whatever the load
```

```
Design: Vin = 8 to 10 V (a run-down 9 V battery), Vz = 5.1 V, load 0 to 20 mA.
 The Zener only regulates while it carries at least about 5 mA (keep Iz_min = 5 mA).

1. The WORST case for keeping the Zener in regulation is the lowest Vin and the biggest
 R_max = (Vin_min - Vz) / (IL_max + Iz_min) = (8 - 5.1) / (0.020 + 0.005) = 2.9 / 0.025 =
 Round DOWN (a smaller R gives more current, which keeps Iz above its minimum): E24 valu

2. Check both ends with R = 110 ohm:
```

```
Vin = 8 V, full load:  I(R) = 2.9 / 110 = 26.4 mA  -> Iz = 26.4 - 20 = 6.4 mA   (above !
Vin = 10 V, NO load:   I(R) = 4.9 / 110 = 44.5 mA  -> Iz = 44.5 mA  (the worst case for

3. Power ratings:
Zener, worst case: Pz = Vz x Iz = 5.1 x 0.0445 = 227 mW     -> use a 500 mW Zener
Resistor, worst case: PR = I^2 x R = 0.0445^2 x 110 = 218 mW -> use a 1/2 W resistor (1,

4. Reality check:
At 9 V and full load: I(R) = 3.9 / 110 = 35.5 mA, so the input power is 9 x 0.0355 = 31
and the output power is 5.1 x 0.020 = 102 mW: efficiency 102 / 319 = 32 %.
The rest is heat, and with no load the Zener still burns its full current.
Load regulation: the Zener has a small dynamic resistance (about 10 to 30 ohm for a 5.1
so Vout moves by a few tens of mV when the load changes. Fine for a reference, poor for
```

WHEN TO USE A REGULATOR IC INSTEAD

Use a Zener for a reference voltage or a small current. For a robot's main 5 V or 3.3 V supply use a linear regulator (such as the 7805 or an LDO) or a switching converter, introduced in Basic Electronics 9. They hold the voltage far better, handle hundreds of milliamps, and waste far less current than a Zener with its resistor.

Zener as a clamp

```
A signal that may go up to 5 V is going into a 3.3 V input pin. Clamp it with a 3.3 V Z

signal o----[ 1 k ]----+-----o to the 3.3 V input pin
                       |
                     K |
                       ---       3.3 V Zener, cathode (bar) to the input node, anode to
                      /_\ A
                       |
0 V    o---------------+-----o 0 V

Signal at 5 V:   current = (5 - 3.3) / 1000 = 1.7 mA; the pin sees 3.3 V.  Zener power
Signal at 3 V:   the Zener does not conduct (Vz is 3.3 V); the pin sees the signal.
A real 3.3 V Zener already leaks a little below 3.3 V and the voltage is only +/-5 % ac
critical work use a divider or a level-shifter (logic levels are covered in Basic Elect
```

4. Schottky diodes

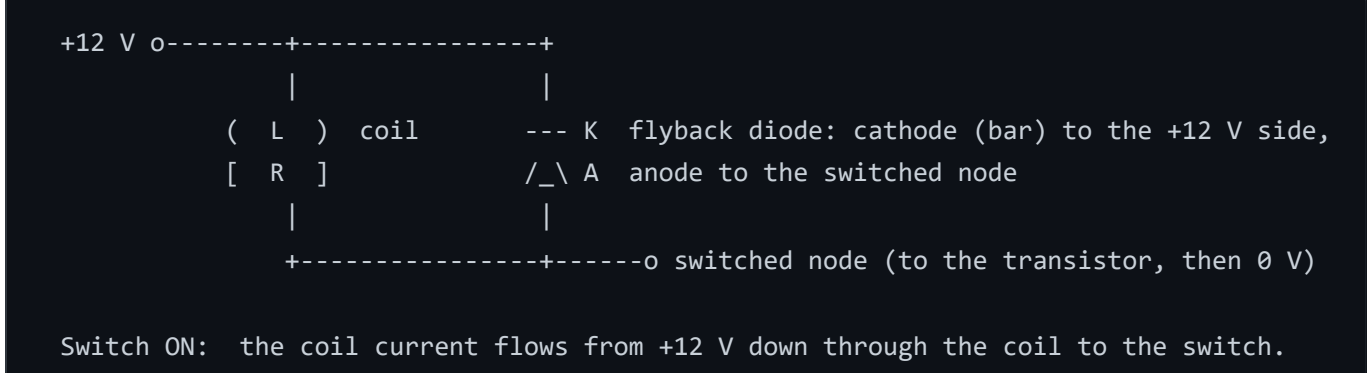
A Schottky diode joins metal to silicon instead of two types of silicon. That gives a lower barrier (a forward drop of about 0.2 to 0.45 V instead of 0.6 to 0.9 V) and a very fast switch-off, because no stored charge has to be cleared. The price is a larger reverse leakage current and a lower reverse-voltage rating, typically 20 to 60 V for common parts.

	SILICON (1N4001)	SCHOTTKY (1N5819)
Forward drop at 1 A	about 0.9 V	about 0.45 V
Power lost at 1 A	0.9 W	0.45 W
Switching speed	Slow (microseconds)	Very fast (nanoseconds)
Reverse leakage	about 5 µA or less	up to a few mA when hot
Max reverse voltage	50 to 1000 V	20 to 60 V (common types)
Best for	Mains rectifiers, high voltage	Low-voltage supplies, battery circuits, flyback in switching converters, protection when every tenth of a volt counts

5. Flyback and protection diodes

Flyback (freewheel) diode: the fix promised in Basic Electronics 4

Basic Electronics 4 showed that switching off the current through a coil makes a huge voltage, v = L di/dt, that can destroy the switching transistor. The flyback diode gives the coil's current somewhere to go. Take the same 12 V relay coil as before: R = 240 Ω, L = 240 mH, running current 50 mA.



```
              The diode has +12 V on its cathode and 0 V on its anode: reverse biased, it
 Switch OFF: the coil keeps pushing the same current the same way, downward through the
              It leaves the coil's bottom end, which swings positive, and returns to the
              through the diode: coil bottom -> diode anode -> diode cathode -> coil top.
              The switched node can only rise to about 12 V + 0.7 V = 12.7 V.
```

In that loop the diode's drop and the coil's own resistance use up the energy. With a clamp voltage V_c across the loop and the coil's resistance R, the current decays according to L di/dt = −(iR + V_c):

```
 Solving (same method as the RL circuit in Basic Electronics 4, starting at I0 = 50 mA):

   i(t) = (I0 + Vc/R) x e^(-t R / L) - Vc/R        tau = L / R = 0.24 / 240 = 1 ms

 Case 1: ideal diode (Vc = 0):   i(t) = 50 mA x e^(-t / 1 ms)       10 % (5 mA) after 2.
 Case 2: real diode (Vc = 0.7 V):
   Vc / R = 0.7 / 240 = 2.92 mA
   i(t) = 52.92 mA x e^(-t / 1 ms) - 2.92 mA
   10 % of 50 mA = 5 mA when e^(-t/tau) = (5 + 2.92) / 52.92 = 0.1496, so t = 1.90 ms
   current reaches zero at t = 1 ms x ln(1 + 50 / 2.92) = 1 ms x ln(18.1) = 2.90 ms

The trade-off: the diode keeps the current flowing for about 3 ms, so a relay is slower
 (and a motor slower to stop braking). To release faster, let the voltage across the coi
 little: put a Zener in series with the flyback diode (the Zener is reverse-biased in th
 its cathode faces the diode's cathode)

 Clamp = Zener 24 V + diode 0.7 V:  Vc = 24.7 V
   Vc / R = 24.7 / 240 = 102.9 mA
   current reaches zero at t = (L/R) ln(1 + I0 R / Vc) = 1 ms x ln(1 + 12 / 24.7) = 1 ms
   10 % (5 mA) at t = 0.349 ms
 The switching transistor now sees 12 + 24.7 = 36.7 V when it turns off, so choose one r
 comfortably above that (60 V or more), not a general-purpose 40 V part.
```

FIT THE DIODE THE RIGHT WAY ROUND

A flyback diode fitted backwards short-circuits the supply through the diode the moment the switch turns on, because the diode is then forward biased across the coil's supply. It will overheat, blow, or damage the supply. Cathode (band) to the **positive** side of the coil, anode to the switched side. For a coil switched by a transistor, almost any 1N4001 or 1N4148 suffices for a small relay (the diode must carry the coil current, 50 mA here, at the moment of switching). For motors use a 1N4001 to 1N4007

or a Schottky such as the 1N5819. Wiring this to a transistor is in Basic Electronics 6, and relays and motors are in Basic Electronics 9.

Protecting an input pin

Two small diodes (1N4148 or a Schottky such as the BAT85) can clamp a signal pin to the supply rails. One goes from the pin up to +V (anode on the pin) so the pin cannot go more than about 0.7 V above the supply; the other from 0 V up to the pin (cathode on the pin, anode on 0 V) so it cannot fall below −0.7 V. Put a 1 kΩ resistor in the signal line to limit the current through whichever diode conducts. Most microcontroller pins already have these diodes built in, but only for a few milliamps.

Reverse-polarity protection

```
Method 1: a diode in series with the supply (anode on the battery side)

 battery +  o----|>|----o +V to the circuit
 battery -  o-------------o 0 V

 Connected correctly the diode conducts; the circuit sees the supply minus the diode dro
 Connected backwards the diode blocks and nothing happens.
   9 V battery, circuit draws 100 mA:
     silicon 1N4001: drop 0.7 V   circuit sees 8.3 V   power lost 0.7 x 0.1 = 70 mW
     Schottky 1N5819: drop 0.3 V  circuit sees 8.7 V   power lost 0.3 x 0.1 = 30 mW
   12 V and a 500 mA load:
     silicon (0.75 V at 0.5 A): 11.25 V delivered; lost 0.375 W (3.1 % of the 12 V x 0.5
     Schottky (0.40 V at 0.5 A): 11.6 V delivered; lost 0.20 W

Method 2: a diode ACROSS the supply, after a fuse (the shunt, or crowbar, type)

 battery + o----[ fuse ]----+-----o +V to the circuit
                            |
                      K  ---     diode across the supply: cathode (bar) to +, anode to
                         /_\ A
                            |
 battery - o----------------+-----o 0 V

 Correct polarity: the diode is reverse biased and does nothing, and there is no voltage
 Reversed battery: the diode conducts, takes the current and blows the fuse in a few mil
 and the circuit sees only about -0.7 V. Needs a diode that can pass the fuse's blowing
 Method 3 (an advanced one) uses a P-channel MOSFET with almost no drop: Basic Electroni
```

6. Rectifiers: AC to DC

A transformer or an AC adapter delivers alternating voltage, which rises and falls through zero (Basic Electronics 1). Electronics needs a one-way voltage. A diode passes only the positive half-cycles, which is rectification. Remember the AC maths: a sine wave of RMS voltage V_{rms} has peak $V_{pk} = V_{rms} \times \sqrt{2} = 1.414\ V_{rms}$. A 9 V_{rms} source peaks at 12.73 V.

Half-wave rectifier

```
AC o----|>|----+----+-----o  +out
            A  K   |    |
                =====  [ RL ]       smoothing capacitor C and load RL, both to 0 V
                   |    |
AC o---------------+----+-----o  0 V (the other AC lead)

Only positive half-cycles pass. The output is a series of humps, one per cycle (50 per
Peak output   = Vpk - Vf     = 12.73 - 0.7 = 12.03 V   for 9 Vrms
Average (no capacitor, resistive load) = Vpeak / pi = 12.03 / 3.14159 = 3.83 V  (wastef
```

Full-wave bridge rectifier

Four diodes in a diamond use both half-cycles, with the output current always going the same way through the load. Name the four corners AC1, AC2, P (positive output) and N (negative output, 0 V).

```
                  P  (+out)
                 / \
        D1      /   \      D2          D1: anode AC1, cathode P
  (AC1 to P)   /     \   (AC2 to P)    D2: anode AC2, cathode P
              /       \                D3: anode N,   cathode AC1
AC1 o--------+         +--------o AC2   D4: anode N,   cathode AC2
              \       /
        D3     \     /     D4
  (N to AC1)    \   /   (N to AC2)
                 \ /
                  N  (-out, 0 V)

The diodes D1 and D2 point UP towards P; D3 and D4 point UP from N towards the AC corne
The load (and the smoothing capacitor) go between P and N.
```

```
Trace the current to check there is no short circuit:
 AC1 positive (AC2 negative):  AC1 -> D1 -> P -> load -> N -> D4 -> AC2        (D1 and D4
 AC2 positive (AC1 negative):  AC2 -> D2 -> P -> load -> N -> D3 -> AC1        (D2 and D3
 In both cases the current goes P to N through the load, so the output is always positive
 D3 and D1 are in series and point the same way (N to AC1 to P), as are D4 and D2, so no
 back to N runs along the diodes: the AC source can never short P to N.

 Two diodes conduct at a time, so:    Vout peak = Vpk - 2 x Vf = Vpk - 1.4 V
 The output humps come at TWICE the AC frequency (100 per second on 50 Hz mains).
```

Smoothing capacitor and ripple

Between the humps the output falls to zero. A capacitor across the load (Basic Electronics 4, "Smoothing a rectified supply") fills at each peak and feeds the load in the gaps. The load current I discharges it for roughly one hump period, T = $1/f_r$, so from Q = C V the voltage falls by Vr = I × T / C:

```
 ripple (peak to peak)  Vr = I / (f_r x C)       f_r = ripple frequency (hump rate)
 mean output voltage    Vdc = Vpeak - Vr / 2

Bridge from the 9 Vrms source with C = 470 uF and a 1 kohm load:
 Vpeak = 9 x 1.4142 - 1.4 = 12.73 - 1.4 = 11.33 V
 Load current I = 11.2 / 1000 = 11.2 mA                (use the mean voltage, near 11.2
 f_r = 100 Hz
 Vr = 0.0112 / (100 x 470e-6) = 0.0112 / 0.047 = 0.238 V
 Vdc = 11.33 - 0.119 = 11.21 V                            (the output swings between 11.(

Half-wave from the same source, same C and load: f_r = 50 Hz, Vpeak = 12.73 - 0.7 = 12.0
 I = 11.8 / 1000 = 11.8 mA                          (mean output near 12.03 - 0.25 = 11.8
 Vr = 0.0118 / (50 x 470e-6) = 0.0118 / 0.0235 = 0.50 V  -> the ripple is more than DOUB

Rule of thumb: for a bridge on 50 Hz mains, allow about 1000 uF per 100 mA for a ripple
 Check: 0.1 / (100 x 1e-3) = 1.0 V.

Choose C for a target ripple: C = I / (f_r x Vr)
 200 mA load, bridge, 0.5 V ripple:  C = 0.2 / (100 x 0.5) = 4 mF = 4000 uF  -> use 4700
 Voltage rating: the capacitor sees Vpeak. 12.73 V peak needs a 25 V part (at least 1.5
```

	HALF-WAVE	FULL-WAVE BRIDGE
Diodes	1	4
Peak output	Vpk − 0.7 V	Vpk − 1.4 V
Ripple frequency	f (50 Hz)	2f (100 Hz)
Ripple for the same C and I	Twice as large	Half
Diode peak reverse voltage	Up to 2 Vpk with a capacitor	Vpk
Used for	Tiny supplies, signal detectors	Power supplies of all sizes

SAFETY: NEVER BUILD THIS FROM THE MAINS

Do not connect a breadboard to mains voltage, ever. The rectifier here is built from a sealed low-voltage AC source: a 9 to 12 V AC plug-in adapter (check the label says "AC" output), a small bell transformer, or a signal generator in a simulator. Measure the adapter's real AC output first with the meter on the AC range: an unregulated AC adapter reads well above its label (for example 12 V instead of 9 V) when lightly loaded.

DIODE RATINGS FOR A RECTIFIER

Each diode needs a reverse rating above the peak voltage (12.7 V: any 1N4001 at 50 V is fine), an average current rating at least equal to the load current (in a bridge each diode carries half the load current on average), and a surge rating for the moment the empty capacitor is first charged, when the current is large for a few cycles. The surge is higher the larger the capacitor and the lower the resistance of the transformer.

7. Build it: LEDs, diodes, Zener and rectifier

Four stages. Stages A to C use a breadboard, a 5 V or 9 V supply and a multimeter. Stage D, the rectifier, shows its ripple only on an oscilloscope, so it is a simulator job, with an optional real version if you own an AC adapter and a scope. (A multimeter on the AC range shows the input; on the DC range it shows the mean output.)

Parts list

PART	VALUE	USED IN
LEDs	Red, green, and blue or white (5 mm)	A
R1	330 Ω, 1/4 W (orange-orange-brown)	A
R2	1 kΩ, 1/4 W (brown-black-red)	B, and the 1 kΩ load in C and D
Diodes	1N4148, 1N4001 and 1N5819 (a Schottky); four more 1N4001 for D	B, D
Zener	5.1 V, 500 mW (BZX55C5V1 or similar)	C
R3	220 Ω, 1/2 W (red-red-brown)	C
Loads for C	1 kΩ (R2) and 330 Ω (R1)	C
C	470 µF electrolytic, 25 V	D
Supply	5 V for A and B; a 9 V battery (PP3) for C; 9 V_{rms} AC (adapter) for D	A to D
Breadboard, jumper wires, multimeter		A to D

Stage A: LEDs and the resistor calculation, measured

```
Hole list (one column per connection):
   R1:   column 5 row a   -> column 10 row a      (330 ohm)
   LED:  anode (long leg) column 10 row b, cathode (flat side, short leg) column 12 row
   J2:   column 12 row c  -> - rail
   J1:   column 5 row b   -> + rail
   Supply + to + rail, supply - to - rail         (connect LAST)
   Measure LED voltage:      red probe column 10 row e, black probe on the - rail
   Measure resistor voltage: red probe column 5 row e, black probe column 10 row d
   (current = resistor voltage / 330; or put the meter in series on the 10 mA range betw
```

1. Check the supply reads 5 V. Power up with the red LED.
2. Record V(LED) and V(R1). Calculate I = V(R1) / 330 and compare with the table.
3. Swap in the green, then the blue or white LED, and repeat.

4. *Try backwards:* turn the LED round for a moment: it stays dark and the meter across it reads nearly the supply (about 5 V), because nothing flows.

Stage B: diode drops and reverse blocking

```
Hole list:
   R2:    column 5 row a   -> column 10 row a      (1 k)
   Diode: anode (plain end) column 10 row b, cathode (band) column 12 row b
   J2:    column 12 row c  -> - rail
   J1:    column 5 row b   -> + rail
   Supply 5 V (connect LAST)
   Measure: red probe column 10 row e (diode anode), black probe on the - rail  ->  the
   Current = (5 - Vf) / 1000. Repeat for 1N4148, 1N4001, 1N5819.
   Then reverse the diode and measure again.
```

Stage C: Zener regulator

```
Hole list (9 V battery; R3 = 220 ohm):
   R3:     column 5 row a   -> column 10 row a
   Zener:  cathode (band) column 10 row b, anode (plain end) column 12 row b     (REVERS
   J2:     column 12 row c  -> - rail
   J1:     column 5 row b   -> + rail
   Load:   one resistor from column 10 row c to column 12 row d   (try: none, 1 k, then
   Battery + to + rail, battery - to - rail     (connect LAST)
   Measure: red probe column 10 row e (the output node), black on the - rail
```

1. No load: record Vout. The Zener current is (9 − 5.1) / 220 = 17.7 mA, power 5.1 × 0.0177 = 90 mW.
2. Add the 1 kΩ load (5.1 mA): the Zener current falls to 12.6 mA; Vout should barely change.
3. Swap to the 330 Ω load (15.5 mA): the Zener is left with only 17.7 − 15.5 = 2.3 mA, which is below the knee; Vout will sag a little. Watch it happen.

Stage D: bridge rectifier

```
Hole list (four 1N4001; the band is the CATHODE). Nodes: AC1 = column 5, P = column 8, A
N = column 14, AC1-extension = column 17 (joined to column 5 by a jumper).
   D1:   anode column 5 row a,  cathode (band) column 8 row a       (AC1 to P)
```

```
D2:   anode column 11 row a, cathode (band) column 8 row b       (AC2 to P)
D4:   anode column 14 row a, cathode (band) column 11 row b      (N to AC2)
D3:   anode column 14 row b, cathode (band) column 17 row a      (N to AC1, via the e
Jumper: column 5 row c -> column 17 row b
AC input: one lead to column 5 row d, the other lead to column 11 row d
C:    + leg column 8 row c, - leg (stripe) column 14 row c       (stripe on N, the 0
Load: 1 k column 8 row d -> column 14 row d
Measure: DC volts, red probe column 8 row e (P), black probe column 14 row e (N)
Measure the AC input with the meter on AC volts across column 5 row e and column 11 r
```

Simulators

Falstad Circuit Simulator (falstad.com/circuit): in the *Circuits* menu pick *Diodes* and the bridge-rectifier example, or draw it: an AC voltage source (amplitude 12.73 V for 9 V_{rms}, frequency 50 Hz), four diodes, a 470 µF capacitor and a 1 kΩ load. Right-click P and choose scope to see the humps and the ripple. Note that Falstad's default diode model has a slightly larger forward drop (about 0.8 V rather than 0.7 V), so expect the output about 0.2 V lower than the calculated value, and its LED has a drop of about 2 V by default. The same program has the Zener (add a *Zener diode* and edit its breakdown voltage) and the half-wave version (delete three diodes).

Tinkercad Circuits (tinkercad.com): the breadboard, resistor, red LED (the LED edit box shows its current), diode, Zener, electrolytic capacitor, battery and multimeter cover stages A to C with the same hole lists. For D use the function generator and oscilloscope: set a sine at 12.7 V peak (check whether its amplitude is a peak or peak-to-peak value; if its range is too small, scale every voltage down by the same factor) and 50 Hz.

- **Flyback (simulator):** take the 12 V, 240 Ω, 240 mH circuit from Basic Electronics 4 stage C. Without a diode the switched node spikes; with the 1N4001 flyback diode, anode on the switched node and cathode on +12 V, it stays near 12.7 V and the current reaches 5 mA about 1.9 ms after opening. Add a 24 V Zener in series with the diode (Zener cathode facing the diode's cathode) and the current dies in about 0.35 ms, with the node reaching about 36.7 V.
- **Reverse polarity:** put a 1N4001 in series with a 9 V source and a 100 Ω load, then swap the supply polarity. Forward: about 8.3 V at the load; reversed: 0 V.

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
A: red LED voltage / current (5 V, 330 Ω)	2.0 V / 9.1 mA	1.8 to 2.1 V / 8.8 to 9.7 mA	about 2.0 V / 9 mA
A: green LED voltage / current	2.1 V / 8.8 mA	1.9 to 2.3 V / 8.2 to 9.4 mA	about 2.0 V / 9 mA
A: blue or white LED voltage / current	3.2 V / 5.5 mA	2.9 to 3.3 V / 5.2 to 6.4 mA	about 3.0 to 3.2 V
A: LED reversed	0 mA, 5 V across it	about 5.0 V across it	5 V across it
B: 1N4148 drop (1 k, 5 V)	0.69 V (4.3 mA)	0.6 to 0.72 V	0.65 to 0.8 V
B: 1N4001 drop (1 k, 5 V)	0.69 V (4.3 mA)	0.62 to 0.72 V	0.65 to 0.8 V
B: 1N5819 Schottky drop (1 k, 5 V)	about 0.25 to 0.3 V (4.7 mA)	0.2 to 0.35 V	0.2 to 0.4 V
B: any diode reversed	5.00 V across the diode	4.99 to 5.00 V	5 V
C: Zener, no load (9 V, 220 Ω)	5.1 V (17.7 mA)	4.9 to 5.3 V	5.1 V
C: Zener, 1 kΩ load	5.1 V (Iz 12.6 mA)	4.9 to 5.3 V	5.1 V
C: Zener, 330 Ω load	5.1 V ideal (Iz only 2.3 mA)	4.6 to 5.0 V (below the knee)	5.0 to 5.1 V
D: bridge peak output (9 V_{rms}, 470 µF, 1 kΩ)	11.33 V	depends on the adapter	11.1 to 11.3 V (see the note)
D: ripple (peak to peak)	0.24 V	0.2 to 0.3 V (scope)	0.24 V
D: mean DC output	11.2 V	meter on DC	11.0 to 11.2 V
D: half-wave, same parts: ripple / peak	0.50 V / 12.03 V	(needs a scope)	0.50 V / 11.8 to 12.0 V
Flyback: time to 5 mA, 24.7 V clamp vs 0.7 V diode	0.35 ms vs 1.9 ms	(needs a scope)	0.35 ms vs 1.9 ms

The real readings spread because diodes and LEDs vary from sample to sample (V_f changes by ±0.1 to 0.2 V) and with temperature. The things that do not vary are the relationships: the LED

current is $(V_s - V_f)/R$, a silicon diode drops about 0.7 V, a Schottky drops less than half of that, and the bridge ripple is half the half-wave ripple.

8. Challenges

CHALLENGE 1: LED RESISTORS

(a) A green LED (V_f = 2.1 V) is run from 5 V at about 12 mA: find the resistor, round it up to E12, then find the actual current, the resistor power and the LED power. (b) A red LED (1.9 V) on a 3.3 V logic pin at about 8 mA: find the resistor and the current with the E12 value above it; then find the current for the LED with V_f = 1.7 V and for V_f = 2.1 V. (c) Four red LEDs (2.0 V each) in series on 12 V at 15 mA: find R (E12 up), the real current and the resistor power; could you also run four *blue* LEDs (3.2 V) on the same supply? (d) A 4-digit display is multiplexed so each digit is lit one quarter of the time: find the peak current needed for a 5 mA average, the resistor for a 5 V supply with red LEDs, and the time each digit is on at a 100 Hz frame rate.

View solution

CHALLENGE 2: ZENER REGULATOR AND POLARITY PROTECTION

(a) Design a Zener shunt regulator giving 5.1 V from a 12 V supply that varies from 11 V to 13 V, for a load of 0 to 30 mA, keeping at least 5 mA through the Zener. Find the resistor (round down to E24: 150, 160, 180, 200 Ω), check the Zener current at both extremes, find the worst-case power in the Zener and in the resistor, and the efficiency at 12 V and full load. (b) A 12 V supply feeds a 500 mA load through a reverse-polarity series diode. Compare a 1N4001 (0.75 V at 0.5 A) with a 1N5819 (0.40 V at 0.5 A): load voltage, power lost in the diode and the diode's share of the total power, and say which is better and why.

View solution

CHALLENGE 3: RECTIFIER AND FLYBACK

(a) A transformer gives 12 V_{rms}. A bridge rectifier (0.7 V per diode) with a 2200 µF capacitor feeds a 200 mA load: find the peak voltage, ripple, mean output, minimum output, and the voltage rating you would choose for the capacitor. Give the average current and peak reverse voltage per diode. (b) Repeat with a single-diode half-wave rectifier and the same capacitor and load, and say how the ripple compares. (c) For the capacitor size that gives 0.5 V ripple with the bridge at 200 mA, find C and the

next E6 value up. (d) A relay coil (12 V, 240 Ω, 240 mH, 50 mA) is switched off. With a 0.7 V flyback diode find the time for the current to fall to 5 mA; then with a 12 V Zener in series with the diode (clamp 12.7 V) find the time to reach zero current and the voltage the transistor sees.

View solution

QUICK REFERENCE

- **Diode:** conducts anode to cathode (band = cathode); silicon drops about 0.7 V, Schottky 0.2 to 0.45 V, LED 1.8 to 3.3 V; reverse blocks until breakdown
- **Shockley:** $I = Is\,(e^{V/(nVt)} - 1)$, Vt = 25.85 mV at 300 K; $V = nVt\,\ln(I/Is)$; +59.5 mV per tenfold current; $rd = nVt/I$
- **LED resistor:** $R = (Vs - Vf)/I$, rounded *up* to E12; typical 5 to 20 mA; always one resistor per parallel branch
- **Series LEDs:** supply > sum of Vf; **multiplexing:** average current = peak × duty cycle, refresh at 100 Hz or more
- **Zener regulator:** $R = (Vin_min - Vz)/(IL_max + Iz_min)$, rounded down; check $Pz = Vz \times Iz$ at no load and $PR = I^2R$
- **Flyback diode:** across every coil, cathode to the supply side; $i(t) = (I0 + Vc/R)\,e^{-tR/L} - Vc/R$; a Zener in series speeds the release
- **Reverse-polarity protection:** a series diode (loses Vf, Schottky is better) or a shunt diode after a fuse
- **Rectifiers:** $Vpk = 1.414\,Vrms$; half-wave peak Vpk − 0.7, bridge Vpk − 1.4; ripple $Vr = I/(f_r\,C)$ with f_r = f (half-wave) or 2f (bridge); $Vdc = Vpeak - Vr/2$

Next, Basic Electronics 6 introduces transistors as switches: using a weak signal to switch a lamp or motor, base and gate resistors, and fitting the flyback diode from this chapter across the load.

Basic Electronics

Course 1 · Chapter 6 · Transistors as Switches

A microcontroller pin can supply a few milliamps at 3.3 V or 5 V. A motor wants hundreds of milliamps at 6 V or 12 V, a relay coil wants 50 mA at 12 V, an LED strip wants amps. The **transistor** bridges the gap: a tiny current or voltage at one terminal controls a much larger current between the other two, so a weak signal can switch a heavy load. Used as a switch it is either fully off or fully on, and that is the only way of using it in this chapter (the half-on, amplifying use comes in Basic Electronics 7). The chapter builds on Ohm's law and power (Basic Electronics 1), series circuits (Basic Electronics 2), voltage dividers and pull-down resistors (Basic Electronics 3), the gate-charging time constant (Basic Electronics 4) and the flyback diode (Basic Electronics 5).

1. The idea: a switch you control with electricity

A transistor has three terminals. Two of them carry the load current; the third is the control. Put it in series with the load and it behaves like a switch whose contacts are operated by the control terminal. The ideal switch has an *off* state with no current and an *on* state with no voltage drop. A real transistor leaves a small drop in the on state (the cost, which turns into heat) and a tiny leakage in the off state (usually negligible).

	BIPOLAR JUNCTION TRANSISTOR (BJT)	MOSFET
Terminals (control first)	Base (B), Collector (C), Emitter (E)	Gate (G), Drain (D), Source (S)
Load current flows	Collector to emitter (NPN)	Drain to source (N-channel)
Controlled by	A small **current** into the base	A **voltage** on the gate (almost no current once charged)
On-state loss	A fixed drop, $V_{CE(sat)}$, about 0.1 to 0.3 V	A resistance, $R_{DS(on)}$, from milliohms to a few ohms
Two polarities	NPN and PNP	N-channel and P-channel

```
Reading the symbols in a schematic:

BJT:    the emitter leg carries an ARROW.
        NPN: the arrow points OUT, away from the base  (mnemonic: "Not Pointing iN").
        PNP: the arrow points IN, towards the base.
        The arrow shows the direction of conventional current in that leg.

MOSFET: the gate is drawn as a plate, separate from the channel (it is insulated: no ga
        once the gate has charged). An arrow on the source/body leg points IN for N-cha
        Most real MOSFETs have a built-in body diode from source to drain (N-channel) t

In this chapter's diagrams each transistor is shown as a labelled box with its three pi

         C                D
         |                |
B o------[Q1 NPN]   G o----[Q2 N-MOSFET]
         |                |
         E                S
```

Three regions, two of them useful

REGION	BJT CONDITION	WHAT THE TRANSISTOR DOES	USE
Cut-off	V_{BE} below about 0.5 V, no base current	No load current: the switch is OFF	Switch off
Active	Base current flows, but not enough to saturate	$I_C = h_{FE} \times I_B$: acts as a current amplifier; V_{CE} in the middle	Amplifiers (Basic Electronics 7), **not** switching
Saturation	So much base current that the load, not the transistor, limits I_C	V_{CE} falls to $V_{CE(sat)}$, about 0.2 V: the switch is ON	Switch on

2. The BJT as a switch

Current gain, h_{FE}

For an NPN transistor, a current I_B flowing into the base (and out of the emitter) lets a much larger current I_C flow into the collector. In the active region the two are proportional:

```
IC = hFE x IB          IE = IC + IB = (hFE + 1) x IB          (the emitter carries both)

hFE (also written beta) is the DC current gain. For a 2N3904 the datasheet gives hFE fr
at 10 mA, but only a MINIMUM of about 60 at 50 mA, and it varies with temperature and f
sample by 2:1 or more. A design must work with the minimum, not the typical.

Base-emitter voltage: the base-emitter junction is a diode (Basic Electronics 5):
  VBE is about 0.65 to 0.7 V in the active region and up to about 0.8 to 0.95 V in hard
  Use 0.8 V for the base resistor sums below.
```

Saturation and the base resistor

With the load connected, the supply and the load resistance cap the collector current: it can never exceed what the load allows when the transistor is a perfect closed switch. Call that the **saturation current**:

```
IC(sat) = (Vs - VCE(sat)) / RL                  the most the load can draw

The minimum base current that just reaches saturation is IB(min) = IC(sat) / hFE(min).
But at that point the transistor is only just on: VCE(sat) is still falling, hFE is a g
transistor has a different hFE. So deliberately supply MUCH more base current: choose a
(IC(sat) divided by the base current you supply) of about 10:

IB = IC(sat) / 10                          forced beta = 10
RB = (Vdrive - VBE(sat)) / IB              Vdrive = the logic HIGH level of the pin; V

Then round RB DOWN to a standard value (more base current is safe; less may not saturat
The datasheet quotes VCE(sat) at a base current of one tenth of the collector current,
is also the condition under which the datasheet's VCE(sat) is true.
```

Worked example 1: a 6 V, 50 mA load from a 5 V pin

A 120 Ω resistor stands in for a 6 V relay coil (or any 50 mA load). The pin is a 5 V output (an Arduino Uno or Nano is the natural board here: 5 V logic and a pin that can supply up to 20 mA comfortably).

```
                          +6 V
                           |
                       [ R2 120 ]    load (a 6 V, 50 mA relay coil stand-in)
```

```
                                |
                                C
                                |
pin o---[ R1 820 ]---+----------B   Q1  2N3904 (NPN)
                     |          |
                [ R3 10 k ]     E
                     |          |
0 V o----------------+----------+

R3 is optional: it keeps the base at 0 V while the pin is floating (during a reset), so

1. Saturation current (VCE(sat) = 0.2 V):
   IC(sat) = (6 - 0.2) / 120 = 48.3 mA
2. Base current for forced beta 10:
   IB = 48.3 / 10 = 4.83 mA
3. Base resistor:
   RB = (5 - 0.8) / 0.00483 = 869 ohm  ->  next standard value DOWN: 820 ohm
4. Check with R1 = 820 ohm:
   current in R1 = (5 - 0.8) / 820 = 5.12 mA   (the pin supplies this; the Uno limit is
   through R3: 0.8 / 10000 = 0.08 mA     into the base: 5.12 - 0.08 = 5.04 mA
   forced beta = 48.3 / 5.04 = 9.6                      (against a minimum hFE of 60 a
5. Power:
   transistor: P = VCE(sat) x IC = 0.2 x 0.0483 = 9.7 mW            (TO-92 limit is ab
   load:       P = IC^2 x RL = 0.0483^2 x 120 = 0.28 W              (a 1/2 W resistor,
   base resistor: 0.00512^2 x 820 = 21.5 mW
6. Voltages with the switch ON:   load gets 6 - 0.2 = 5.8 V.   OFF: the load gets 0 V a
```

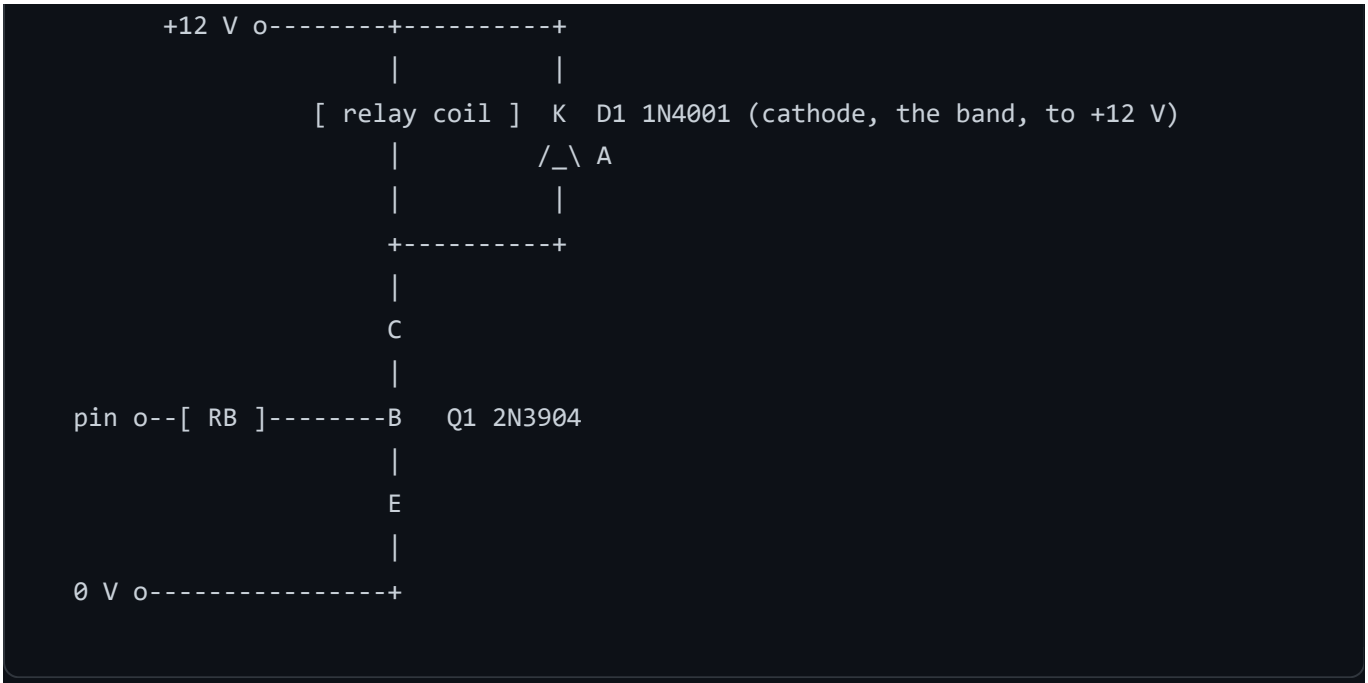
Worked example 2: a 12 V relay from a 3.3 V pin

The relay coil is 12 V, 240 Ω (50 mA). A 3.3 V board (a Raspberry Pi Pico or an ESP32) cannot supply as much base current as an Uno: the Pico defaults to 4 mA per pin (12 mA at most when configured), and about 10 mA is a sensible ceiling on an ESP32.

```
IC(sat) = (12 - 0.2) / 240 = 49.2 mA
forced beta 10:  IB = 4.92 mA,   RB = (3.3 - 0.8) / 0.00492 = 508 ohm  ->  470 ohm
  check: IB = 2.5 / 470 = 5.3 mA, forced beta 9.2        (OK on an ESP32; above the Pic

If the pin cannot give 5 mA, relax the forced beta. The datasheet minimum hFE at 50 mA
so forced beta 20 still leaves a safety factor of 3:
RB = 1 kohm:  IB = 2.5 / 1000 = 2.5 mA,  forced beta = 49.2 / 2.5 = 19.7        (fine f

The coil needs the flyback diode from Basic Electronics 5 (cathode to +12 V, anode to t
```



TIE THE GROUNDS TOGETHER

The 3.3 V board and the 12 V supply are two separate supplies. The transistor's emitter must connect to the **common 0 V** shared by both. With no shared ground there is no complete path for the base current and nothing switches, or it switches erratically.

What goes wrong if the base current is too small

```
Same circuit as example 1, but R1 = 22 kohm (and no R3), 5 V drive:

  IB = (5 - 0.7) / 22000 = 0.195 mA
  Sample with hFE = 100:  IC would be 100 x 0.195 mA = 19.5 mA
       load drop = 19.5 mA x 120 = 2.35 V,   VCE = 6 - 2.35 = 3.65 V      NOT saturated
       transistor power = 3.65 x 0.0195 = 71 mW,  load only gets 2.35 V (a relay may no
  Sample with hFE = 300:  IC would be 58.6 mA, more than the 48.3 mA the load allows, s

The same circuit behaves differently with different transistors and at different temper
a design that works only for a lucky hFE is a bad design. Forced beta 10 removes the de

Why the half-on state is dangerous for the transistor, with a resistive load RL on supp
  P = VCE x IC = VCE x (Vs - VCE) / RL
  This is largest when VCE = Vs / 2:   Pmax = Vs^2 / (4 RL) = 36 / 480 = 75 mW  here.
Fine for a 2N3904, but a 12 V, 2 ohm load would give 144 / 8 = 18 W: the part would bur
A switch must pass through the half-on region quickly and spend its time fully off or f
```

When the BJT cannot do it: the Darlington

Suppose the load is a 6 V motor drawing 400 mA. Forced beta 10 asks for I_B = 40 mA, far more than a microcontroller pin can give. Raising the forced beta needs a transistor with far more gain. A **Darlington pair** puts two transistors in one package, with the first one's emitter feeding the second one's base, so the gains multiply:

```
Q1 (small) feeds the base of Q2 (power). Overall gain = hFE1 x hFE2.     Example: 100 x
The TIP120 guarantees hFE of at least 1000 (a ratio of 250 or more is easy to meet).

Two junctions in series:    VBE(total) = 0.7 + 0.7 = 1.4 V   (use 1.4 V in the base-res
Q1 can never saturate Q2 fully, because Q1's collector sits one VBE above Q2's emitter:
  VCE(sat) = VCE1(sat) + VBE2 = 0.2 + 0.7 = about 0.9 V   (datasheets give 1 to 2 V at

Example: 6 V motor, 400 mA, 5 V pin, TIP120.   Forced beta 250.
  IB = 400 / 250 = 1.6 mA
  RB = (5 - 1.4) / 0.0016 = 2250 ohm  ->  2.2 kohm   (IB = 3.6 / 2200 = 1.64 mA)
  Motor voltage = 6 - 0.9 = 5.1 V     (15 % of the supply is lost)
  Heat in the transistor = 0.9 x 0.4 = 0.36 W      (a TO-220 is fine alone up to about
  A logic-level MOSFET with RDS(on) of 25 milliohm in the same job loses 0.4^2 x 0.025
```

PNP, briefly

A PNP transistor is the mirror image: the emitter connects to the *positive* supply, the load sits between collector and 0 V, and the transistor turns on when the base is pulled about 0.7 V *below* the emitter, with current flowing *out* of the base. That makes it a high-side switch (section 4). With a logic signal the trouble is the off state: if the emitter is at +12 V and the pin's HIGH level is only 5 V (or 3.3 V), the base is still 7 V (or 8.7 V) below the emitter and the PNP stays **on**. A PNP is therefore easy to use only when the logic level equals the supply; otherwise use the level-shifter circuit in section 6.

3. The MOSFET as a switch

How a MOSFET switches

In an **N-channel enhancement MOSFET** the gate is a metal plate separated from the silicon by a thin layer of oxide. A positive voltage on the gate (relative to the **source**) pulls electrons into a channel under the plate, connecting drain to source. No current flows into the gate itself (apart

from the brief charge of its capacitance), so the pin supplies nothing once the gate is charged. A **P-channel** device is the mirror image: its source goes to the positive supply and it turns on when the gate is pulled *below* the source.

```
N-channel:  VGS = VG - VS   must be POSITIVE and above the threshold to turn on.   Sour
P-channel:  VGS = VG - VS   must be NEGATIVE:  VGS = -10 V means the gate is 10 V BELOW

Two numbers decide everything for a switch:
1. Gate threshold voltage VGS(th): the VGS at which the drain current is only a tiny te
   (typically 250 microamps). It is the voltage where the part just begins to conduct.
   at which it is fully on.
2. On-resistance RDS(on): the drain-source resistance when fully on, and it is quoted A
   (for example 25 milliohm at VGS = 5 V). Look for the test voltage in the datasheet.

Small-signal model for a MOSFET in the ohmic (switch) region, for small VDS:
   ID = k x (VGS - VGS(th)) x VDS     so     RDS(on) = VDS / ID = 1 / ( k x (VGS - VGS(

k is a constant of the device. The overdrive, VGS - VGS(th), sets how low RDS(on) can g
double the overdrive and RDS(on) roughly halves. (Real parts follow this loosely: use i
datasheet, not to predict it.)

Illustration, taking VGS(th) = 1.5 V:  RDS(on) at VGS = 3.3 V is about 3.5 / 1.8 = 1.9
  Taking a standard-level part with VGS(th) = 3 V, rated 44 milliohm at 10 V: at VGS =
  0.3 V instead of 7 V, so the model gives 44 mohm x 7 / 0.3 = about 1 ohm. And the wor
```

Standard-level versus logic-level

A **logic-level** MOSFET is designed so that RDS(on) is low with only 4.5 V (or 2.5 V) on the gate, so a microcontroller pin can switch it directly. A **standard-level** MOSFET is specified at VGS = 10 V and needs a gate-driver stage if driven from 3.3 V or 5 V. Never choose by threshold voltage alone: look for the RDS(on) quoted at your own gate voltage.

PART	TYPE	V_{DS} MAX	I_D MAX (CONTINUOUS)	$V_{GS(TH)}$	$R_{DS(ON)}$ (AT THE V_{GS} SHOWN)	SUITS
IRLZ44N	N, logic-level	55 V	47 A	1 to 2 V	25 mΩ at 5 V; 22 mΩ at 10 V	5 V logic (Arduino Uno), motors and strips to a few amps
IRLB8721	N, logic-level	30 V	62 A	1.35 to 2.35 V	16 mΩ at 4.5 V; 8.7 mΩ at	3.3 V to 5 V logic, low-voltage supplies (3.3 V is

PART	TYPE	V_{DS} MAX	I_D MAX (CONTINUOUS)	$V_{GS(TH)}$	$R_{DS(ON)}$ (AT THE V_{GS} SHOWN)	SUITS
					10 V	at the edge: test it)
IRF540N	N, standard	100 V	33 A	2 to 4 V	44 mΩ at 10 V	Needs a 10 V gate drive; not for a microcontroller pin
2N7000	N, small signal	60 V	0.2 A	0.8 to 3 V	about 5 Ω at 10 V	Small loads on 5 V (relay at 50 mA); not for 3.3 V gates
BSS138	N, small signal	50 V	0.22 A	0.8 to 1.5 V	about 3.5 Ω at 10 V	Small loads at 3.3 V; the level-shifter in section 6
IRF9540N	P, standard	−100 V	−23 A	−2 to −4 V	117 mΩ at −10 V	High-side switch on 12 V (section 4)

These are typical datasheet figures; always read the datasheet of the exact part you buy, since brands differ and the gate-voltage test condition is what matters. Parts for lower voltages (an SMD such as the AO3400A, with $R_{DS(on)}$ of about 50 mΩ quoted at 2.5 V) exist for 3.3 V boards.

Losses: conduction and heat

```
In the ON state the MOSFET is just a resistor:    VDS = ID x RDS(on)      P = ID^2 x RD

Motor: 6 V, 0.25 A running, 1.2 A when stalled, IRLZ44N with 5 V on the gate (RDS(on) =
  running: VDS = 0.25 x 0.025 = 6.25 mV      P = 0.25^2 x 0.025 = 1.6 mW
  stalled: VDS = 1.2  x 0.025 = 30 mV        P = 1.2^2  x 0.025 = 36 mW
  The motor gets within 6 mV of the full supply and the MOSFET stays cold. Compare the

The same motor through a standard-level part on 3.3 V (using the 1 ohm model estimate a
  running: VDS = 0.25 x 1.03 = 0.26 V     P = 64 mW        stalled: VDS = 1.2 x 1.03 =
  Stalled, it would run hot, and on a cool-gate sample it might not turn on at all.

Heat: temperature rise = P x thermal resistance. A TO-220 in free air is about 62 degre
RDS(on) also rises with temperature (it roughly doubles from 25 C to 150 C), so check t
```

Gate resistor, pull-down and gate capacitance

The gate is a capacitor (C_{iss}, the input capacitance, a few nanofarads for a power MOSFET). To change state the pin must charge or discharge it, which causes a short pulse of current and takes a short time. Two resistors manage this:

```
pin o---[ Rg 220 ]---+---o G
                     |
                [ Rp 10 k ]           Rg: series GATE RESISTOR       Rp: gate PULL-DOWN
                     |
0 V o----------------+---o S

Rg limits the charging current. Without it the pin sees almost a short circuit for an i
  Peak gate current = V / Rg = 5 / 220 = 22.7 mA     (inside a pin's 20 to 40 mA limits
                                                     internal resistance of about 25 oh
  With Rg = 0 the peak would be limited only by that internal resistance: about 200 mA,
  Rg also damps ringing that a long wire's inductance would cause (Basic Electronics 4)

Rp holds the gate at 0 V while the pin is not driving it (reset, boot, a cut wire), so
A floating gate acts as an antenna; a hand near the wire can turn the MOSFET on. Rp mus
or reduce the gate voltage (it forms a divider with Rg):
  Rp = 10 kohm:  gate = 5 x 10000 / (10000 + 220) = 4.89 V    current wasted 5 / 10220 =
  at 3.3 V:      gate = 3.3 x 10000 / 10220 = 3.23 V
  Rp = 1 kohm:   gate = 5 x 1000 / 1220 = 4.10 V     current wasted 4.1 mA      (too lo
```

```
Switching speed. The gate charges through Rg, like the RC circuit in Basic Electronics 4

  IRLZ44N input capacitance Ciss about 3.3 nF,  Rg = 220 ohm:
  tau = Rg x Ciss = 220 x 3.3e-9 = 0.73 microseconds
  10 % to 90 % rise time = 2.2 x tau = 1.6 microseconds

The datasheet's total gate charge gives a second estimate. IRLZ44N: Qg is about 48 nC a
half the starting value (11 mA on average) fills it in  t = Q / I = 48e-9 / 0.0114 = ab
The real value is of this order, a few microseconds.

Consequences for PWM (pulse width modulation, covered in Basic Electronics 10):
  at 20 kHz the period is 50 microseconds, so a 4 to 5 microsecond edge is under 10 % o
  at 500 kHz the period is 2 microseconds: the gate would never reach its full voltage.

The MOSFET dissipates most heat during the edges, when both VDS and ID are large. A rou
the 6 V, 0.25 A motor at 20 kHz, edges of 5 microseconds each:
  P(switching) = 0.5 x V x I x (t_rise + t_fall) x f = 0.5 x 6 x 0.25 x (10e-6) x 20000
```

```
That is about 100 times the conduction loss (1.6 mW). A higher PWM frequency or a lar
slower edges and a higher frequency both mean more heat.
```

The N-channel motor switch

```
+6 V o----------------------------+-------------+
                                  |             |
                                ( M )           ---  K   D1 1N5819: cathode (band) to +6 V
                                  |             /_\ A
                                  |              |
                                  +--------------+
                                  |
                                  D
pin o--[ Rg 220 ]-----+----------G    Q2  IRLZ44N (N-channel)
                      |           S
                 [ Rp 10 k ]      |
                      |           |
0 V o-----------------+-----------+

Trace it: pin HIGH (5 V) charges the gate; current flows +6 V -> motor -> drain -> sour
Pin LOW: the MOSFET opens. The motor's current (it is an inductor, Basic Electronics 4)
lower end, passes up through D1 (anode to cathode) and back into the top end: the loop

Without D1:  v = L di/dt.   An illustrative 2 mH motor winding with 0.25 A falling to z
  2e-3 x 0.25 / 1e-6 = 500 V     (far above the 55 V rating: the MOSFET avalanches and
With D1:     the drain rises only to 6 + the diode drop (about 6.3 V).
```

4. Low-side versus high-side switching

You can put the switch on either side of the load. The load works in both cases, but the choice affects what is wired to what.

```
LOW-SIDE (switch between the load and 0 V):       HIGH-SIDE (switch between +V and the

  +V o------+                                       +V o------+
            |                                         |
         [ LOAD ]                                  [ SWITCH ]
            |                                         |
         [ SWITCH ]                                [ LOAD ]
```

```
          |                                          |
   0 V o-----+                              0 V o-----+

Low-side: the control pin is referenced to 0 V, the same 0 V as the logic. An N-channel
does it directly with VGS = the pin's HIGH level. This is the easy and usual way. Disad
bottom terminal sits at +V, not 0 V. If the load is also wired to ground by anything el
current can find another path and the load is "off" in name only.

High-side: the load's bottom terminal is always at 0 V, which suits loads that must sha
with their own connector, a motor on a metal chassis). The difficulty is the control:
  P-channel MOSFET: source to +V. VGS = VG - VS must be negative. To turn OFF, the gate
       A 3.3 V pin can only reach 3.3 V, which with a 12 V source leaves VGS = -8.7 V:
       high-side switch on 12 V needs a level shifter, a small NPN that pulls the gate
  N-channel MOSFET on the high side: its source is the load's top terminal, which rises
       VGS = VG - VS needs about 5 V ABOVE the source, so the gate must reach +V + 5 V:
       Done with a dedicated gate-driver chip with a charge pump (the high-side drivers
```

	LOW-SIDE	HIGH-SIDE
Device	NPN or N-channel MOSFET	PNP or P-channel MOSFET (or an N-channel with a gate driver)
Drive from a 3.3 V	Direct (logic-level MOSFET) or	Needs an inverting level shifter when +V is above the
Load ground	Floats up to +V when off	Always at 0 V
Typical use	Motors, relays, lamps, LED strips	Power rails that must be cut cleanly, reverse-polarity protection, loads with a fixed ground

5. Driving real loads

LOAD	BEHAVIOUR TO PLAN FOR	WORKED CHECK	SWITCH AND PROTECTION
Relay coil (12 V, 240 Ω)	Inductive (L = 240 mH): needs the flyback diode of Basic Electronics 5; draws a steady 50 mA	Section 2, example 2: RB = 470 Ω from 3.3 V	2N3904 (up to 200 mA) or a 2N7000 or BSS138 on a gate; 1N4001 across the coil
DC motor (6 V, 0.25 A run, 1.2 A stall)	Inductive and generates back-EMF; the start-up and stall currents are 3 to 5 times the	Section 3: RDS(on) 25 mΩ loses 36 mW stalled	Logic-level MOSFET; Schottky flyback diode (1N5819) across the motor

LOAD	BEHAVIOUR TO PLAN FOR	WORKED CHECK	SWITCH AND PROTECTION
	running current; size the switch for the *stall*		
Filament lamp (6 V, 0.1 A)	Cold tungsten has about one tenth of its hot resistance, so the **inrush** is about 10 times the running current for several tens of milliseconds	Hot R = 60 Ω, cold about 6 Ω: inrush up to 6 / 6 = 1 A	The inrush (1 A) is five times a 2N3904's 200 mA rating: use a MOSFET. No flyback diode needed (a lamp is not inductive).
LED strip (12 V, 2 m at 9.6 W/m)	Resistive, with the resistors already built in; you only dim it with PWM	I = 19.2 W / 12 V = 1.6 A. IRLZ44N: VDS = 1.6 × 0.025 = 40 mV, P = 64 mW	Logic-level N-channel MOSFET; no flyback diode; at 3.3 V use a part with RDS(on) quoted at 2.5 V (AO3400A: $1.6^2 \times 0.05 = 0.13$ W)

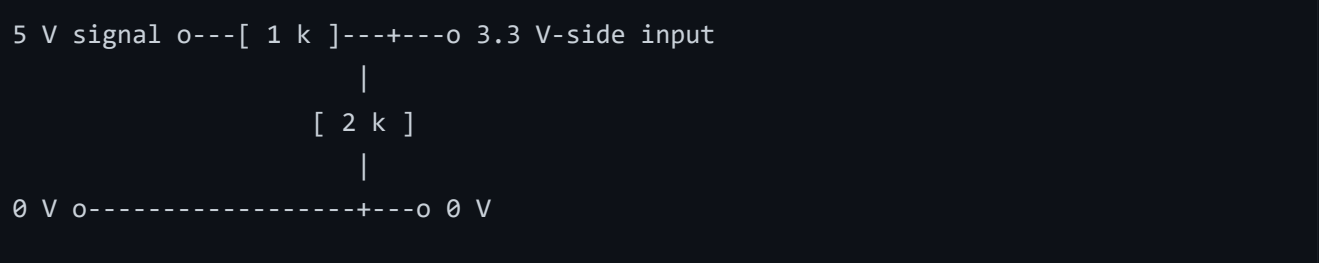
CHOOSING THE BOARD FOR EACH JOB

Relay from 5 V: an Arduino Uno or Nano, pin current 20 mA, plenty for a BJT base resistor. Motor or LED strip from 3.3 V: an ESP32 or Raspberry Pi Pico driving a logic-level MOSFET, because a MOSFET gate needs almost no pin current, which suits boards with weak pins. A Raspberry Pi (the full computer) has 3.3 V pins, only a few milliamps each, and no protection if you overload one: always use a transistor between its pin and any load, a gate resistor on a MOSFET, and never connect 5 V back to a pin. Basic Electronics 11 compares the boards in full.

6. Level shifting: when the signal and the load disagree

Logic levels differ: a Pico and an ESP32 output 3.3 V, an Uno outputs 5 V, a motor supply may be 12 V. A signal at one level can be too small to drive a part that works at another (the PNP and P-channel examples above), or too large and destroy an input (5 V into a 3.3 V pin). Level shifting converts between them. Three simple ways:

1. A resistor divider (for going down, 5 V to 3.3 V)



```
Vout = 5 x 2 / (1 + 2) = 3.33 V        current = 5 / 3000 = 1.67 mA
Speed: the divider's output resistance is 1 k || 2 k = 667 ohm. With 20 pF of pin and w
tau = 667 x 20e-12 = 13 ns: fast enough for ordinary signals (up to about a megahertz).
Works for one direction only, high to low, and the divider cannot raise a level.
```

2. A bidirectional MOSFET shifter (3.3 V and 5 V sides, such as I2C)

```
+3.3 V o-------+------------+           +5 V o-------+
               |            |                       |
           [ 10 k ]         G                   [ 10 k ]
               |            |                       |
LV  o----------+-----S [Q4 BSS138] D-----------------+--------o HV

Q4's gate is tied to +3.3 V; its source is on the 3.3 V (LV) side, its drain on the 5 V
  Both sides idle high:   VGS = 3.3 - 3.3 = 0 V, Q4 is off, LV = 3.3 V, HV = 5 V (throu
  LV side pulled low:     VGS = 3.3 V, Q4 turns on, and pulls the HV side low too.
  HV side pulled low:     current first flows through Q4's body diode, taking LV to abo
                          Q4 turns on and pulls LV to 0 V. A low on either side appears
This is the standard breakout for I2C buses. For fast one-way signals use a buffer chip
threshold is low enough for 3.3 V when it is powered from 5 V). Note an Uno's input nee
it usually works but with little margin.)
```

3. An NPN shifter (for going UP to a higher voltage, inverting)

A transistor does the job of the NPN switch in section 2 with the load replaced by a pull-up resistor to the higher supply. The collector node swings 0 V to the higher supply, and the signal is *inverted* (the pin HIGH gives a LOW at the collector). Stage C of the build uses this to drive a P-channel high-side switch on 12 V from a 3.3 V or 5 V pin.

7. BJT or MOSFET?

	BJT (2N3904, BC337, TIP120)	MOSFET (IRLZ44N, BSS138, IRF9540N)
Control	Needs a base **current**: 1/10 of the load current	Needs a gate **voltage**; no steady current
Load current range	Small: up to about 0.2 A (2N3904); 0.8 A (BC337); more with a Darlington	Milliamps to dozens of amps

	BJT (2N3904, BC337, TIP120)	MOSFET (IRLZ44N, BSS138, IRF9540N)
On-state loss	Fixed $V_{CE(sat)}$ of 0.2 V (about 1 V for a Darlington): power = $V_{CE(sat)} \times I$	$R_{DS(on)}$ of milliohms: power = $I^2 R_{DS(on)}$
Works from a 3.3 V pin	Yes, if you can spare the base current	Only if it is logic-level and quoted at 3.3 V or lower
Switching speed	Fast; storage time slows turn-off a little	Fast, limited by gate charge; gate driver for MHz
Static damage	Robust	The gate oxide dies above about ±20 V; static can kill it; keep Rp fitted
Cost, parts count	Cheapest, one resistor	A little dearer; gate resistor and pull-down resistor
Best for	Small loads (LEDs, relays, buzzers) of up to 100 to 200 mA; level shifters	Motors, strips, lamps, anything over about 0.3 A; PWM; 3.3 V logic; high efficiency; battery-powered builds

RULE OF THUMB

Under about 100 mA, use a BJT: one resistor and a 5 cent part. Above about 300 mA, or with PWM, use a logic-level MOSFET. In between, check how much heat each design makes: $P = V_{CE(sat)} \times I$ for the BJT or $I^2 R_{DS(on)}$ for the MOSFET.

8. Build it: switch a load from a weak signal

Three stages, each switching a load that a pin cannot drive directly. A is the BJT; B is the project of the chapter, a motor switched by a logic-level MOSFET; C is a high-side switch on 12 V from a 3.3 V or 5 V signal, with a level shifter (simulator first, real parts optional).

TWO SUPPLIES, ONE GROUND

Use a separate supply for the load (4 AA cells for 6 V, a 12 V adapter for stage C) and a 5 V logic supply (a USB breakout or the 5 V pin of an Arduino). Join all the 0 V rails with a jumper wire. Connect each supply **last**, and disconnect the battery before changing the wiring. A stalled motor or a short through a MOSFET can draw amps: never run a motor from the USB port of a computer.

Parts list

PART	VALUE	USED IN
Q1	2N3904 NPN (TO-92)	A, C
Q2	IRLZ44N logic-level N-MOSFET (TO-220)	B
Q3	IRF9540N P-MOSFET (TO-220)	C
R1	820 Ω, 1/4 W (grey-red-brown)	A
R2	120 Ω, 1/2 W (brown-red-brown), the load	A
R3, Rp, R6, R8	10 kΩ, 1/4 W (brown-black-orange)	A (R3), B (Rp), C (R6, R8)
Rg	220 Ω, 1/4 W (red-red-brown)	B
R7	4.7 kΩ (yellow-violet-red) for 5 V drive; 2.2 kΩ (red-red-red) for 3.3 V	C
R9	470 Ω, 1/2 W (yellow-violet-brown), the load	C
D1	1N5819 Schottky (or a 1N4001)	B
Motor	Small 6 V hobby DC motor (about 0.25 A running, about 1.2 A stalled)	B
Supplies	5 V logic supply; 6 V (4 AA cells) for A and B; 12 V adapter for C	A to C
Breadboard, jumper wires, multimeter		A to C

Stage A: the BJT switch (6 V, 50 mA load)

```
Use the top rails for the 6 V supply (top + is 6 V) and the bottom rails for the 5 V log
Hole list (one column per connection):
   Q1 2N3904, flat face towards you, legs down:   E column 10 row a,  B column 11 row a,
   (check the datasheet or the meter's diode test for your brand; some 2N3904s have a di
   R1:  column 5 row a   -> column 11 row b          (820 ohm: signal to base)
   R3:  column 11 row c  -> bottom - rail            (10 k: base to 0 V)
   Emitter: column 10 row b -> bottom - rail
   R2:  column 12 row b  -> column 16 row b          (120 ohm load)
   Jumper: column 16 row c -> top + rail (6 V)
```

```
    Signal jumper: column 5 row b -> bottom + rail (5 V = ON)  or  bottom - rail (0 V = O
    Join: top - rail -> bottom - rail                  (common 0 V)
    Connect the supplies LAST.
    Measure ON:   Vce: red probe column 12 row e, black on the - rail
                  Vload: red column 16 row e, black column 12 row e   (load current = Vlo
                  Vbe: red column 11 row e, black column 10 row e
                  Base current: Ib = (V across R1) / 820, measured red column 5 row e, bl
```

1. With the signal on 0 V (OFF): V_{load} should be 0 V and V_{CE} should be 6 V.
2. Move the jumper to 5 V (ON): V_{CE} falls to about 0.1 to 0.3 V; the load has about 5.8 V across it (48 mA).
3. Measure V_{R1} (about 4.2 V); calculate I_B and the forced beta.
4. *Try the half-on case:* replace R1 with 22 kΩ. V_{CE} will be somewhere around 2 to 5 V and will

action.

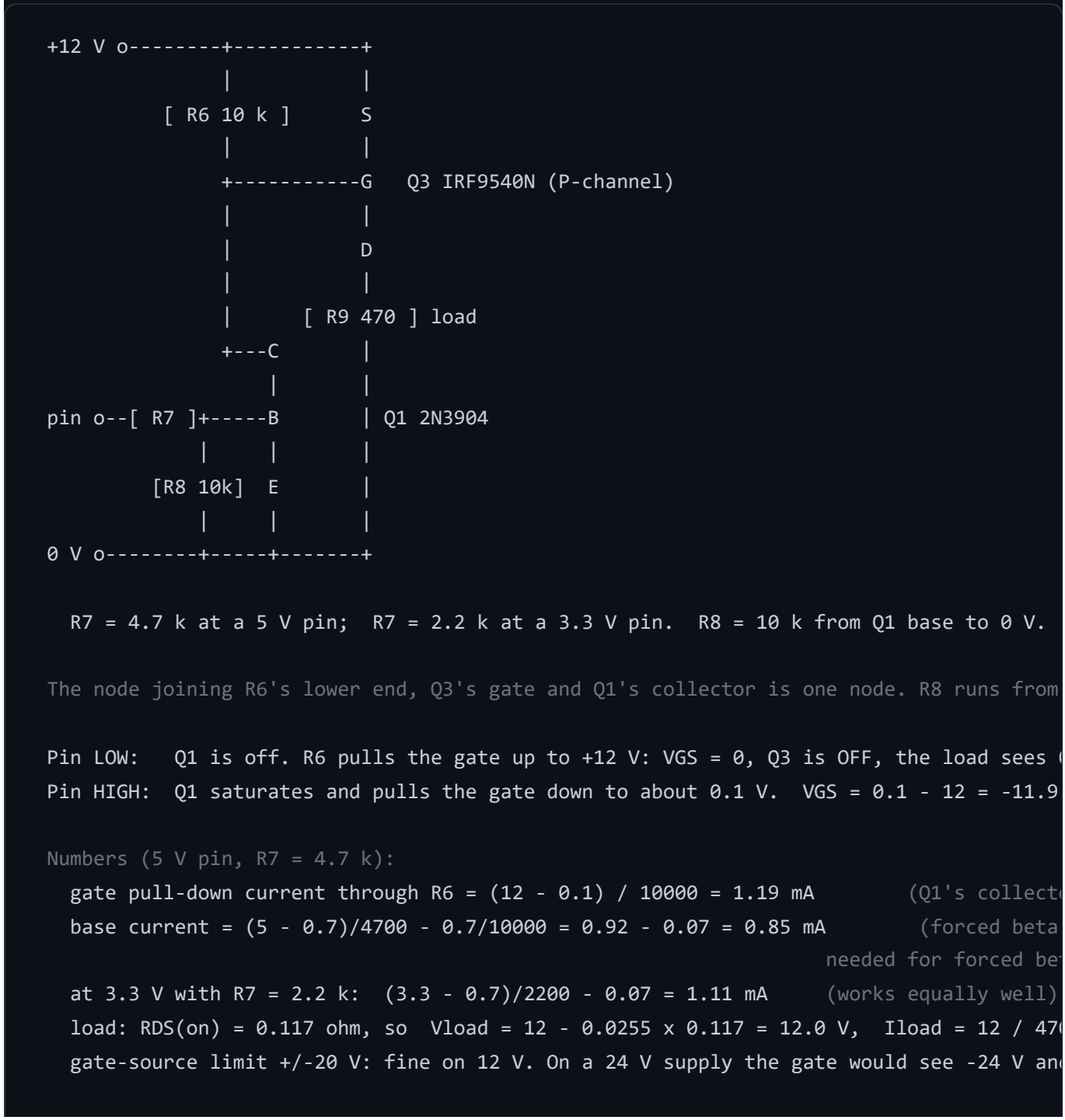
Stage B: the MOSFET motor switch (the project of the chapter)

```
  Hole list. Q2 IRLZ44N, printed face towards you, legs down: G column 10 row a, D column
     Rg:  column 5 row a   -> column 10 row b          (220 ohm: signal to gate)
     Rp:  column 10 row c  -> column 12 row c          (10 k: gate to source)
     Source: column 12 row b -> bottom - rail
     D1 1N5819: anode column 11 row b,  cathode (band) column 15 row b
     Jumper: column 15 row c -> top + rail (6 V)
     Motor: one lead to column 11 row c,  the other lead to column 15 row d
     Signal jumper: column 5 row b -> bottom + rail (5 V = ON) or bottom - rail (0 V = OFF
     Join: top - rail -> bottom - rail
     Connect the supplies LAST.
     Measure:  gate:       red probe column 10 row e, black on the - rail
               Vds:        red column 11 row e, black column 12 row e         (use the 200
               motor volts: red column 15 row e, black column 11 row e
```

1. Jumper to 0 V: gate 0 V, motor off, V_{DS} 6 V.
2. Jumper to 5 V: gate about 4.9 V; the motor runs; V_{DS} only a few millivolts; motor volts close to 6 V.

3. Hold the motor shaft: V_{DS} rises to about 30 mV as the current approaches 1.2 A; the MOSFET stays cold. Do not hold for more than a few seconds.
4. Take the signal jumper out completely (floating signal): the motor stays off because R_p holds the gate low. Then try removing R_p as well (battery disconnected while you do it) and reconnect the battery: a floating gate may switch on by itself; do not rely on it.
5. If you have a microcontroller board, use `analogWrite` or PWM on the pin to vary the speed; Basic Electronics 10 covers this.

Stage C: high-side P-channel switch on 12 V (simulator first)



```
    Turn-OFF speed: Ciss of about 1.3 nF recharges through R6:  tau = 10000 x 1.3e-9 = 13
    Turn-on is fast (Q1 sinks the current). So this circuit suits switching and PWM below
```

```
  Hole list for a real build (12 V adapter; the 5 V or 3.3 V signal on the bottom rails; j
    Q3 IRF9540N, printed face towards you, legs down: G column 10 row a, D column 11 row
    Source: column 12 row b -> top + rail (12 V)
    R6:   column 10 row b -> column 12 row c          (10 k: gate to source)
    R9:   column 11 row b -> column 15 row b          (470 ohm load)
    Jumper: column 15 row c -> top - rail
    Q1 2N3904, flat face towards you: E column 20 row a, B column 21 row a, C column 22 r
    Jumper: column 22 row b -> column 10 row c        (Q1 collector to Q3 gate)
    Emitter: column 20 row b -> top - rail
    R7:   column 25 row a -> column 21 row b          (4.7 k at 5 V, 2.2 k at 3.3 V)
    R8:   column 21 row c -> top - rail               (10 k: base to 0 V)
    Signal jumper: column 25 row b -> bottom + rail (signal HIGH) or bottom - rail (LOW)
    Join: top - rail -> bottom - rail
    Measure: gate: red probe column 10 row d, black on the - rail      load: red column 1
```

Simulators

Falstad Circuit Simulator (falstad.com/circuit): build each circuit from the *Draw* menu (*Transistors*: NPN, N-MOSFET, P-MOSFET; *Passive* parts; *Diodes*; *Inputs and Sources*: DC voltage sources and a logic input to flip the signal). The default NPN has a gain of 100 and a V_{BE} near 0.7 V. The default MOSFET has a threshold voltage of about 1.5 V, which suits a logic-level part: right-click it, choose *Edit*, raise the threshold to 3 V, and watch a 3.3 V gate fail to turn it on, which shows why standard-level parts need more gate voltage. A motor is not a stock part: use a series resistor and inductor (for example 5 Ω and 2 mH) and see the voltage spike at the drain with and without D1. The simulator's meters on each node read the values in the table below, within its own transistor models (a few tenths of a volt different).

Tinkercad Circuits (tinkercad.com): the component list contains an NPN transistor, a power MOSFET, a DC motor, a multimeter, a battery, an Arduino Uno and an oscilloscope. Use the Uno's pin 9 as the weak signal for stage B (pin HIGH is 5 V; put 220 Ω between pin and gate), add the multimeter in voltage mode to the nodes in the hole lists, and check each transistor's part number in its property box (model names and values differ from real parts, so compare to the table's ranges). Tinkercad's Uno also lets you run the PWM sketch from Basic Electronics 10 against the same wiring.

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
A OFF: V_{CE} / V_{load}	6.0 V / 0 V	6.0 V / 0.00 V	6 V / 0 V
A ON: V_{BE}	0.8 V	0.72 to 0.85 V	0.7 to 0.8 V
A ON: V_{CE}	0.2 V	0.1 to 0.3 V	0.05 to 0.3 V
A ON: V_{load} / current	5.8 V / 48 mA	5.7 to 5.9 V / 47 to 49 mA	5.8 V / 48 mA
A ON: V_{R1} / base current	4.2 V / 5.1 mA	4.1 to 4.3 V / 5.0 to 5.2 mA	4.2 V / 5.1 mA
A with R1 = 22 kΩ: V_{CE}	3.65 V (if h_{FE} = 100)	1.5 to 5.5 V, depends on the sample	3.6 V (default gain 100)
B OFF: gate / V_{DS}	0 V / 6.0 V	0.00 V / 6.0 V	0 V / 6 V
B ON: gate voltage	4.89 V	4.8 to 4.95 V	4.9 V
B ON (running): V_{DS}	6 mV	5 to 20 mV	a few mV (depends on the model)
B ON (running): motor volts	5.99 V	5.9 to 6.0 V (battery voltage sags under load)	6.0 V
B stalled: V_{DS} / current	30 mV / 1.2 A	20 to 60 mV (depends on the motor)	(motor model dependent)
B turn-off drain spike, 1N5819 / none	about 6.3 V / above the 55 V rating	(scope needed) about 6.3 V / tens of volts	about 6.3 V / large (hundreds of volts)
C signal LOW: gate / V_{load}	12 V / 0 V	12.0 V / 0.00 V	12 V / 0 V
C signal HIGH: gate / V_{GS}	0.1 V / −11.9 V	0.05 to 0.2 V / about −11.9 V	0.1 V / −11.9 V
C signal HIGH: V_{load} / current	12.0 V / 25.5 mA	11.9 to 12.0 V / 25 mA	12.0 V / 25.5 mA
C turn-off 10 to 90 % time	29 µs	(scope needed) about 25 to 45 µs	depends on the MOSFET model's capacitance

Real readings spread because transistors vary (h_{FE} by 2:1 or more, V_{BE} by tenths of a volt, $R_{DS(on)}$ by 30 % or more) and with temperature and the battery's state. The things that do not vary are the relationships: a saturated BJT drops about 0.2 V whatever the load; a logic-level MOSFET at full gate voltage drops I × $R_{DS(on)}$, millivolts at these currents; a floating gate is held off by its pull-down; and the base or gate resistor sets the control current, not the load current.

9. Challenges

CHALLENGE 1: BJT BASE RESISTORS

(a) A 2N3904 switches a 12 V relay coil of 400 Ω (flyback diode fitted) from a 5 V pin. Using $V_{CE(sat)}$ = 0.2 V, V_{BE} = 0.8 V and a forced beta of 10, find the saturation current, the base current, the base resistor (calculated, then the E12 value *below*), the actual base current and forced beta, and the power in the transistor. Then do the same from a 3.3 V pin that can supply no more than 2.5 mA, and state the forced beta you end up with and whether the minimum h_{FE} of 60 still leaves a margin. (b) The designer uses 47 kΩ by mistake (V_{BE} = 0.7 V). Find the base current and the collector voltage for a transistor with h_{FE} = 100 and with h_{FE} = 300, and say what each does to the relay. (c) A 9 V motor draws 500 mA and is driven from a 5 V pin through a TIP120 Darlington (V_{BE} total 1.4 V, $V_{CE(sat)}$ 0.9 V, forced beta 250): find the base current, the base resistor (round to E12 down), the motor voltage and the transistor's power, and compare with a MOSFET of $R_{DS(on)}$ = 25 mΩ.

View solution

CHALLENGE 2: MOSFET LOSSES AND THE GATE

(a) A 12 V motor runs at 1 A and stalls at 3 A. An IRLZ44N ($R_{DS(on)}$ 25 mΩ at 5 V, doubling when hot to 50 mΩ) switches it. Find V_{DS} and the power running and stalled with the hot value, the temperature rise when stalled in free air at 62 °C/W, and compare the stalled power with a BJT-style $V_{CE(sat)}$ of 0.9 V at 3 A. (b) The gate is driven from a 5 V pin through R_g = 220 Ω with C_{iss} = 3.3 nF: find the peak gate current, τ, and the 10 to 90 % rise time; then for R_g = 1 kΩ find the same values and say what that does to a 20 kHz PWM edge. (c) A 10 kΩ pull-down is fitted: find the gate voltage and the wasted current with a 5 V pin and R_g = 220 Ω; then with a 3.3 V pin and a 100 kΩ pull-down (R_g still 220 Ω), and say which is the better compromise for a floating pin. (d) An LED strip of 24 W on 12 V is switched by an AO3400A with $R_{DS(on)}$ of 50 mΩ from a 3.3 V pin: find the current, V_{DS} and the power in the MOSFET.

View solution

CHALLENGE 3: HIGH-SIDE SWITCHING AND LEVEL SHIFTING

(a) A 5 V signal must drive a 3.3 V input through a divider using a 1 kΩ series resistor and a 2 kΩ to ground: find the output voltage, the divider current, and the time constant with 20 pF of load; then find what happens if the pull-down is replaced by 4.7 kΩ. (b) A P-channel IRF9540N switches a 470 Ω load on a 24 V supply from a 3.3 V pin using the section 6 NPN shifter. With the gate pull-up R6 = 10 kΩ alone, find V_{GS} when Q1 is on and say why the part is destroyed; then add a second 10 kΩ resistor in series between the gate and Q1's collector and find the new V_{GS} ($V_{CE(sat)}$ = 0.1 V), Q1's collector current and V_{CE} when off, whether a 2N3904 (V_{CEO} 40 V) copes, and the base resistor for a 1 mA base current from 3.3 V (V_{BE} = 0.7 V, E12 down, with no base pull-down). Find also the 10 to 90 % turn-off time with C_{iss} = 1.3 nF. (c) Choose BJT or MOSFET and a board for each job and give a one-line reason: a 12 V buzzer at 30 mA from an Uno; a 12 V, 2 A pump pulsed at 1 kHz from an ESP32; a 5 V relay at 70 mA from a Pico; a 6 V, 100 mA filament lamp from an Uno.

View solution

QUICK REFERENCE

- **BJT switch:** $I_{C(sat)} = (V_s - V_{CE(sat)})/R_L$; $I_B = I_{C(sat)}/10$ (forced beta 10); $R_B = (V_{drive} - 0.8)/I_B$, round *down*; $V_{CE(sat)}$ about 0.2 V; power = $V_{CE(sat)} \times I_C$
- **Active region:** $I_C = h_{FE}\, I_B$; use the datasheet *minimum* h_{FE} at your current; the half-on transistor dissipates up to $V_s^2/(4R_L)$
- **Darlington:** gains multiply, V_{BE} about 1.4 V, $V_{CE(sat)}$ about 0.9 V or more; forced beta about 250
- **MOSFET switch:** $V_{DS} = I \times R_{DS(on)}$, $P = I^2 R_{DS(on)}$; read $R_{DS(on)}$ at *your* gate voltage, not the threshold; logic-level for 3.3 V and 5 V pins
- **Gate network:** R_g (100 to 470 Ω) limits the charging pulse (V/R_g) and sets the speed ($\tau = R_g C_{iss}$, 10 to 90 % = 2.2τ); R_p = 10 kΩ gate to source keeps it off at boot
- **Low-side:** NPN or N-MOSFET between load and 0 V, the easy way; **high-side:** PNP or P-MOSFET (source to +V); P-channel on 12 V from a 3.3 V pin needs an NPN shifter that pulls the gate down; keep $|V_{GS}|$ under 20 V
- **Loads:** relay and motor need a flyback diode across the load; a lamp has about 10× inrush; a motor stalls at 3 to 5× its running current; size for the worst case
- **Level shift:** divider (down only), BSS138 shifter with a pull-up on each side (two-way, I2C), NPN with a pull-up to the higher supply (up, inverting)

- **Choice:** BJT under about 100 mA and for small signal jobs; logic-level MOSFET over about 300 mA, for PWM, or for 3.3 V logic

Next, Basic Electronics 7 uses transistors in the half-on region on purpose: amplifiers, then the op-amp as comparator, buffer and amplifier, and conditioning the signal from a sensor so a microcontroller can read it.

Basic Electronics

Course 1 · Chapter 7 · Analogue Signals and Op-Amps

The real world is analogue. Temperature, light, force, current and sound all vary smoothly, and a sensor turns them into a voltage that can be any value in a range. A microcontroller, on the other hand, only understands numbers. The bridge between the two has two halves: **signal conditioning** (making the sensor's voltage the right size and shape) and the **analogue-to-digital converter** (ADC), which turns that voltage into a number. The star of the conditioning half is the **operational amplifier**, or op-amp, a small chip with two inputs and one output that can buffer, amplify, subtract, add or compare voltages. The chapter uses the divider rules of Basic Electronics 3, the RC filter of Basic Electronics 4 and the idea of a switch that is fully on or off from Basic Electronics 6 (in this chapter we meet the opposite, a circuit that works at every voltage in between).

1. Analogue and digital signals

	ANALOGUE SIGNAL	DIGITAL SIGNAL
Values it can take	Any voltage in a range (0 to 3.3 V, say), continuously	Two levels only: LOW and HIGH (a few volts apart)
Carries	A measurement: the voltage *is* the information	A bit; many bits in sequence make a number
Weakness	Any added noise is part of the signal and cannot be told apart from it	Noise below the switching threshold is ignored completely
Examples	Thermistor divider, light sensor, microphone, strain gauge, battery voltage	Button, encoder pulses, I2C and UART data, a PWM output

The ADC: sampling and quantisation

An ADC does two things. **Sampling** measures the voltage at regular instants, f_s times a second. **Quantisation** rounds each measurement to one of 2^N levels, where N is the number of bits. The step between levels is the **least significant bit (LSB)**:

```
LSB = Vref / 2^N                    code = floor( Vin / LSB )      (the number the pro
```

```
Each code covers a step of one LSB, so a reading is only right to within one LSB (about
converter rounds). That rounding error behaves like random noise of
  rms quantisation noise = LSB / sqrt(12) = 0.29 LSB
and an ideal converter fed a full-scale sine wave has a signal-to-noise ratio of
  SNR = 6.02 x N + 1.76 dB            (each extra bit buys 6 dB, a factor of 2 in volta

Worked example (12 bits, Vref = 3.3 V):
  LSB = 3.3 / 4096 = 0.806 mV
  Vin = 1.000 V:   1.000 / 0.000806 = 1241.2   so the code is 1241   (1241 x 0.806 mV =
  rms quantisation noise = 0.806 / 3.46 = 0.23 mV;   ideal SNR = 6.02 x 12 + 1.76 = 74
```

CONVERTER	BITS	FULL SCALE	LSB	NOTES
Arduino Uno / Nano (built in)	10	5 V (default)	4.88 mV	1024 codes; 6 inputs; never exceed 5 V on a pin
Raspberry Pi Pico (built in)	12	3.3 V	0.806 mV	3 usable inputs; the real accuracy is about 9 effective bits, so the bottom 3 bits are noise
ESP32 (built in)	12	about 0 to 3.1 V (attenuation setting)	0.8 mV	Many inputs, but noisy and non-linear near 0 V and 3.3 V; keep signals between about 0.15 and 3.0 V; the Wi-Fi radio blocks one of its two converters
ADS1115 (external, over I2C)	16	±4.096 V (gain setting)	125 µV	The upgrade for a precision measurement, and the way to add analogue inputs to a Raspberry Pi, which has none

These figures are typical; the datasheet of your board is the authority. The bit count says how finely a converter *can* read, not how accurately it does: noise, gain error and a poorly decoupled reference can ruin the bottom bits of a 12-bit converter.

Why the sampling rate matters: aliasing

A converter sampling at f_s can only see frequencies up to $f_s/2$ (the **Nyquist limit**). Anything faster is not just lost: it is **folded** back down and appears as a false, slower signal.

```
alias frequency = | f - k x fs |        where k is the whole number that gives the answ

fs = 1000 samples/s  (Nyquist limit 500 Hz).   A noise tone at 1100 Hz appears at |1100
```

```
The program cannot tell it from a real 100 Hz signal. The cure has to come BEFORE the A
(an RC network, Basic Electronics 4) that removes everything above fs/2. Section 7 desi
```

2. The ideal op-amp

An op-amp is a **differential amplifier with a very high gain**: it amplifies the *difference* between its two inputs.

```
Vout = A x ( V+ - V- )          A = open-loop gain: 100,000 or more (100 dB) at DC for

The symbol (the sign is printed beside each input: read the SIGN, not the position on t

         +----------+
IN-  ----| -        |
         |    U1    |----  OUT
IN+  ----| +        |
         +----------+
     (with power pins V+ at the top and V- at the bottom, usually left out of the drawi

Real chip, DIP-8, a dual op-amp (two amplifiers in one package: LM358, MCP6002):
  notch at the left, pins numbered anticlockwise from pin 1 at the bottom-left, as seen

       pin 8  V+ (positive supply)     pin 7  OUT B       pin 6  IN- B       pin 5  IN+
       pin 1  OUT A                    pin 2  IN- A       pin 3  IN+ A       pin 4  V-

Pin 8 is directly across from pin 1, and pin 5 across from pin 4. Check the datasheet o
op-amps have different pin orders.
```

Why you never use it open-loop as an amplifier

With A = 100,000, an input difference of 30 µV already drives the output through 3 V, and the gain varies wildly between chips, with temperature and with frequency. You make it useful by feeding part of the output back to the inverting input, called **negative feedback**. The feedback makes the circuit's gain depend only on the resistors, not on A.

```
Take the non-inverting circuit of section 3, with feedback fraction b = Rg / (Rf + Rg)
  V+ = Vin          V- = b x Vout          Vout = A x (Vin - b x Vout)
```

```
  Vout x (1 + A x b) = A x Vin     so     Gain = Vout / Vin = A / (1 + A x b) = (1 / b)

When A x b is much bigger than 1 the last bracket is almost exactly 1, so  Gain = 1 / b

Example, Rf = 10 k, Rg = 1 k:   b = 1/11,   ideal gain 11,   A = 100,000:
  Gain = 100,000 / (1 + 100,000 / 11) = 10.9988        (only 0.011 % below 11: the res

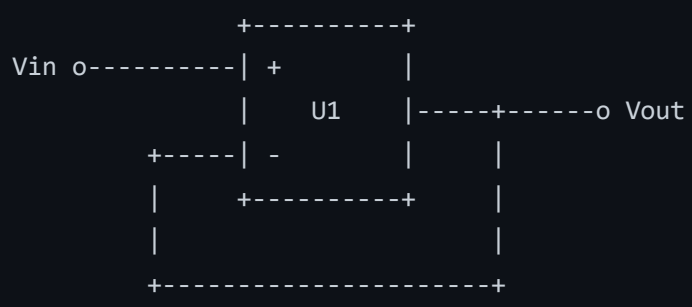
And the voltage between the inputs is  V+ - V- = Vout / A:  for Vout = 3 V that is 30 m
```

THE TWO GOLDEN RULES (FOR AN OP-AMP WITH NEGATIVE FEEDBACK)

Rule 1 (the virtual short): the output does whatever is needed to make V+ equal to V-. If you know one input's voltage, you know the other's. **Rule 2:** the inputs draw no current (a bipolar chip needs tens of nanoamps, a CMOS chip picoamps; both are negligible next to the currents in 1 kΩ to 100 kΩ resistors). Every circuit in section 3 follows from these two rules and Ohm's law. The rules apply only when feedback takes the output *back to the inverting input*; a circuit with no feedback, or with feedback to the non-inverting input (the comparator of section 4), follows different rules.

3. Building blocks with negative feedback

The voltage follower (buffer)




```
Rule 1: V- = V+ = Vin, and V- is wired straight to Vout, so   Vout = Vin       gain = 1

It copies the voltage but changes the impedance: the input draws almost nothing, the ou
Use it to stop a high-impedance source being loaded. Example: a divider of 10 k over 10
An ADC that needs to draw 20 microamps from it through 5 k of divider resistance would
between the divider and the ADC removes that error. Basic Electronics 3 showed the load
```

The non-inverting amplifier

```
                +----------+
Vin o-----------| +        |
                |    U1    |-----+------o Vout
          +-----| -        |     |
          |     +----------+     |
          +--------[ Rf ]--------+
          |
        [ Rg ]
          |
0 V o-----+

Rule 1: V- = V+ = Vin.      Rule 2: no current enters the - input, so the current throu
  current through Rg = (Vin - 0) / Rg = Vin / Rg
  Vout = Vin + (current) x Rf = Vin + Vin x Rf / Rg

  Gain = Vout / Vin = 1 + Rf / Rg               always 1 or more; the output has the S

Example: Rf = 39 k, Rg = 1 k:  gain 40.   A sensor output of 0 to 80 mV becomes 0 to 3.
Input impedance: huge (the + input draws nothing).     Rf = 0 (a wire) and Rg open give
```

The inverting amplifier

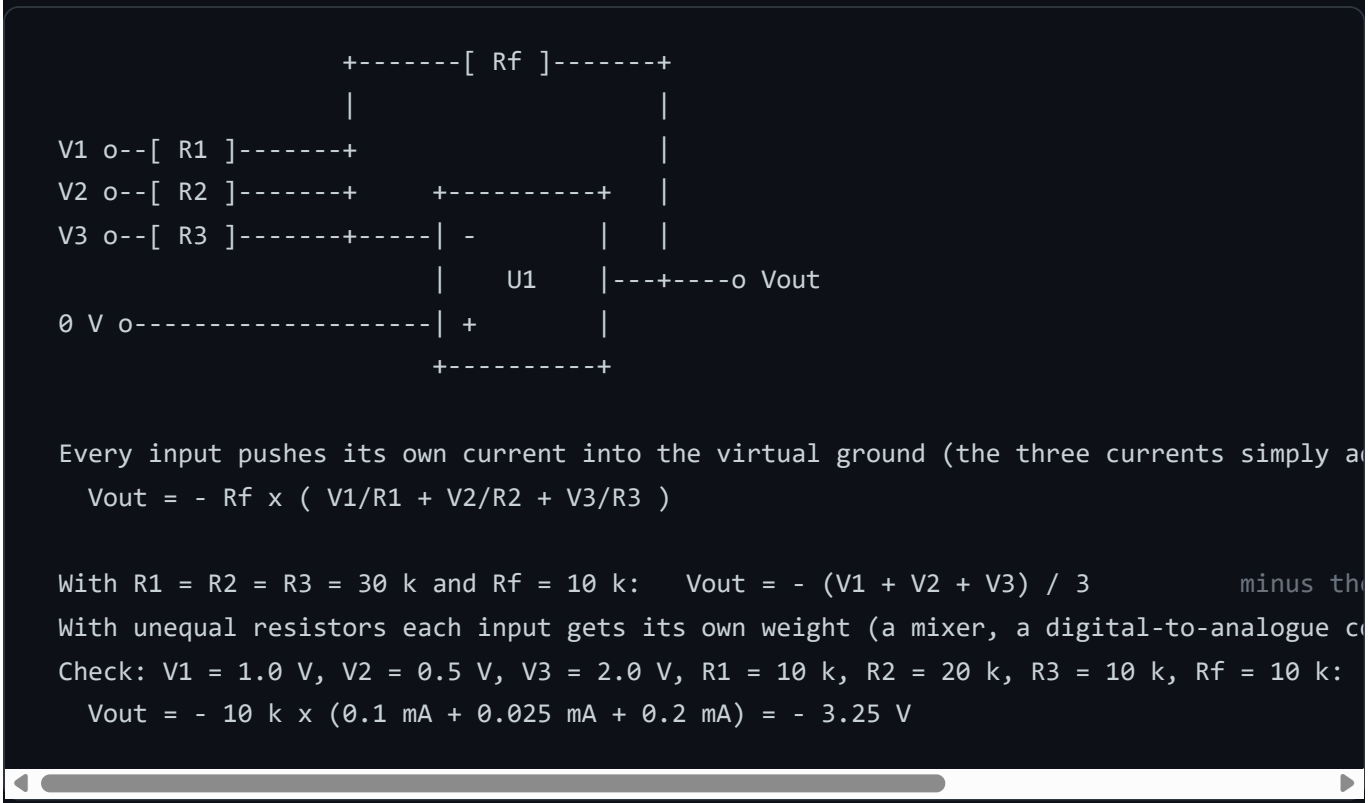
```
                   +-------[ Rf ]-------+
                   |                    |
                   |     +----------+   |
Vin o--[ Rin ]-----+-----| -        |   |
                         |    U1    |---+----o Vout
0 V o--------------------| +        |
                         +----------+

Rule 1: V+ = 0 V, so V- = 0 V: the - node is a "virtual ground" (at 0 V, but not connec
Rule 2: no current enters the - input, so the whole input current flows through Rf:
  input current = (Vin - 0) / Rin = Vin / Rin
  Vout = 0 - (input current) x Rf

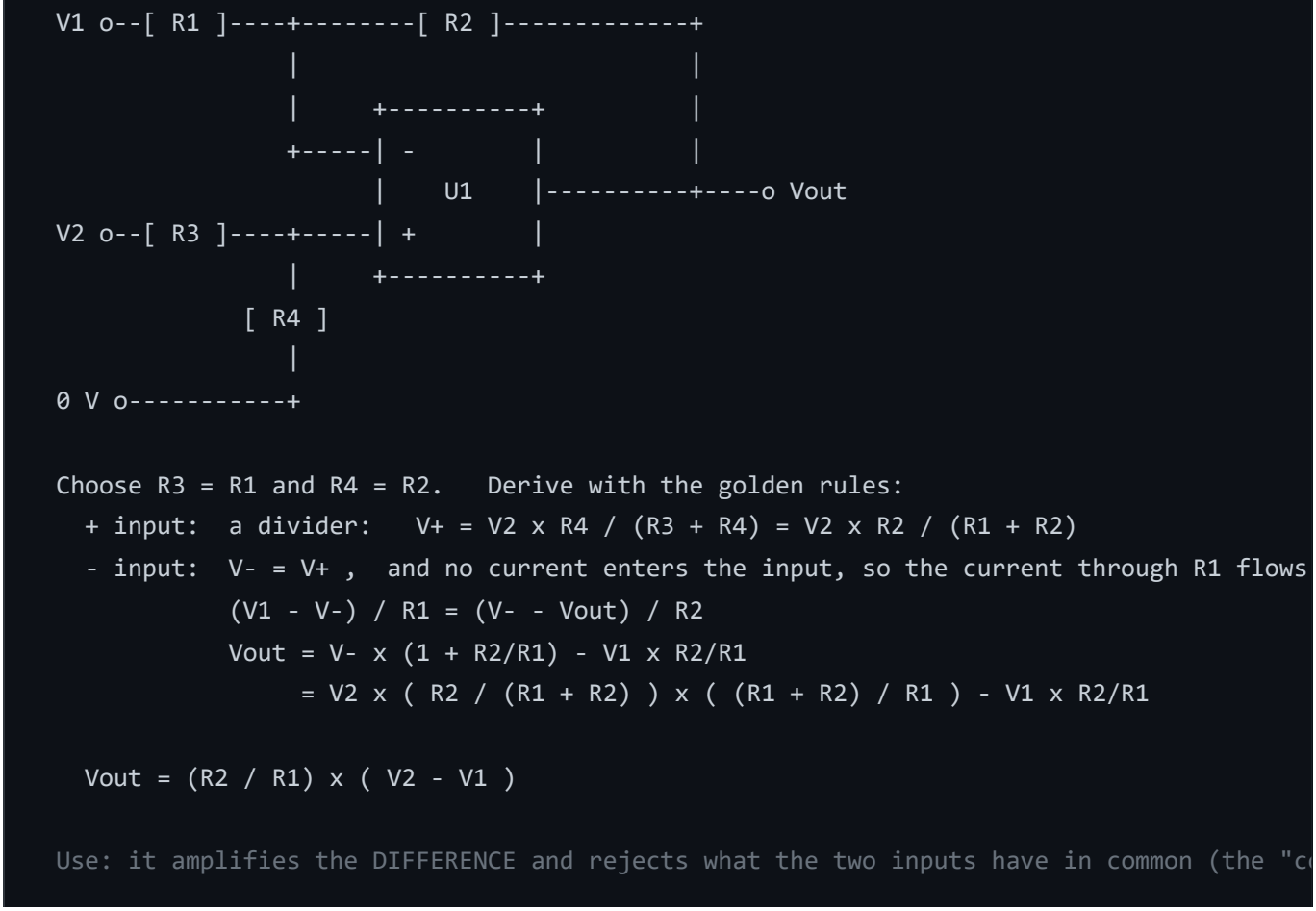
  Gain = Vout / Vin = - Rf / Rin                the minus sign: the output is upside-d

Example: Rin = 2 k, Rf = 10 k:  gain -5.    Vin = +0.5 V gives Vout = -2.5 V     (a neg
Input impedance = Rin (the signal source sees Rin to a virtual ground), unlike the non-
Unlike the non-inverting amplifier, the gain can be anything: 0.1 (attenuator) as well
```

The summing amplifier



The difference (subtracting) amplifier



```
or interference picked up by both wires). How well it rejects it depends on how closely
  worst-case CMRR  is about  (1 + R2/R1) / (4 x t)     for resistors of tolerance t
  unity gain, t = 1 %:   CMRR = 2 / 0.04 = 50   (34 dB)        hardly better than no r
  t = 0.1 %:             CMRR = 2 / 0.004 = 500  (54 dB)
The instrument amplifier of section 6 solves the weakness with matched resistors inside
Input impedance is only R1 + R2 (on the V1 side R1) and the two inputs see DIFFERENT im
```

CIRCUIT	GAIN	INPUT IMPEDANCE	USE
Follower	1	Very high	Isolate a high-impedance source; drive a load or an ADC
Non-inverting	$1 + R_f/R_g$	Very high	Scale a sensor signal up to the ADC range
Inverting	$-R_f/R_{in}$	R_{in}	Attenuate or amplify with a sign change; the base of the summer
Summing	$-R_f/R_i$ per input	R_i per input	Mix signals; add an offset to a signal
Difference	R_2/R_1	R_1 and R_1+R_2 (unequal)	Subtract two voltages; remove an offset; bridge sensors

4. The comparator: a decision, not an amplifier

With **no feedback** an op-amp is a **comparator**. The huge gain pushes the output hard against one supply rail or the other, depending only on which input is higher:

```
V+ higher than V-:   output goes to the HIGH rail (V+ supply)
V+ lower than V-:    output goes to the LOW rail (0 V on a single supply)

It turns an analogue voltage into a digital decision: "is the light above this level?",
Weakness: when the input hovers near the threshold, any noise makes the output chatter:
A sensor with 20 mV of noise sitting within 20 mV of the threshold gives a burst of fal
```

Hysteresis: the Schmitt trigger

The cure is **positive feedback**: a little of the output is fed back to the *non-inverting* input, so that the threshold itself moves each time the output switches. Once it has switched, the input must go back *past* the original threshold before it switches again. The two thresholds are VH (rising input) and VL (falling input), and the gap between them is the **hysteresis**.

```
                  +----------+
Vref o------------| -        |
                  |    U1    |----+----------o OUT
            +-----| +        |    |
            |     +----------+    |
Vin o-[ R1 ]-+                    |
            |                     |
            +-------[ Rf ]--------+

R1 carries the input to the + node, Rf feeds the output back to it. No current enters t
a weighted average (superposition):
  V+ = ( Vin x Rf + Vout x R1 ) / ( R1 + Rf )

The output switches when V+ equals Vref. Solve that for Vin:
  Vin(switch) = Vref x ( 1 + R1/Rf ) - Vout x ( R1/Rf )

Two possible values of Vout, two thresholds (Vh and Vl are the output's HIGH and LOW vo
  Output HIGH (Vout = Vh):  falling threshold   VL = Vref x (1 + R1/Rf) - Vh x R1/Rf
  Output LOW  (Vout = Vl):  rising threshold    VH = Vref x (1 + R1/Rf) - Vl x R1/Rf

  Hysteresis = VH - VL = (Vh - Vl) x R1 / Rf

Design (Vl = 0, Vh = Vcc): choose R1/Rf = hysteresis / Vcc;   then   VH = Vref x (1 + R

Worked example: Vcc = 3.3 V, Vref = 1.65 V, R1 = 10 k, Rf = 100 k:   R1/Rf = 0.1
  VH = 1.65 x 1.1 = 1.815 V          VL = 1.815 - 3.3 x 0.1 = 1.485 V          hysteres
  Output LOW, Vin rising:  the output goes HIGH at 1.815 V.   Output HIGH, Vin falling:
  A noise level of up to 0.33 V peak-to-peak can no longer cause a false switch.

Source impedance: the circuit above assumes Vin comes from a low-impedance source. A so
use (R1 + Rs) in the formulas, or put a follower in front, as the build does.
```

An op-amp or a comparator chip?

A comparator *chip* (LM393, LM339, TLV3201) is built for the job: it switches fast (tens to hundreds of nanoseconds), recovers at once from being driven hard, and an LM393 has an **open-collector** output that needs a pull-up resistor to whatever voltage the next circuit uses

(the same pull-up idea as Basic Electronics 3). A general op-amp as a comparator is slower (microseconds) and slow to recover from saturation, but it is fine for light levels, temperatures and battery voltages that change over seconds. Do not use a comparator chip in the linear circuits of section 3: it has no internal compensation, and feedback to its inverting input makes it oscillate.

5. Real op-amps

Gain-bandwidth product and slew rate

```
Open-loop gain falls with frequency. Above a few hertz, A x f is a constant, the gain-b
  A(f) = GBW / f                MCP6002 and LM358: about 1 MHz.   A = 1 at 1 MHz (the

So a closed-loop amplifier of gain G has the bandwidth (the -3 dB frequency)
  f(-3 dB) = GBW / G
  G = 11:   1 MHz / 11 = 91 kHz         G = 40:   1 MHz / 40 = 25 kHz         G = 1000:
Rule of thumb: you cannot have both high gain and high bandwidth from a cheap op-amp. F
20 MHz of GBW, or two amplifier stages of gain 32 each (1 MHz / 32 = 31 kHz each).

Example: gain 40 at 10 kHz: the gain is down by a factor of 1 / sqrt(1 + (10/25)^2) = 0

Slew rate (SR): the fastest the output can change, in volts per microsecond, set by an
A sine wave of peak voltage Vp and frequency f has a steepest slope of 2 x pi x f x Vp,
  2 x pi x f x Vp  is larger than  SR             so the full-power bandwidth is     f

  MCP6002 (SR 0.6 V/us), 1.65 V peak (a full 3.3 V swing):   f(max) = 0.6e6 / (2 x pi x
  LM358   (SR 0.3 V/us), same swing:                         f(max) = 0.3e6 / 10.37
For small signals the bandwidth limit is what you hit; for large signals the slew rate
```

Offset, bias current and single supply

- **Input offset voltage (V_{OS}):** a real op-amp behaves as if a small voltage (tens of microvolts to a few millivolts) sat in series with one input. It is amplified along with the signal: V_{OS} = 2 mV in a gain of 40 gives 80 mV at the output, 99 ADC counts at 3.3 V, 12 bits. Use a precision amplifier, or measure the offset and subtract it in software.
- **Input bias current (I_B):** 20 nA for a bipolar LM358, 1 pA for a CMOS MCP6002. Flowing through a source resistance R_s it drops $I_B \times R_s$: 20 nA × 1 MΩ = 20 mV. With a high-

impedance sensor (a pH probe, a photodiode) use a CMOS or JFET input.

- **Single supply.** A text-book op-amp has ±15 V supplies and swings both ways around 0 V. A robot has one battery, so the amplifier gets one supply, V+ and 0 V, and the signals must sit between 0 V and V+. Two limits then matter: the **input common-mode range** (how close to the rails the inputs may go) and the **output swing** (how close the output can get to each rail).
- **Rail-to-rail** chips (MCP6002, MCP6004, LMV358, OPA2340) have inputs and output that can go to within millivolts of both rails. A classic **LM358** can take its inputs down to 0 V and its output swings down to a few millivolts, but its output only goes up to about V+ minus 1.5 V: on a 3.3 V supply it tops out at 1.8 V, and it is out of spec below 3 V. *That is the reason the builds use an MCP6002.*
- **Inverting stages on a single supply** need the + input biased at a reference voltage (normally V+/2) instead of 0 V, and the signal is then amplified about that reference. The same trick, a reference node held by a follower, is used in stage B of the build to shift the sensor's voltage down.

PART	SUPPLY	SWING	GBW	SLEW RATE	USE HERE
LM358 / LM324 (dual / quad)	3 to 32 V	Inputs to 0 V; output 0 V to V+ − 1.5 V	about 1 MHz	0.3 V/µs	5 V and 12 V builds where the output needs not reach V+; cheap, tough, everywhere
MCP6002 / MCP6004 (dual / quad)	1.8 to 6 V	Rail-to-rail in and out	1 MHz	0.6 V/µs	3.3 V boards (Pico, ESP32): the output reaches both rails
LM393 (dual comparator)	2 to 36 V	Open-collector output	(not an amplifier)	(fast, about 1 µs)	Threshold detection with a pull-up to any logic level
TL072 (dual JFET)	±5 to ±18 V	Within 1.5 V of the rails	3 MHz	13 V/µs	Audio, from a split supply; not a 3.3 V part
INA333 (instrumentation amp)	1.8 to 5.5 V	Rail-to-rail output	(gain set by one resistor)	—	Strain-gauge and other bridge sensors (section 6)

All figures are typical; read the datasheet of the exact part. The pinouts of the dual versions are the same as the diagram in section 2.

6. Sensor conditioning

A sensor rarely gives a voltage between 0 and 3.3 V. Conditioning converts what it gives (a resistance change, a few millivolts, a small current) into a voltage that fills as much of the ADC's range as possible, without ever leaving it.

The thermistor

An **NTC thermistor** is a resistor whose resistance falls as it warms, steeply and not in a straight line. The usual model is the **B-parameter equation**, with R_0 the resistance at T_0 = 25 °C (298.15 K) and B (about 3950 K for a common 10 kΩ disc, quoted between 25 and 85 °C):

```
R(T) = R0 x exp( B x ( 1/T - 1/T0 ) )              T in KELVIN  (K = degrees C + 273.15

Solve for the temperature (this is the formula the program uses):
  T = 1 / ( 1/T0 + ln( R / R0 ) / B )

10 k, B = 3950:
  R(0 C)  = 10 k x exp(3950 x (1/273.15 - 1/298.15)) = 10 k x exp(1.2126) = 33.6 k
  R(50 C) = 10 k x exp(3950 x (1/323.15 - 1/298.15)) = 10 k x exp(-1.0250) = 3.59 k

Sensitivity: differentiate:   dR/dT = - B x R / T^2
  at 25 C:   dR/dT = - 3950 x 10000 / 298.15^2 = - 444 ohm per kelvin   (that is - 4.4 %

The curve is not straight; the equation is good to about 1 degree between 0 and 50 C and
accuracy use the Steinhart-Hart equation (three fitted constants) or a look-up table fr
```

A resistance is measured by turning it into a voltage with a divider (Basic Electronics 3). With the thermistor on the *top* and a fixed resistor R_b at the bottom, the voltage rises as it gets warmer:

```
+3.3 V o------+
              |
         [ NTC 10k ]
              |
              +--------o Vs
```

```
              |
         [ Rb 10 k ]
              |
0 V o---------+

Vs = 3.3 x Rb / ( Rt + Rb )          choose Rb = Rt at the middle of the range (10 k at

  T (C)    Rt          Vs
   0       33.6 k      0.757 V
  10       20.2 k      1.094 V
  25       10.0 k      1.650 V
  35        6.51 k     1.999 V
  40        5.30 k     2.157 V
  50        3.59 k     2.429 V

Sensitivity at 25 C:  dVs/dT = 3.3 x Rb x (-dR/dT) / (Rt + Rb)^2 = 3.3 x 10000 x 444 / 1
Over 0 to 50 C Vs spans only 1.67 V, about half of the ADC range and not centred: ampli

Self-heating: the current through the thermistor warms it, so the reading errs high. P =
  at 25 C: I = 3.3 / 20 k = 0.165 mA, P = 0.165e-3^2 x 10000 = 0.27 mW      (about 0.2 m
  Typical dissipation constant in still air = about 2 mW per C    so the error is  0.27
```

The photoresistor (LDR)

A **light-dependent resistor** (cadmium-sulphide cell, such as the GL5528) falls from megohms in the dark to a few hundred ohms in bright light. Its resistance follows a power law, a straight line on log axes:

```
R(L) = R10 x ( L / 10 lux ) ^ ( - gamma )        R10 = the resistance at 10 lux (given

Illustration, R10 = 15 k, gamma = 0.7, in a divider with the LDR on TOP and a 10 k resi
   1 lux (a dim room):          R = 15 k x 10^0.7  = 75 k          Vs = 3.3 x 10 / (75 +
  10 lux:                       R = 15 k                           Vs = 3.3 x 10 / 25 =
 100 lux (an indoor desk):      R = 15 k x 10^-0.7 = 3.0 k         Vs = 3.3 x 10 / 13 =
1000 lux (a bright window):     R = 0.6 k                          Vs = 3.3 x 10 / 10.6

Four decades of light become 0.4 to 3.1 V: the divider does the job unaided, and the lo
gamma vary by 2:1 between samples, so calibrate each unit, or use it only to compare wi
and its cadmium is restricted in some regions; a phototransistor or a TEMT6000 module (
```

The strain gauge and the Wheatstone bridge

A **strain gauge** is a thin metal foil pattern bonded to a part: stretch it and its resistance rises a tiny amount. The **gauge factor** GF (about 2) is the ratio of the fractional change in resistance to the strain ε (the fractional change in length):

```
dR / R = GF x strain          350 ohm gauge, GF = 2, strain 2000 microstrain (0.002):

A change of 0.4 % cannot be seen by a divider at the volt level. A bridge measures only

        +3.3 V (excitation)
     +-------+--------+
     |                |
 [ R1 350 ]      [ R3 350 ]
     |                |
     +----- Va        +----- Vb
     |                |
 [ R2 350 ]      [ Rg 350 (1 + x) ]      Rg = the active gauge,  x = dR/R
     |                |
     +-------+--------+
           0 V

Balanced (x = 0): Va = Vb = 1.65 V and the difference is exactly 0, whatever the supply
  Vdiff = Vb - Va = Vex x ( Rg/(R3+Rg) - 1/2 ) = Vex x x / ( 2 x (2 + x) )       about
  x = 0.004 (2000 microstrain):   Vdiff = 3.3 x 0.004 / (2 x 2.004) = 3.293 mV     (the

A few millivolts, sitting on 1.65 V that both nodes share: a plain amplifier cannot sep
and a difference amplifier from resistors needs matching to 0.01 % or better. This is t
```

The instrumentation amplifier

An **instrumentation amplifier** (in-amp) is three op-amps in a standard arrangement, with laser-trimmed matched resistors, in one chip: two input buffers (which give each input a very high impedance, fixing the weakness in the difference amplifier of section 3) followed by a difference amplifier. One external resistor R_G sets the gain, for the INA333:

```
Inside: two buffer amplifiers, each with a resistor R to its output, sharing RG between
By rule 1 the voltage across RG equals the differential input Vd, so the current throug
  voltage across the whole chain (R + RG + R) = Vd x ( 1 + 2R / RG )
The output stage subtracts the two buffer outputs with gain 1, so
  Gain = 1 + 2R / RG        INA333: 2R = 100 k:   G = 1 + 100 k / RG        (RG open: g

Strain gauge, gain chosen so that 2000 microstrain gives a 1.5 V change:    Vout = REF
```

```
    REF pin tied to a 1.65 V node (the output can then move up AND down about 1.65 V for
    G = 1.5 / 3.293 mV = 455.5      RG = 100 k / (G - 1) = 220 ohm  (a standard value, G
    600 microstrain:  Vdiff = 0.989 mV,  Vout = 1.65 + 455.5 x 0.989 mV = 2.10 V
```

For a complete weighing scale, a ready-made module such as the HX711 combines the in-amp and a 24-bit ADC and talks to any board with two pins; the principles above are what is inside it.

Current sensing

To measure a current, put a small known **shunt resistor** in its path and measure the voltage across it: $V = I \times R_s$ (Ohm's law, Basic Electronics 1). The amplifier makes the small voltage readable.

```
+V o-------[ LOAD ]--------+---- to the + input of a non-inverting amplifier  (Vsense =
                           |
                      [ Rs 0.05 ]
                           |
0 V o----------------------+

Choose Rs so the loss is small, but the voltage large enough to read:
  Rs = 0.05 ohm, I = 3 A:   Vsense = 0.15 V      power in Rs = 3^2 x 0.05 = 0.45 W
  Non-inverting amplifier Rf = 20 k, Rg = 1 k:   gain 21,   Vout = 3.15 V at 3 A     (1

Disadvantage of the "low-side" shunt drawn here: the load's return is lifted 0.15 V abo
anything else connected to the load differ by that amount. A "high-side" shunt (between
difference amplifier or a dedicated current-sense chip (INA219, INA180, ACS712 modules,
Offset error: an op-amp V(OS) of 2 mV gives 2 x 21 = 42 mV at the output, equal to 42 m
```

7. Noise, decoupling, grounding and filtering

An analogue signal's enemy is noise. The small signals that sensors produce (millivolts) are easily swamped, and the converter's LSB is only 0.8 mV on a 3.3 V, 12-bit ADC.

SOURCE	WHERE IT COMES FROM	SIZE (EXAMPLE)	CURE
Thermal (Johnson) noise	Random motion of charge in every resistor	$v_n = \sqrt{(4\ k\ T\ R\ \Delta f)}$: 10 kΩ, 10 kHz: 1.29 µV rms; 1 MΩ: 12.9 µV	Lower resistances, narrower bandwidth (filter)

SOURCE	WHERE IT COMES FROM	SIZE (EXAMPLE)	CURE
Mains pickup	Capacitance between your wire and the 230 V, 50 Hz wiring	1 pF coupling: $i = 2\pi fCV = 2\pi \times 50 \times 1\text{ pF} \times 230\text{ V} = 72$ nA; into 100 kΩ = 7.2 mV; into 1 kΩ = 72 µV	Low-impedance nodes, short wires, shielding, averaging over 20 ms
Supply noise and spikes	Motors, relays, a microcontroller's switching, a USB supply's ripple	Tens of millivolts at switching edges	Decoupling capacitors, separate analogue supply or filter
Ground shifts	Large currents flowing in wires that are also the signal's return	1 A through 20 mΩ of wire = 20 mV = 25 LSB of a 12-bit, 3.3 V ADC	Star grounding, separate return wire

Decoupling the op-amp

Every op-amp and ADC needs a **100 nF ceramic capacitor right at its supply pins** (the leads as short as possible), plus a 10 µF capacitor somewhere on the board for slower changes. The capacitor supplies the brief bursts of current the chip draws when its output changes, so that the burst does not travel along the supply wires and appear as a voltage ripple on everything else (the decoupling idea of Basic Electronics 4). Without it an op-amp may oscillate at megahertz, giving strange, unstable readings.

Grounding

- **Separate the paths.** A motor's current should never share a wire with a sensor's return. Join the motor ground and the signal ground at *one* point only, near the supply (a **star** ground).
- **Reference the sensor to the ADC's own ground,** not to a ground at the other end of a noisy wire. A ratiometric sensor (the divider and bridge circuits above, whose output is a fraction of the supply) powered from the same 3.3 V as the ADC's reference also cancels supply variations: if the supply drops 2 %, the output and the converter's full scale both drop 2 % and the code is unchanged.
- **Keep analogue and switching signals apart** on the board; cross at right angles if they must cross.

Shielding and cabling

- Run a sensor's two wires **twisted together** so both pick up the same interference, which a difference amplifier or in-amp then cancels (common-mode rejection again).

- For a long run, use shielded cable with the shield connected to ground at **one end only** (the amplifier end), so that no current can flow along the shield.
- High-impedance nodes (the + input of an op-amp, an ADC pin with a large source resistance) are the most sensitive; keep their wires very short, or drive them from a follower.

The anti-alias filter

The cure for aliasing is a low-pass filter ahead of the ADC, a simple RC (Basic Electronics 4: f_c = 1/(2πRC)), with its corner well below $f_s/2$ and above the highest frequency you want to measure.

```
op-amp OUT o----[ R 1.5 k ]----+----o to the ADC pin
                               |
                          [ C 1 uF ]
                               |
0 V o--------------------------+

fc = 1 / (2 x pi x R x C) = 1 / (2 x pi x 1500 x 1e-6) = 106 Hz
Attenuation at frequency f:   |H| = 1 / sqrt( 1 + (f / fc)^2 )
  50 Hz:    1 / sqrt(1 + 0.222) = 0.905    (-0.9 dB)       hum at 50 Hz passes almost u
  10 kHz:   1 / sqrt(1 + 8900)  = 0.0106   (-39.5 dB)      fast noise is reduced almost

This suits a temperature sensor read 10 to 100 times a second (the thermistor changes o
Choose R no larger than the ADC allows: an Uno's input wants a source resistance of 10
from the capacitor, which could otherwise make a cheap op-amp unstable.

Digital helpers:
  Averaging N readings cuts random noise by sqrt(N):  16 readings = 4 times less noise,
  4^n readings add n bits, if there is enough noise to dither the signal).
  Mains hum: average over a whole number of mains cycles, 20 ms for 50 Hz (16.7 ms for
```

8. Which board for analogue work?

JOB	BEST BOARD	WHY
A single thermistor or light sensor, easy to wire, 5 V signals	Arduino Uno or Nano	10-bit ADC, 5 V reference, rugged pins, a one-line `analogRead()`; the stage B output (0.15 to 3.0 V) simply uses 60 % of its range

JOB	BEST BOARD	WHY
Several sensors on a small robot, 3.3 V logic, low cost	Raspberry Pi Pico	12-bit ADC and a clean 3.3 V reference pin (ADC_VREF); a pair of 3.3 V rails that match the op-amp stage of this chapter
Sensors plus Wi-Fi in one board	ESP32	Many analogue pins and a radio; use the middle of the range (0.15 to 3.0 V), average readings, and do not use the second converter while Wi-Fi is on
Precise bridge or gauge measurement (weighing, strain)	Any board with an external ADS1115 or HX711	The board's own converter is too coarse and too noisy for microvolt signals
A full Raspberry Pi computer	Raspberry Pi with an external ADC	It has no analogue inputs at all: add an ADS1115 (I2C) or an MCP3008 (SPI) and power it from the Pi's 3.3 V

Basic Electronics 10 reads these voltages in code and Basic Electronics 11 compares the boards in full.

9. Build it: a light trigger and a conditioned temperature sensor

Two stages on a 3.3 V supply. **A** is a light-triggered comparator with hysteresis, whose output lights an LED (it could equally drive a pin, a relay stage from Basic Electronics 6, or a robot's decision to turn on a lamp). **B** is a thermistor conditioned so that 0 to 50 °C gives about 0.15 to 3.0 V, ready for an ADC. Both use the MCP6002, which swings its output to both rails.

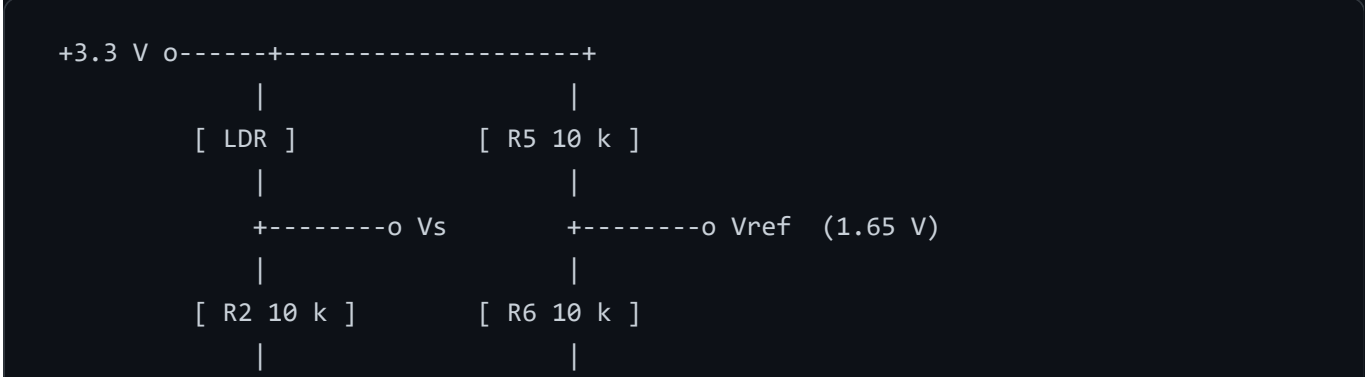
> **POWER AND STATIC**
>
> The MCP6002 runs from 1.8 to 6 V and is damaged above about 7 V: use a 3.3 V supply (a breadboard power module set to 3.3 V, or the 3V3 pin of a Pico or ESP32 board) and check it with the meter **before** you connect the chip. Fit the 100 nF capacitor across the supply rails next to each chip. Connect the power last. A 5 V supply also works (all the voltages scale; the comparator thresholds are ratios of the supply), but then the stage B output reaches 4.5 V, above what a 3.3 V board's pin can take.

Parts list

PART	VALUE	USED IN
U1, U2	MCP6002 dual rail-to-rail op-amp, DIP-8 (two chips; or an LMV358 / MCP6002 breakout)	A (U1), B (U2)
LDR	Photoresistor, GL5528 type (about 10 to 20 kΩ at 10 lux)	A
NTC	10 kΩ at 25 °C NTC thermistor, B = 3950	B
R1, R2, R5, R6, R8, R9, R10, Rg	10 kΩ, 1/4 W (brown-black-orange)	A (R1, R2, R5, R6), B (R8, R9, R10, Rg)
Rf (stage A)	100 kΩ (brown-black-yellow)	A
R7	1 kΩ (brown-black-red)	A
Rf (stage B)	6.8 kΩ (blue-grey-red)	B
Raa	1.5 kΩ (brown-green-red)	B (anti-alias, optional)
Caa	1 µF ceramic or electrolytic (+ to the op-amp side)	B (optional)
C1, C2	100 nF ceramic	decoupling, one per chip
LED	Red, 5 mm	A
Supply	3.3 V (breadboard module or the 3V3 pin of a board)	A, B
Breadboard, jumpers, multimeter	A cup of ice water and a mug of warm water for the temperature test	A, B

Stage A: the light-triggered comparator

The signal path: the LDR divider gives V_s; a follower (U1A) buffers it so that its source resistance cannot disturb the hysteresis resistors; U1B is the comparator with hysteresis of section 4; the LED shows the output.



```
 0 V o--------+--------------------+

 U1A: a follower.  Vs to pin 3 (IN+ A);  pin 2 (IN- A) wired to pin 1 (OUT A).   Pin 1 i

 U1B: comparator with hysteresis:


                   +----------+
 Vref o------------| -  (6)   |
                   |    U1B   |----+------+---o OUT (pin 7)
             +-----| +  (5)   |    |      |
             |     +----------+    |  [ R7 1 k ]
 Vbuf o[ R1 ]-+                    |      |
       10 k   |                    |  [ LED ]
              |                    |      |
              +-----[ Rf 100 k ]----+      |
                                  0 V o

 Check the dividers' effect on the thresholds: R1 = 10 k, Rf = 100 k, Vref = 1.65 V, Vcc
   VH = 1.65 x (1 + 0.1) = 1.815 V         VL = 1.815 - 3.3 x 0.1 = 1.485 V         hyst
 In terms of the LDR (LDR on top, 10 k below):  Vs = 3.3 x 10 k / (R(LDR) + 10 k)
   Vs rising to 1.815 V:   R(LDR) = 10 k x (3.3 / 1.815 - 1) = 8.18 k     the LED comes
   Vs falling to 1.485 V:  R(LDR) = 10 k x (3.3 / 1.485 - 1) = 12.2 k     the LED goes O
   In between (8.2 k to 12.2 k) the LED keeps whatever state it already had: no flicker.
   LED current = (3.3 - 2.0) / 1 k = 1.3 mA (dim but clear; use 470 ohm for 2.8 mA).
   Pin 5 voltage when the output switches: it jumps by Vcc x R1/(R1+Rf) = 3.3 x 0.0909 =

 The comparator makes a "light is on" decision with the dead band built in. For a night-
 Vbuf to the - input and Vref (with R1 and Rf) to the + input, or put the LDR at the bot
```

```
Breadboard layout. Top rails: + rail = 3.3 V, - rail = 0 V.  C1 (100 nF) sits across the
U1 straddles the centre gap with the notch towards column 1. Pins 1 to 4 in row e, colum
(pin 8 across from pin 1 in column 10, pin 5 across from pin 4 in column 13). Columns ha
   U1 pin 1 = column 10 row e     pin 2 = column 11 row e     pin 3 = column 12 row e
   U1 pin 8 = column 10 row f     pin 7 = column 11 row f     pin 6 = column 12 row f
   (if your board has the rows lettered the other way round, turn the chip so pin 8 is s

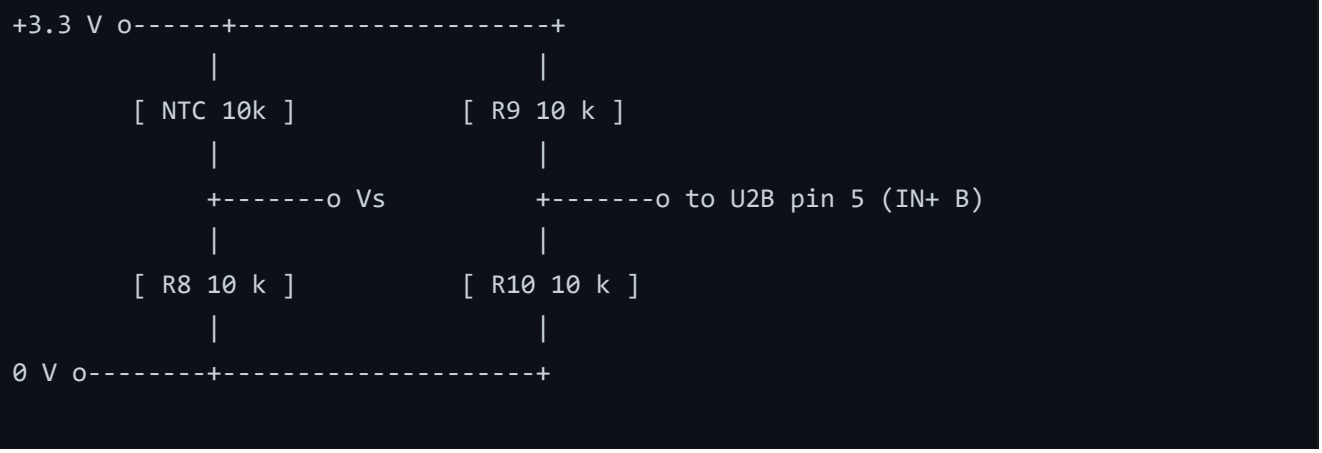
Power:   jumper column 10 row g -> top + rail          (pin 8, V+)
         jumper column 13 row a -> top - rail          (pin 4, V-)
Follower U1A:  jumper column 10 row a -> column 11 row a        (OUT A to IN- A)
Sensor divider:   LDR:  column 20 row a -> column 21 row a;   jumper column 21 row b ->
                  R2 (10 k):  column 20 row b -> column 22 row b;   jumper column 22 row
                  jumper column 20 row c -> column 12 row b            (Vs to IN+ A, pin
Buffered signal:  jumper column 10 row b -> column 16 row g            (Vbuf)
```

```
Comparator U1B:   R1 (10 k):   column 16 row h -> column 13 row h       (Vbuf to IN+ B, p
                  Rf (100 k):  column 11 row h -> column 13 row i       (OUT B, pin 7, b
Reference:        R5 (10 k): column 18 row g -> column 20 row g;   jumper column 20 row h
                  R6 (10 k): column 18 row i -> column 21 row i;   jumper column 21 row j
                  jumper column 18 row j -> column 12 row g           (Vref to IN- B, pi
Indicator:        R7 (1 k): column 11 row j -> column 25 row j;   LED anode column 25 ro
                  jumper column 26 row h -> top - rail
Connect the power LAST.
Measure:   Vs = red probe column 12 row a;   Vref = red column 12 row h;   pin 5 = red c
           (black probe on the - rail for all)
```

1. Cover the LDR with your hand: V_s falls to a few hundred millivolts, the LED is off and OUT is about 0 V. Pin 6 (V_{ref}) reads 1.65 V.
2. Bring the LDR slowly towards a lamp, watching V_s: the LED should switch on when V_s reaches about 1.8 V; at the same instant pin 5 jumps by 0.3 V.
3. Take the lamp away slowly: the LED stays lit until V_s drops below about 1.5 V, then goes off. That is the hysteresis (0.33 V).
4. *Test it without light:* swap the LDR for a 20 kΩ potentiometer wired as a variable resistor and turn it slowly, reading V_s and the LED. Then remove R_f (100 kΩ) from the breadboard: the circuit loses its hysteresis, and with the pot held near the threshold the LED may flicker or
5. Change R_f to 47 kΩ and find the new thresholds by calculation (challenge 2) and by test.

Stage B: the conditioned temperature sensor

The thermistor divider gives V_s from 0.757 V (0 °C) to 2.429 V (50 °C), about 1.67 V of span. U2A scales and shifts it so it fills the ADC, using the reference V_o of 1.65 V (buffered by U2B):



```
U2B: a follower.  Pin 5 (IN+ B) = the 1.65 V node;  pin 6 (IN- B) wired to pin 7 (OUT B

U2A: the non-inverting amplifier of section 3, but the bottom of Rg is connected to Vo

                  +----------+
Vs o--------------| +  (3)   |
                  |    U2A   |-----+----------o Vout   (to the ADC, through the optiona
            +-----| -  (2)   |     |
            |     +----------+     |
            +------[ Rf 6.8 k ]----+
            |
         [ Rg 10 k ]
            |
Vo o---------+          (pin 7 of U2B)

Derive with superposition (the amplifier is linear): Vs enters the + input, so it is amp
Vo is on the bottom of Rg, so the circuit treats it as an inverting input: - (Rf/Rg) x V
  Vout = (1 + Rf/Rg) x Vs - (Rf/Rg) x Vo           Rf/Rg = 0.68       Vo = 1.65 V
       = 1.68 x Vs - 1.122 V

  0 C:    Vs = 0.757 V:  Vout = 1.68 x 0.757 - 1.122 = 0.149 V
  25 C:   Vs = 1.650 V:  Vout = 1.68 x 1.650 - 1.122 = 1.650 V      (Vs = Vo: no offset
  50 C:   Vs = 2.429 V:  Vout = 1.68 x 2.429 - 1.122 = 2.958 V
The span is now 2.81 V, and nothing is outside 0.15 to 3.0 V, comfortably inside every

Gain at the ADC: dVout/dT = 1.68 x 36.6 mV/K = 61.6 mV per kelvin at 25 C.
  12-bit, 3.3 V (LSB 0.806 mV):   0.806 / 61.6 = 0.013 K per count        far finer than
  10-bit, 5 V (LSB 4.88 mV; Vout scales to 0.22 to 4.5 V if the whole circuit runs on 5

In the program: from a reading (12-bit, 3.3 V reference), e.g. code 1000:
  Vout = 1000 x 3.3 / 4095 = 0.806 V        Vs = (Vout + 1.122) / 1.68 = 1.1475 V
  Rt = 10 k x (3.3 / Vs - 1) = 10 k x (2.876 - 1) = 18.76 k        T = 1 / (1/298.15 +
```

```
Breadboard layout (the same breadboard, further along; the pin positions work as in stage
   U2 pin 1 = column 40 row e     pin 2 = column 41 row e     pin 3 = column 42 row e
   U2 pin 8 = column 40 row f     pin 7 = column 41 row f     pin 6 = column 42 row f
   C2 (100 nF) across the top rails next to U2.
Power:   jumper column 40 row g -> top + rail          (pin 8)
         jumper column 43 row a -> top - rail          (pin 4)
Thermistor divider:  NTC: column 50 row a -> column 51 row a;   jumper column 51 row b -
                     R8 (10 k): column 50 row b -> column 52 row b;   jumper column 52 r
                     jumper column 50 row c -> column 42 row b             (Vs to IN+ A,
Reference:           R9 (10 k): column 55 row a -> column 56 row a;   jumper column 56 r
```

```
                    R10 (10 k): column 55 row b -> column 57 row b;   jumper column 57 r
                    jumper column 55 row c -> column 43 row g            (1.65 V to IN+
  Follower U2B:     jumper column 41 row g -> column 42 row g            (OUT B pin 7 to
  Gain network U2A: Rf (6.8 k): column 40 row b -> column 44 row b;   jumper column 44 r
                    (OUT A pin 1 through Rf to IN- A pin 2)
                    Rg (10 k): column 41 row h -> column 46 row h;   jumper column 46 ro
                    (Vo from OUT B pin 7 through Rg to IN- A pin 2)
  Output:           column 40 row d is Vout.   Optional anti-alias filter: Raa (1.5 k):
                    Caa (1 uF): column 47 row b -> column 48 row b;   jumper column 48 r
                    (electrolytic: + leg in column 47);  the ADC wire goes to column 47
  Connect the power LAST.
  Measure:   Vs = red probe column 50 row d;   Vo = red column 55 row d;   Vout = red colu
```

1. With the thermistor at room temperature (about 20 °C): V_o reads 1.65 V; V_s about 1.47 V; V_{out} about 1.34 V (compare the table below, and measure the room temperature with a thermometer).
2. Warm the thermistor between your fingers (about 30 to 33 °C at its surface): V_{out} rises smoothly by about 60 mV for each kelvin; it climbs over seconds because the thermistor has thermal mass.
3. Put it into a bag in ice water (0 °C) and then warm (about 45 °C) water, checking the table. Keep the leads and joints dry: the water's conductivity would short the divider.
4. Connect V_{out} to an ADC pin (with the 1.5 kΩ and 1 µF filter) of a Pico or ESP32, apply the conversion at the end of the stage's schematic in code, and compare with the thermometer: agreement within about 1 °C is typical (the thermistor tolerance is the limit).

Simulators

Falstad Circuit Simulator (falstad.com/circuit): from the *Draw* menu take the op-amp (*Active Components*), resistors and a voltage source (*Inputs and Sources*). Falstad's op-amp is a very high-gain ideal amplifier with a default output limit of ±15 V: right-click the op-amp, choose *Edit* and set *Max Output* to 3.3 and *Min Output* to 0 to model the single-supply chip. Use a 3.3 V source and a slider-controlled resistor (a *Potentiometer*, or a resistor with its value editable) in place of the LDR, then watch the output switch at the thresholds, and view the hysteresis loop by adding a scope trace of Vs and OUT. The thermistor can be modelled with a potentiometer (or the thermistor element if your version has one) set to the values in the table. Falstad's op-amp has no offset or noise, so it shows the calculated numbers exactly.

Tinkercad Circuits (tinkercad.com): the component list includes an Op Amp, a Photoresistor, a Potentiometer, a multimeter, an Arduino Uno and a power supply that can be set to 3.3 V. Tinkercad's op-amp model need not swing all the way to the rails, so measure the output while overdriven and compare with the table: any difference tells you about the model, not the circuit. The Photoresistor part is convenient: Tinkercad lets you change the light level in the simulation with a slider. Feed the stage B output to the Uno's A0 pin and print the reading with `analogRead` (the Uno's 5 V full scale; 3.3 V corresponds to a code of about 675).

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
A: V_{ref} (pin 6)	1.650 V	1.63 to 1.67 V (1 % resistors)	1.650 V
A: LDR covered, V_s / OUT	below 0.3 V / 0 V	0.05 to 0.3 V / 0.00 V	(set the resistor to 100 kΩ) 0.3 V / 0 V
A: rising threshold (LED on)	V_s = 1.815 V (LDR 8.18 kΩ)	1.78 to 1.85 V	1.815 V
A: falling threshold (LED off)	V_s = 1.485 V (LDR 12.2 kΩ)	1.45 to 1.52 V	1.485 V
A: hysteresis	0.330 V	0.32 to 0.34 V	0.330 V
A: OUT high / low	3.3 V / 0 V	3.27 to 3.30 V / 0.00 to 0.03 V	3.3 V / 0 V
A: pin 5 just after switching on	1.95 V	1.93 to 1.97 V	1.95 V
A: LED current	1.3 mA	1.2 to 1.4 mA	1.3 mA
B: V_o (pin 7 of U2B)	1.650 V	1.63 to 1.67 V	1.650 V
B at 0 °C: R_t / V_s / V_{out}	33.6 kΩ / 0.757 V / 0.149 V	V_{out} 0.10 to 0.25 V	0.149 V
B at 10 °C: R_t / V_s / V_{out}	20.2 kΩ / 1.094 V / 0.715 V	0.65 to 0.78 V	0.715 V
B at 25 °C: R_t / V_s / V_{out}	10.0 kΩ / 1.650 V / 1.650 V	1.60 to 1.70 V	1.650 V

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
B at 35 °C: R_t / V_s / V_{out}	6.51 kΩ / 1.999 V / 2.237 V	2.18 to 2.30 V	2.237 V
B at 40 °C: R_t / V_s / V_{out}	5.30 kΩ / 2.157 V / 2.501 V	2.44 to 2.56 V	2.501 V
B at 50 °C: R_t / V_s / V_{out}	3.59 kΩ / 2.429 V / 2.958 V	2.90 to 3.02 V	2.958 V
B: 12-bit ADC codes (3.3 V) at 0, 25, 50 °C	184, 2048, 3671	within about ±60 counts	(from V_{out})
B: V_{out} change per kelvin at 25 °C	61.6 mV/K	about 55 to 68 mV/K	61.6 mV/K

Real readings spread because resistors are 1 % parts, the thermistor has a 1 % tolerance on R_0 and 1 % on B (about 0.5 °C of error away from 25 °C), the LDR varies by 2:1 from sample to sample, and the chip has a small offset. What does not vary is the structure: the comparator has two distinct thresholds 0.33 V apart; U2B's output equals the 1.65 V reference; V_{out} equals V_o when V_s equals V_o (25 °C); and V_{out} rises steadily with temperature, always inside 0 to 3.3 V.

10. Challenges

CHALLENGE 1: QUANTISATION AND AMPLIFIER GAIN

(a) Find the LSB of a 10-bit converter on a 5 V reference, a 12-bit on 3.3 V, and a 16-bit on 3.3 V. For the 12-bit converter find the code for 1.000 V, the voltage that code stands for, the rms quantisation noise and the ideal SNR. (b) A bridge sensor gives 0 to 80 mV. Using R_g = 1 kΩ choose R_f from the E12 series so that the non-inverting amplifier gives the largest gain that keeps the output under 3.3 V, find the full-scale output, the sensor-side resolution (LSB divided by the gain) with the 12-bit converter, the exact closed-loop gain if the op-amp's open-loop gain is 100,000, and the closed-loop bandwidth and gain at 10 kHz for GBW = 1 MHz. (c) A summing amplifier with R_f = 10 kΩ has inputs V_1 = 1.0 V (10 kΩ), V_2 = 0.5 V (20 kΩ), V_3 = 2.0 V (10 kΩ): find V_{out}. Then choose equal input resistors so that it gives minus the average of the three inputs.

View solution

CHALLENGE 2: COMPARATORS AND HYSTERESIS

(a) A rail-to-rail comparator on 5 V has V_{ref} = 2.5 V, R_1 = 10 kΩ and R_f = 220 kΩ. Find V_H, V_L and the hysteresis. (b) Design a fan switch with the stage B thermistor divider (NTC 10 kΩ, B = 3950, on top of 10 kΩ, 3.3 V) that turns the fan ON at 40 °C and OFF at 35 °C: find the thermistor resistances and the divider voltages at the two temperatures, the ratio R_1/R_f and E12 values (R_1 = 10 kΩ), the required V_{ref} and a divider of two resistors from 3.3 V that makes it, and the actual switching temperatures you end up with. (c) In stage A change R_f from 100 kΩ to 47 kΩ: find the new V_H, V_L, hysteresis and the LDR resistances at which the LED switches, and say whether a 50 mV peak-to-peak flicker from a mains-powered lamp could still cause false switching with each choice.

View solution

CHALLENGE 3: SENSORS AND THE SIGNAL CHAIN

(a) A 350 Ω strain gauge (GF = 2.0) in a quarter bridge with 3.3 V excitation sees 2000 microstrain: find the resistance change and the bridge's output voltage (exact and approximate). For an INA333 (G = 1 + 100 kΩ/R_G) with its REF pin at 1.65 V, find the gain and E12 value of R_G that give +1.5 V at 2000 microstrain, then the output at 600 microstrain. (b) A low-side current sensor uses R_s = 0.05 Ω for currents up to 3 A: find the shunt voltage and power, choose a non-inverting amplifier (R_g = 1 kΩ, R_f to be chosen from the E24 series) that keeps 3 A below 3.3 V, and find the offset error for V_{OS} = 2 mV in amps. (c) A signal is sampled at 2 kHz. Design an RC anti-alias filter (R = 1 kΩ, find C) with a corner at about 160 Hz; find the alias frequency and attenuation of a 1.7 kHz noise tone of 100 mV peak-to-peak and its size in ADC counts (12-bit, 3.3 V). Then find the highest frequency at which an MCP6002 (SR 0.6 V/µs) can reproduce a 3 V peak-to-peak sine, and the lower limit for a gain-40 stage.

View solution

QUICK REFERENCE

- **ADC:** LSB = $V_{ref}/2^N$; code = floor(V_{in}/LSB); rms quantisation noise = LSB/√12; SNR = 6.02N + 1.76 dB; sample at least twice the highest frequency, and filter before the converter (alias = $|f - k f_s|$)
- **Op-amp rules** (with negative feedback): V+ = V− and no input current; open-loop A is large and the closed-loop gain = A/(1+Aβ), about 1/β
- **Follower:** gain 1; **non-inverting:** 1 + R_f/R_g; **inverting:** $-R_f/R_{in}$ (virtual ground); **summing:** $-R_f\,\Sigma(V_i/R_i)$; **difference:** $(R_2/R_1)(V_2 - V_1)$, CMRR about (1 + G)/(4 t)

- **Comparator with hysteresis:** $V_H = V_{ref}(1 + R_1/R_f) - V_l\, R_1/R_f$, $V_L = V_{ref}(1 + R_1/R_f) - V_h\, R_1/R_f$; hysteresis = $(V_h - V_l)R_1/R_f$
- **Speed:** bandwidth = GBW/G; full-power bandwidth = $SR/(2\pi V_p)$; single supply needs a rail-to-rail chip if the output must reach V+ (an LM358 stops 1.5 V short)
- **Thermistor:** $T = 1/(1/T_0 + \ln(R/R_0)/B)$ in kelvin; $dR/dT = -BR/T^2$; divider with $R_b = R$ at the middle temperature; watch self-heating
- **Bridge and in-amp:** $V_{diff} = V_{ex}\, x/(2(2 + x)) \approx V_{ex}\, x/4$; in-amp gain = $1 + 2R/R_G$ (INA333: $1 + 100\ k\Omega/R_G$)
- **Shunt:** $V = I\, R_s$, power $I^2\, R_s$, amplify with gain $1 + R_f/R_g$; offset error = $V_{OS}/(R_s \times G)$ in amps
- **Noise:** 100 nF at every supply pin plus 10 µF per board; star ground; twisted pair and a one-ended shield; RC anti-alias filter $f_c = 1/(2\pi RC)$; average N readings for $\sqrt{N}$ less noise; average over 20 ms for 50 Hz hum

Next, Basic Electronics 8 moves from the smooth world of analogue voltages to the two-level world of digital logic: logic levels, gates, latches and counters, and the 555 timer.

Basic Electronics

Course 1 · Chapter 8 · Digital Logic Basics

Chapter 7 dealt with signals that can take any voltage. This chapter is about the opposite: signals that are deliberately only ever one of two things, **LOW** or **HIGH**, 0 or 1. Everything inside a microcontroller works this way, and so do the small logic chips that sit around it on a robot: chips that combine switch signals, remember a button press, count wheel pulses, turn three pins into sixteen, or time a pulse with no program at all. A digital signal is tough: noise smaller than the gap between the two levels vanishes without trace, which is why a computer can copy a file perfectly while a tape cannot. The chapter uses the divider and pull-up ideas of Basic Electronics 3, the RC timing of Basic Electronics 4, the transistor switch of Basic Electronics 6 and the comparator with hysteresis of Basic Electronics 7. The *mathematics* of logic (Boolean algebra, simplification, adders) is a separate course, Boolean Algebra & Digital Logic; here we look at the *electrical* side: voltages, currents, chips and wiring.

1. Logic levels and noise margins

A digital input does not compare against exactly 2.5 V. Each chip family guarantees that a voltage *below* V_{IL} is read as LOW, a voltage *above* V_{IH} is read as HIGH, and that anything in between is undefined (it may read either, or oscillate). A driving output guarantees the opposite promise: its LOW is *at most* V_{OL} and its HIGH is *at least* V_{OH}, when supplying the rated current. For two chips to talk, the output's promise must fit inside the input's requirement; the room left over is the **noise margin**.

```
VIL = highest voltage an input still reads as LOW        VIH = lowest voltage an input r
VOL = highest voltage an output gives for LOW           VOH = lowest voltage an output g

Noise margin, LOW  :  NML = VIL - VOL          (how much noise a LOW can pick up before
Noise margin, HIGH :  NMH = VOH - VIH
Both must be positive for the connection to work at all. A negative number means a guar

Voltage levels on one wire (5 V CMOS, 74HC):

  5.0 V  ---------------------------------   supply
  4.4 V  ---------------------------------   VOH  (an output HIGH is at least this)
           NMH = 0.9 V
```

```
3.5 V  ---------------------------------   VIH  (an input reads HIGH above this)
            undefined region
1.5 V  ---------------------------------   VIL  (an input reads LOW below this)
        NML = 1.4 V
0.1 V  ---------------------------------   VOL  (an output LOW is at most this)
0.0 V  ---------------------------------   ground
```

FAMILY / PIN	SUPPLY	V_{IL} MAX	V_{IH} MIN	V_{OL} MAX	V_{OH} MIN	NM_L	NM_H
74LS (old TTL)	5 V	0.8 V	2.0 V	0.5 V	2.7 V	0.3 V	0.7 V
74HC (CMOS)	5 V	1.5 V (0.3 V_{CC})	3.5 V (0.7 V_{CC})	0.1 V	4.4 V	1.4 V	0.9 V
74HCT (CMOS, TTL inputs)	5 V	0.8 V	2.0 V	0.1 V	4.4 V	0.7 V	2.4 V
74HC (CMOS)	3.3 V	0.99 V	2.31 V	0.1 V	3.2 V	0.89 V	0.89 V
LVTTL (3.3 V logic, 74LVC)	3.3 V	0.8 V	2.0 V	0.4 V	2.4 V	0.4 V	0.4 V
ESP32 GPIO pin	3.3 V	0.825 V (0.25 V_{DD})	2.475 V (0.75 V_{DD})	0.33 V (0.1 V_{DD})	2.64 V (0.8 V_{DD})	0.50 V	0.17 V
ATmega328P pin (Uno, Nano), input only	5 V	1.5 V (0.3 V_{CC})	3.0 V (0.6 V_{CC})	—	—	—	—

These figures are typical datasheet values at light load (74HC outputs quoted at 20 µA; at 4 mA, V_{OL} rises to about 0.26 V and V_{OH} falls to about 3.8 V). The datasheet of the exact part is the authority; the method is what matters.

Checking a connection: the two inequalities

```
A driver output can feed a receiver input only if   VOH(driver) >= VIH(receiver)   AND
and the output voltage never exceeds what the input tolerates (usually VCC + 0.3 V).

Worked examples:
  74LS output to a 74HC input (both 5 V):   VOH = 2.7 V is below VIH = 3.5 V        FAI
  74HC (5 V) to a 74LS input:               4.4 V > 2.0 V and 0.1 V < 0.8 V           
  ESP32 pin (3.3 V) to a 74HC at 5 V:       VOH = 2.64 V is below VIH = 3.5 V       FA
```

```
ESP32 pin to a 74HCT at 5 V:      2.64 V > 2.0 V, NMH = 0.64 V; 0.33 V < 0.8 V
74HC at 5 V to an ESP32 pin:      4.4 V is more than 3.3 V + 0.3 V = 3.6 V  DA
74HC at 3.3 V to an ESP32 pin:    VOH about 2.9 V (at 4 mA) is above VIH = 2.4
```

WHY THE NOISE MARGIN IS NOT A SPARE "EXTRA"

The margin has to cover everything that gets onto the wire: the ground-voltage difference between two boards, crosstalk from a neighbouring wire, and the supply ripple of Basic Electronics 7. Half a volt sounds ample until a motor starts on the same ground wire. A margin that is only 0.17 V (the ESP32 HIGH into a 2.475 V threshold) works on the bench and fails on the robot; the cure is a shorter wire, a better ground or a buffer chip.

2. Gates and truth tables

A **gate** is the smallest logic circuit: one output that is a fixed function of one or two inputs. The function is written as a **truth table**, which lists the output for every input combination (Boolean Algebra & Digital Logic 2 covers the operators and truth tables in depth; Boolean Algebra & Digital Logic 6 covers gates and how to combine them). The six you need every day:

A	B	AND	OR	NAND	NOR	XOR	XNOR	NOT A
0	0	0	0	1	1	0	1	1
0	1	0	1	1	0	1	0	1
1	0	0	1	1	0	1	0	0
1	1	1	1	0	0	0	1	0

GATE	IN WORDS	EXPRESSION	74HC CHIP (4 OR 6 GATES)	A ROBOT USE
NOT (inverter)	Output is the opposite of the input	Y = Ā (A with a bar)	74HC04 (6)	Turn an active-low sensor into an active-high signal
AND	Output HIGH only if both inputs are HIGH	Y = A · B	74HC08 (4)	Run the motor only if the enable switch AND the bumper-clear signal are both true

GATE	IN WORDS	EXPRESSION	74HC CHIP (4 OR 6 GATES)	A ROBOT USE
OR	HIGH if either input is HIGH	Y = A + B	74HC32 (4)	Stop if the left OR the right bumper is hit
NAND	The opposite of AND: LOW only if both inputs are HIGH	$Y = \overline{(A \cdot B)}$	74HC00 (4)	The universal building block (below); also latches
NOR	The opposite of OR: HIGH only if both inputs are LOW	$Y = \overline{(A + B)}$	74HC02 (4; its pins differ)	"Nothing detected": HIGH when all sensors are quiet
XOR	HIGH if the inputs differ	$Y = A \oplus B$	74HC86 (4)	Compare two bits; the heart of an adder; detect that two switches disagree

Bubbles, bars and active-low names

A small circle on a gate symbol means *inversion*; a bar over a signal name (or a trailing `#` or `_N` or `n`) means *active low*: the signal does its job when it is LOW. A chip's **$\overline{\text{CS}}$** (chip select), **$\overline{\text{RESET}}$** and **$\overline{\text{OE}}$** (output enable) are almost always active low, because open-drain wiring (section 5) and old TTL both pull more strongly low than high. When reading a datasheet, the name tells you which level means "yes".

Why NAND is called universal

Any logic function can be built from AND, OR and NOT. A NAND gate can make all three, so a pile of NAND gates (a single 74HC00 holds four) can build *any* logic circuit; NOR has the same property. This is why chips and old computers were designed around one gate type.

```
Using only NAND gates   (N(a, b) means NAND of a and b)

  NOT A  = N(A, A)                          tie the two inputs together: 1 gate
  AND    = N( N(A,B), N(A,B) )              a NAND followed by a NOT: 2 gates
  OR     = N( N(A,A), N(B,B) )              invert both inputs, then NAND: 3 gates
  XOR    = N( N(A, M), N(B, M) )  with  M = N(A, B)       4 gates, exactly one 74HC00

Why OR works (De Morgan's law, Boolean Algebra & Digital Logic 3):   NOT( NOT(A) AND NO
```

```
Check the XOR for A = 1, B = 0:   M = N(1,0) = 1;   N(A,M) = N(1,1) = 0;   N(B,M) = N(0
Check A = 1, B = 1:               M = N(1,1) = 0;   N(1,0) = 1;   N(1,0) = 1;   N(1,1) =
```

3. Gates from transistors

A gate is only a few switches. Chapter 6 showed that a transistor is a switch that a voltage on its input can turn on or off; a logic gate is transistors arranged so that the output is pulled to LOW or HIGH by the right input combination.

Discrete NPN gates

The first logic circuits used bipolar transistors and resistors. With one NPN transistor and a pull-up resistor you have a **NOT** gate (input HIGH: transistor on, output pulled LOW; input LOW: transistor off, the resistor pulls the output HIGH). Put two transistors in *parallel* and the output goes LOW if either is on: a **NOR**. Put them in *series* and the output goes LOW only if both are on: a **NAND**. Build 1 builds the NAND.

```
NOR: two NPNs in PARALLEL (Rc = collector pull-up resistor; B, C, E = base, collector, 

+5 V o------+
            |
        [ Rc ]
            |
            +------------+---------------o Y
            |            |
            C            C
A o-[ R ]-B Q1      B Q2 -[ R ]-o B
            E            E
            |            |
            +-----+------+
                  |
0 V o-------------+

Either transistor, when on, pulls Y LOW.   Y is HIGH only if A and B are both LOW.   (N

NAND: two NPNs in SERIES (this is Build 1)

+5 V o------+
            |
        [ Rc ]
```

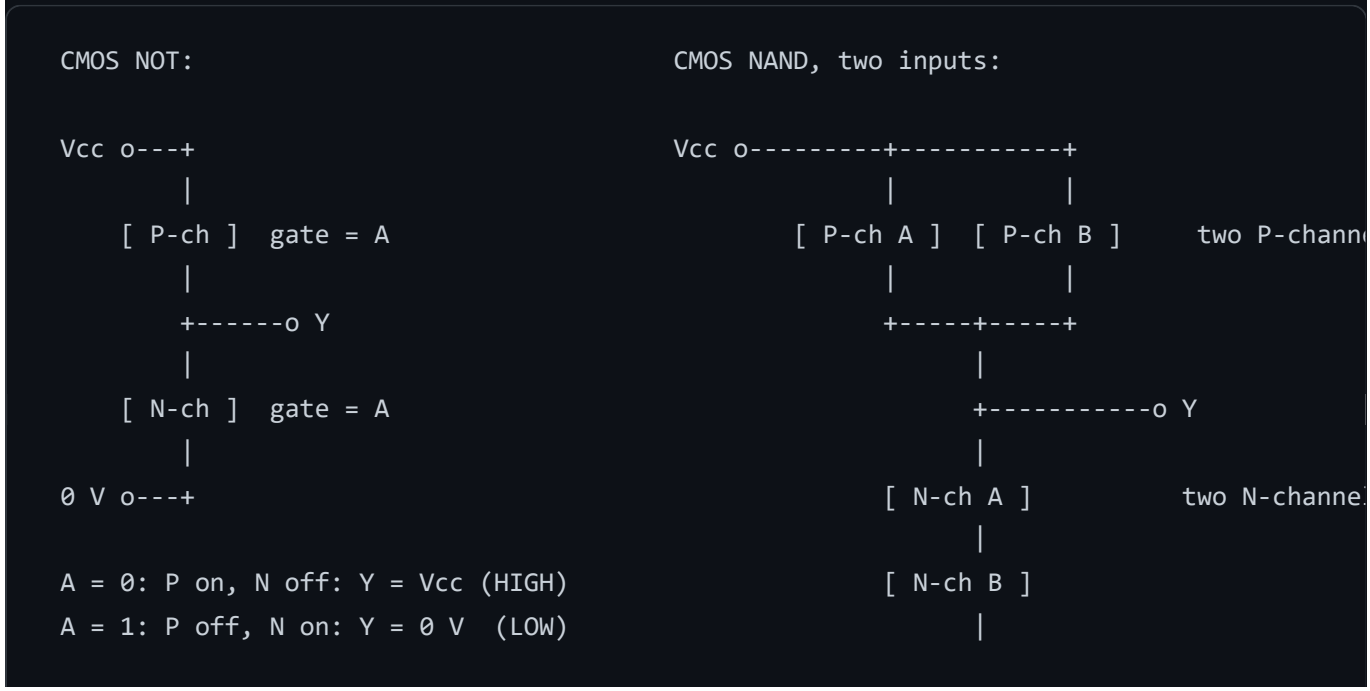
```
            |
            +----------------o Y
            |
            C
A o-[ R ]-B Q1
            E
            |
            C
B o-[ R ]-B Q2
            E
            |
0 V o-------+

Y falls only when BOTH transistors conduct.   Y is LOW only if A and B are both HIGH.
```

Why this was replaced: the resistor conducts the whole time the output is LOW. With Rc = 1 kΩ on 5 V, that is (5 − 0.2)/1 kΩ = 4.8 mA *per gate* while LOW, so 100 gates draw half an ampere. A discrete gate also has a weak HIGH (it relies on a resistor that is slow when it must charge a wire's capacitance) and a switching threshold of about 0.7 V (one base-emitter drop), so noise margins are poor.

CMOS: complementary MOSFETs

Modern logic (the 74HC family and everything inside a microcontroller) uses pairs of MOSFETs, one **P-channel** device that conducts when its gate is *LOW* and one **N-channel** that conducts when its gate is *HIGH*. Wired complementary, one is always off:



```
                                                  0 V o-------+
  gates:  P-ch A and N-ch A are driven by A;  P-ch B and N-ch B are driven by B.
  A = B = 1:  both N on, both P off:  Y = LOW.     Any input LOW:  that P is on, a series
```

PROPERTY	RESISTOR-TRANSISTOR GATE	CMOS GATE (74HC)
Current when sitting still	Milliamps while the output is LOW	Nanoamps: one transistor of each pair is always off, so no path from Vcc to ground
Output HIGH	Through a resistor: weak, slow	Through a P-channel switch: strong, equal to the LOW
Input threshold	About 0.7 V, not adjustable	About half the supply (2.5 V at 5 V): symmetrical noise margins
Input current	Base current (hundreds of µA)	About 1 µA leakage (a MOSFET gate is insulated): one output can drive dozens of inputs
Weakness	Power waste	Draws current only when switching, so power rises with frequency (below); static electricity can punch through the gate oxide

```
  Dynamic power of a CMOS gate:     P = Cpd x Vcc^2 x f          + the power spent charg
    Cpd = the chip's "power dissipation capacitance" from the datasheet (about 20 pF per

  Example: one gate, 5 V, 1 MHz:   P = 20e-12 x 5^2 x 1e6 = 0.5 mW          (current 0.1
  Example: the same gate at 10 MHz = 5 mW: 10 times the frequency, 10 times the power
  Example: at 3.3 V, 1 MHz:        P = 20e-12 x 3.3^2 x 1e6 = 0.22 mW        (power goes

  Compare 74LS: about 2 mW per gate at ANY frequency, even when idle.
```

4. Using the 74HC series

The **74HC** chips are the standard small logic parts for hobby work: cheap, widely available in DIP (breadboard) packages, and with a supply range of 2 V to 6 V, so one chip serves a 5 V Arduino and a 3.3 V Pico. The 74HC chips are numbered by function (74HC00 is four NAND gates whichever maker sells it). Pinouts follow a pattern: the supply on the top corner, ground on the

bottom corner, gate inputs and outputs in order. Read the datasheet before wiring a chip you have not used (the 74HC02 NOR chip, for example, puts the output *first* on each gate).

```
74HC00 (four NAND gates) and 74HC08 (AND), 74HC32 (OR), 74HC86 (XOR) all share this DIP

       pin 14 VCC   13 4B      12 4A      11 4Y      10 3B      9 3A       8 3Y
       pin  1 1A     2 1B       3 1Y       4 2A       5 2B       6 2Y       7 GND

Gate 1: inputs 1A, 1B (pins 1, 2), output 1Y (pin 3).   Gate 2: pins 4, 5 -> 6.   Gate
74HC04 and 74HC14 (six inverters) use the same power pins: 1A 1, 1Y 2, 2A 3, 2Y 4, 3A 5
```

The rules that keep a 74HC chip working

- **Never leave an input floating.** A CMOS input has almost no leakage, so an unconnected pin floats to a random voltage picked up from the air (50 Hz hum, your hand), sits near the switching threshold where both transistors conduct (the chip then draws extra supply current and may get warm), and the output chatters. **Tie every unused input to GND or VCC**, and tie unused gates' inputs too (an unused gate is a gate with an unconnected input). Leave unused *outputs* unconnected.
- **Decouple every chip:** 100 nF ceramic across VCC and GND, as close to the chip's pins as the breadboard allows (Basic Electronics 4). When all the output transistors switch together, they pull a brief spike of current. Without the capacitor the spike drops the supply voltage by tens of millivolts or more, and other chips glitch.
- **Respect the supply range.** 74HC chips run from 2 to 6 V and are damaged above about 7 V. 74HCT is 4.5 to 5.5 V only. A 74HC chip powered from 3.3 V will not understand a 5 V input and may be destroyed by it (section 10).
- **Never tie two normal outputs together.** If one drives HIGH and another LOW the two outputs fight: the current is about 5 V / (60 Ω + 60 Ω) = 40 mA, which exceeds the rating and can destroy both. Where several devices must share one wire, use open-drain outputs (section 5) or a tri-state bus.
- **Keep input edges fast.** A plain 74HC input needs the voltage to cross the threshold quickly (the datasheet specifies a rise time of 500 ns or less at 4.5 V). A slow edge, such as the output of an RC network or a human-speed switch, lingers in the undefined region and gives multiple output pulses; this is the job of the Schmitt-trigger chips in section 6.

Fan-out and propagation delay

```
Fan-out = how many inputs one output can drive.   For CMOS-to-CMOS the DC current is on
  each input adds about 3.5 pF to the output's load; the wire adds about 10 pF per 10 c

An output works as a resistor of about Rout = VOL / IOL = 0.26 V / 4 mA = 65 ohm   (74H
It must charge the load capacitance, so the extra delay is about   t = 0.69 x Rout x C(

Example: one output drives 10 inputs plus 10 cm of wire:   C = 10 pF + 10 x 3.5 pF = 45
  extra delay = 0.69 x 65 ohm x 45 pF = 2.0 ns          (tiny: ten inputs is a perfectl
A long ribbon cable of 200 pF would give 0.69 x 65 x 200 pF = 9.0 ns: that is when you

Propagation delay tpd = the time from an input crossing its threshold to the output cro
  74HC00 at 5 V: about 8 to 10 ns typical;  at 3.3 V: about 12 to 16 ns  (the lower sup
  The delay limits the speed and gives hazards: when a signal takes two paths of differ
  the output may flicker briefly ("glitch") before it settles. Inputs that change one a
  through the SAME number of gates as its data, or use a flip-flop (section 7), which s

Example, a ring oscillator: three inverters in a loop (Y of the last goes back to the f
  period = 2 x 3 x 10 ns = 60 ns      f = 1 / 60 ns = 16.7 MHz     (an odd number of in
```

FAMILY	SUPPLY	INPUT THRESHOLDS	TYPICAL DELAY	OUTPUT DRIVE (RATED)	USE
74HC	2 to 6 V	0.3 and 0.7 of V_{CC}	8 ns at 5 V	±4 mA	The general choice; 3.3 V and 5 V projects
74HCT	4.5 to 5.5 V	TTL: 0.8 V and 2.0 V	10 ns	±4 mA	A 5 V chip that reads 3.3 V signals (a one-chip level shifter, up only)
74AHC / 74AHCT	2 to 5.5 V	as HC / as HCT	4 to 5 ns	±8 mA	Faster HC / HCT replacement; LED-strip level shifting (74AHCT125)
74LVC	1.65 to 3.6 V	LVTTL, inputs tolerate 5.5 V	3 to 5 ns	±24 mA	3.3 V logic that is safe to feed from 5 V (down-shifting)
74LS	5 V only	TTL	10 ns	4 to 8 mA low	Old, power-hungry; look-alike parts in old kits
CD4000	3 to 15 V	0.3 and 0.7 of V_{CC}	50 to 100 ns at 5 V	about 1 mA	Slow, but runs directly from a 9 V or 12 V battery

Typical figures; the datasheet of each part decides.

5. Pull-ups, pull-downs and open-drain outputs

Floating inputs and the switch

Basic Electronics 3 introduced the pull-up and pull-down resistor. It exists because a switch connects an input to only *one* of the two levels. With a button between the input and ground and nothing else, the input floats when the button is released. A 10 kΩ pull-up resistor to VCC supplies the other level:

```
Pull-up (the usual way, buttons to ground):         Pull-down (buttons to VCC):

+5 V o----+                                    +5 V o----+
          |                                              |
     [ R 10 k ]                                     [ SW ]
          |                                              |
          +------o to the input                          +------o to the input
          |                                              |
       [ SW ]                                       [ R 10 k ]
          |                                              |
0 V o-----+                                      0 V o---+

Released: no current in R (input draws about 1 uA), so the input sits at 5 V - 10 k x 1
Pressed:  the input is 0 V and the resistor wastes   I = 5 V / 10 k = 0.5 mA   only whi

Choosing R:  too small wastes current when pressed; too large makes the HIGH level slow
  Rise time of the node =  2.2 x R x C    (10 % to 90 %)       C = input + wire, 20 pF:
  A long wire of 500 pF (a bumper switch on the end of a cable):   2.2 x 10 k x 500 pF =
  Microcontroller boards have internal pull-ups of 20 to 50 k (INPUT_PULLUP in Arduino
```

Open-drain and open-collector outputs

A normal (totem-pole or push-pull) output has two switches: it drives HIGH *and* LOW. An **open-drain** output (MOSFET, as in 74HC03 or an I2C pin) or **open-collector** output (bipolar, as in an LM393 comparator, from Basic Electronics 7) has only the *lower* switch: it can pull the wire to ground or let go, and a **pull-up resistor** to a supply does the HIGH. Because the output cannot actively drive HIGH, many open-drain outputs can safely share one wire:

```
+5 V o-----+
           |
      [ Rp 3.3 k ]                        A    B    wire
           |                              0    0     0        (0 = that output is pull
           +----------+----------+----o   0    1     0        (1 = that output has let
           |          |          |      1    0     0
       [ out A ]  [ out B ]  [ out C ]   1    1     1
           |          |          |
0 V o------+----------+----------+

The wire is HIGH only if EVERY output lets go, so the wire = A AND B AND C:  a "wired-A
Read it as active-low: the wire goes LOW when ANY device pulls it low.   One pin can th
shared by many chips, and I2C's SDA and SCL lines work exactly this way, which also all

Choosing the pull-up Rp   (5 V, wire capacitance 200 pF, each output can sink IOL = 3 m
  Lower limit:  the output must be able to hold the wire LOW against the pull-up:   Rp(
  Upper limit:  the wire must rise fast enough:  rise time from 30 % to 70 % of Vcc  t
       for t at most 1 us (I2C at 100 kHz):   Rp(max) = 1 us / (0.8473 x 200 pF) = 5.9
  Choose 3.3 k:   t = 0.8473 x 3.3 k x 200 pF = 0.56 us      current when pulled low =
```

OPEN-DRAIN ALSO GIVES YOU FREE LEVEL SHIFTING

Because an open-drain output only pulls to ground, the pull-up can go to *any* voltage that the receiving chip accepts: pull the same wire up to 3.3 V for a 3.3 V board or to 5 V for a 5 V board. Section 10 uses a MOSFET to make this work in both directions.

6. Schmitt-trigger inputs and debouncing

Basic Electronics 7 added hysteresis to a comparator with a resistor from output to input. A **Schmitt-trigger input** builds the same idea into a logic gate: the input has two thresholds, V_{T+} for a rising signal and V_{T-} for a falling one. Once the output has switched, the input must travel the whole gap back before it switches again, so a slow or noisy edge produces one clean transition. The 74HC14 is a hex inverter with Schmitt inputs; 74HC132 is the NAND version.

```
74HC14 at 5 V, typical:   VT+ = 2.4 V (rising),   VT- = 1.4 V (falling),   hysteresis =

Example: a slow input ramps from 0 V to 5 V with 0.5 V of noise riding on it.
  Plain 74HC04 (one threshold, about 2.5 V):  the noise crosses 2.5 V several times, so
```

```
   74HC14:  the output switches once when the input rises past 2.4 V, and cannot switch
            Noise of 0.5 V cannot bridge the 1.0 V gap, so there is exactly one clean ed

  Use it for: a slow RC signal (below), a push button, a sensor with a slowly changing ou
```

Debouncing

A mechanical switch does not close cleanly. The contacts bounce, making and breaking for 1 to 10 ms (sometimes 20 ms) and producing a burst of pulses that a fast chip counts as many presses. Three cures:

1. **Software:** ignore further changes for about 20 ms after the first one. Free, and usual with a microcontroller.
2. **RC filter plus a Schmitt gate** (Basic Electronics 4 used the RC; here is the gate that makes it work):

```
  +5 V o-----+
             |
        [ R 10 k ]
             |
             +----------+-----------+----o to 74HC14 pin 1 (1A);   output 1Y (pin 2) is
             |          |           |
          [ SW ]    [ C 1 uF ]     (the node N)
             |          |
        [ Rs 100 ]      |
             |          |
  0 V o------+----------+

  Switch released: C charges through R toward 5 V.       Time to reach VT+ = 2.4 V:    t
        t = - 10 k x 1 uF x ln( 1 - 2.4/5 ) = 10 ms x 0.654 = 6.5 ms          (with C =
  Switch pressed:  C discharges through Rs (100 ohm) in   t = - Rs C ln( VT-/Vcc ) = 100
  A bounce during release lets C charge only while the contacts are open.   A gap of 1 ms
    far below VT+, and the next closure discharges it again: the output never moves.
  The delay is the price: 6.5 ms after the switch opens.   Rs is optional; it limits the
```

3. **A latch** with a two-way (changeover, SPDT) switch: the first contact closing sets the latch, and bounce on the same contact changes nothing. See the SR latch in section 7. This is how the earliest hardware did it, and it needs no timing at all.

7. Latches and flip-flops: one bit of memory

All the circuits so far are **combinational**: the output depends only on the inputs *now*. A **sequential** circuit remembers: its output depends on what happened before. The smallest memory is two gates that feed each other (Boolean Algebra & Digital Logic 8 builds flip-flops and memory in logic terms).

The SR latch

```
Two NAND gates, each output fed into the other gate's input (a 74HC00; the inputs are a

  gate 1:   Q     = NAND( S-bar, Q-bar )        inputs: S-bar and the output of gate 2
  gate 2:   Q-bar = NAND( R-bar, Q )            inputs: R-bar and the output of gate 1

  (draw it as two NAND symbols one above the other; the output of each runs back, cross

  S-bar  R-bar  |  Q     Q-bar  | What happens
    1      1    | holds  holds  | Remembers the last state
    0      1    |  1       0    | SET: Q goes to 1
    1      0    |  0       1    | RESET: Q goes to 0
    0      0    |  1       1    | Not allowed: both outputs HIGH; on releasing both tog

Why it remembers: with both inputs HIGH, each NAND acts as an inverter of the other outp
Pin wiring on a 74HC00:  S-bar -> pin 1 (1A);   Q-bar (pin 6) -> pin 2 (1B);   output Q
                         R-bar -> pin 4 (2A);   Q (pin 3) -> pin 5 (2B);        output (
```

With a changeover switch joined to ground, one throw to $\bar{S}$ and the other to $\bar{R}$ (each with its pull-up), the latch is a perfect debouncer: the first touch of a contact flips it and the later bounce changes nothing, because the other contact is already open and the first can only re-assert what it has set.

The D flip-flop and the 74HC74

A latch changes whenever its inputs do. A **flip-flop** samples its input only at an *instant*: the rising edge of a clock input. The **D flip-flop** is the commonest: on each rising clock edge, Q takes the value that D has at that moment, and keeps it until the next edge. Because the output changes at one instant, everything driven by the same clock changes together, and the glitches of section 4 are filtered out.

```
74HC74: two D flip-flops, DIP-14, notch at the left, pins numbered anticlockwise from p

      pin 14 VCC    13 2RD-bar   12 2D     11 2CP     10 2SD-bar     9 2Q      8 2Q-ba
      pin  1 1RD-bar  2 1D        3 1CP     4 1SD-bar    5 1Q        6 1Q-bar   7 GND

  D = data;   CP = clock (rising edge);   SD-bar = asynchronous SET (active low);   RD-
  The SET and RESET inputs act at once and override the clock: tie both HIGH (to VCC) w

  On each rising edge of CP:   Q becomes D;   Q-bar is its opposite.

  Timing:   setup time tsu = D must be steady this long BEFORE the clock edge (about 20
            hold time  th  = D must stay steady this long AFTER the edge (about 0 to 5
            clock-to-output delay tpd about 15 ns;   maximum clock frequency: tens of M
  Violate setup or hold and the flip-flop may land half-way and take a long time to set
  This is why a signal that arrives at random times, like a button, is re-timed through
```

Toggle: divide a frequency by two

```
Connect Q-bar back to D.  At each rising clock edge, Q takes the value of D = the OLD Q

  clock rising edge number:   1    2    3    4    5    6 ...
  Q after that edge:          1    0    1    0    1    0 ...       (starting from Q = 0
  Q makes one full cycle (HIGH then LOW) for every TWO clock cycles:    f(Q) = f(clock)

Chain n toggle stages (Q of one into the clock of the next):   f(out) = f(clock) / 2^n
  1 kHz clock:  500 Hz, 250 Hz, 125 Hz, 62.5 Hz after stages 1, 2, 3, 4.      The 555 +
```

8. Counters and shift registers: more outputs and inputs from few pins

The counter: 74HC161

A **counter** is a row of flip-flops that counts clock pulses in binary. The 74HC161 is a 4-bit **synchronous** counter (all four flip-flops share the clock, so the outputs change together and do not glitch through wrong codes); the cheaper **ripple** counters like the 74HC4040 clock each stage from the previous one, so the outputs briefly show wrong codes while the edge ripples through (about 10 to 15 ns per stage).

```
74HC161, DIP-16, notch at the left:

      pin 16 VCC   15 TC    14 Q0    13 Q1    12 Q2    11 Q3    10 CET    9 PE-bar
      pin  1 MR-bar  2 CP    3 D0     4 D1     5 D2     6 D3     7 CEP     8 GND

  CP = clock (rising edge)   MR-bar = asynchronous clear (LOW clears to 0000)   PE-bar =
  CEP, CET = count enables: both HIGH to count.   TC = terminal count: HIGH when the cou

  Count sequence: 0000, 0001, 0010 ... 1111, then 0000.      f(Q0) = f/2,  f(Q1) = f/4,
  Wire for plain counting: MR-bar, PE-bar, CEP, CET to VCC.  Cascading to 8 bits: TC of

  Robot use: a wheel encoder gives 20 kHz pulses at full speed, too fast to count in so
  Count them in hardware or divide by 16:   20 kHz / 16 = 1.25 kHz at Q3, which the prog
  To count to N other than 16, decode N and clear: a decade counter (0 to 9) is made wi
  with both Q3 and Q1 HIGH.  The clear is asynchronous, so the count of 10 appears for
  PE-bar = NAND(Q3, Q0) and D0 to D3 tied LOW, so the next clock loads 0000 instead of
```

The shift register: more outputs (74HC595)

A microcontroller never has quite enough pins. A **serial-in, parallel-out shift register** turns **three** output pins into **eight** (or sixteen, or sixty-four, by chaining): bits are pushed in one at a time on a data line, and then all eight leave together on the outputs.

```
74HC595, DIP-16, notch at the left:

      pin 16 VCC   15 Q0   14 DS    13 OE-bar   12 STCP   11 SHCP   10 SRCLR-bar   9 Q'
      pin  1 Q1      2 Q2    3 Q3     4 Q4       5 Q5      6 Q6      7 Q7           8 G

  DS (data in)   SHCP (shift clock: data shifts one place on each rising edge)   STCP (s
  OE-bar LOW enables the outputs;  SRCLR-bar HIGH = normal (LOW clears the shift regist

  To send one byte (8 bits, most significant first):  for each bit:  put the bit on DS,
  Control pins used:  DS, SHCP, STCP = 3  (plus 3.3 V or 5 V, ground).   Chain a second
  Time to update 16 outputs at a 1 MHz clock:  16 bits x 1 us = 16 us.   In Arduino:  sh

  Limits:  each output about 6 mA for guaranteed levels, and about 70 mA total through
    8 red LEDs at 6 mA = 48 mA: fine.     8 LEDs at 10 mA = 80 mA: over the limit.
```

The shift register in reverse: more inputs (74HC165)

```
74HC165, DIP-16 (parallel-in, serial-out), notch at the left:

      pin 16 VCC   15 CE-bar   14 D3   13 D2   12 D1   11 D0   10 DS (serial in)   9 Q
      pin  1 PL-bar  2 CP        3 D4    4 D5    5 D6    6 D7     7 Q7-bar

  D0 to D7: eight parallel inputs (bumper switches, each with its pull-up).   PL-bar LO
  CP: clock;  after the snapshot each rising edge moves the next bit to Q7 (pin 9);  CE
  Reading 8 inputs:   pulse PL-bar low then high,  then for each bit: read Q7, pulse CP
  Chain two chips (Q7 of the first into DS of the second, CP and PL-bar shared): 16 inp
  Unused D inputs: tie to VCC or GND. A 74HC165 input is a normal CMOS input: floating
```

9. The 555 timer

The **555** (1971 and still made) is the best-selling timing chip: two comparators, a flip-flop and a transistor in one 8-pin package, with a few resistors and a capacitor outside it. It makes a steady stream of pulses (the **astable** mode: a blinker, a clock, a PWM source) or one pulse of a set length when triggered (the **monostable** mode: a delay or a pulse stretcher). The ideas are all in this book already: a comparator (Basic Electronics 7), an RC charge curve (Basic Electronics 4) and an SR flip-flop (section 7).

```
Inside the 555 (the name comes from the three 5 k resistors):

  Vcc (8) o----+------------------------------+
               |                              |
            [ 5 k ]                         (supply to all)
               +---------- 2/3 Vcc ------ also on pin 5 (CONTROL)
            [ 5 k ]
               +---------- 1/3 Vcc
            [ 5 k ]
               |
  GND (1) o----+

  THRESHOLD (6) ------->(+)[ comparator 1 ](-)<-- 2/3 Vcc ----+
                                                              +--> RESET input of the SR
  TRIGGER (2)   ------->(-)[ comparator 2 ](+)<-- 1/3 Vcc ----+--> SET input   of the S

  RESET-bar (4): LOW forces OUT low (tie HIGH to disable)

  TRIGGER below 1/3 Vcc  => flip-flop SET:   OUT goes HIGH, discharge transistor OFF
```

```
  THRESHOLD above 2/3 Vcc => flip-flop RESET: OUT goes LOW,  discharge transistor ON (p

Pins (DIP-8, notch at the left, pin 1 at the bottom-left):
       pin 8 VCC    pin 7 DISCHARGE    pin 6 THRESHOLD    pin 5 CONTROL
       pin 1 GND    pin 2 TRIGGER      pin 3 OUT          pin 4 RESET-bar
```

Astable operation

```
+Vcc o----+--------------------+------ pin 8 (VCC)  and  pin 4 (RESET-bar)
          |                    |
       [ R1 ]                  |          pin 5 (CONTROL): 10 nF to GND (always, wit
          |                    |
          +----- pin 7 (DISCH) |          pin 3 = the output
          |                    |
       [ R2 ]                  |
          |                    |
          +----- pin 6 (THRESH)|
          +----- pin 2 (TRIG)  |
          |
        [ C ]
          |
0 V o-----+----- pin 1 (GND)

Capacitor C charges through R1 + R2 and discharges through R2 (pin 7 pulls to ground).

  charging  (OUT HIGH):  from Vcc/3 toward Vcc, stopping at 2Vcc/3:   t_high = (R1 + R2
  discharging (OUT LOW): from 2Vcc/3 toward 0, stopping at Vcc/3:      t_low  = R2 C ln
  (ln 2 = 0.693; Vcc cancels: the frequency does not depend on the supply voltage.)

  period T = 0.693 (R1 + 2 R2) C            f = 1 / T = 1.44 / ( (R1 + 2 R2) C )

Worked example, a blinker (R1 = 1 k, R2 = 68 k, C = 10 uF):
  t_high = 0.693 x 69 k x 10 uF = 0.478 s        t_low = 0.693 x 68 k x 10 uF = 0.471 s
  duty = 69 k / 137 k = 50.4 %.  With R1 much smaller than R2 the duty is very close to

Worked example, 1 kHz at 60 % duty (C = 10 nF):    duty 0.6 requires R2 = 2 R1;   f = 1
  Choose E12 R1 = 27 k, R2 = 56 k:   f = 1.44 / (139 k x 10 nF) = 1036 Hz;   duty = 83 

To get a duty below 50 %, put a diode (1N4148) across R2, anode on the pin 7 side: the 
because of the diode's 0.5 V drop) while the discharge still goes through R2.   Build 3
and the 555 itself adds a percent or so, so a 1 Hz blinker really runs at 0.85 to 1.3 H
```

Monostable operation

```
+Vcc o----+---------------------+------ pin 8 and pin 4
          |                     |
        [ R ]                   |          pin 5: 10 nF to GND
          |                     |
          +----- pin 7 (DISCH)  |          trigger: pin 2 held HIGH by a pull-up (10 k
          +----- pin 6 (THRESH)
          |
        [ C ]
          |
0 V o-----+----- pin 1

At rest the discharge transistor holds C empty and OUT is LOW.   A falling edge on pin 2
releases C, and C charges through R toward Vcc until it reaches 2Vcc/3, when the flip-fl

  t = R C ln( (Vcc - 0) / (Vcc - 2Vcc/3) ) = R C ln 3 = 1.0986 R C         about    t =
  R = 100 k, C = 10 uF:   t = 1.1 x 100 k x 10 uF = 1.1 s        For a 2 s pulse choose

The trigger pulse must be shorter than t (a longer one just holds OUT HIGH); a re-trigge
Robot use: turn the beeper on for exactly 0.5 s after a bump, hold a motor on for a fixe
```

Which 555?

	BIPOLAR NE555 / LM555	CMOS TLC555 / LMC555 / ICM7555
Supply	4.5 to 16 V	about 2 to 15 V
Supply current	3 to 10 mA	about 0.1 to 0.2 mA
OUT HIGH at 5 V	about 3.3 V (V_{CC} minus 1.7 V): below the 3.5 V that a 74HC input needs at 5 V!	almost the full supply (4.5 to 5 V)
Output drive	200 mA: can drive a small load directly; spikes in the supply when switching (needs a 10 µF + 100 nF)	tens of mA at most; smoother supply
Top frequency	about 100 to 500 kHz	about 1 to 2 MHz
Choose	Driving a heavy load directly (a relay, a buzzer), 9 V or 12 V supply	3.3 V and 5 V logic, battery power, clocking a 74HC chip

The builds use a **TLC555** (the same pins as the NE555).

10. Level shifting between 3.3 V and 5 V

The parts of a robot do not all use the same voltage: an Arduino Uno or Nano is 5 V, but a Raspberry Pi Pico, an ESP32 and a Raspberry Pi are 3.3 V; many sensors, motor drivers and LED strips are 5 V. Joining a 3.3 V pin to a 5 V pin needs thought in *both* directions:

DIRECTION	THE PROBLEM	FIXES
5 V output to a 3.3 V input	Over-voltage: 5 V exceeds the 3.3 V + 0.3 V absolute maximum of most 3.3 V chips; the protection diodes conduct and the pin or the chip dies	Resistor divider (slow signals); 74LVC buffer powered at 3.3 V (its inputs tolerate 5.5 V); a level-shifter module; a MOSFET shifter; open-drain with a 3.3 V pull-up
3.3 V output to a 5 V input	Under-voltage: 3.3 V (2.6 V worst case) falls below the 5 V CMOS V_{IH} = 0.7 V_{CC} = 3.5 V; sometimes works, sometimes fails with the temperature, noise or the batch of chips	74HCT or 74AHCT buffer powered at 5 V (input threshold 2.0 V); a level-shifter module; a MOSFET shifter; a transistor

Down-shifting: a resistor divider

```
5 V signal o----[ R1 1 k ]----+----o to the 3.3 V input
                              |
                         [ R2 2 k ]
                              |
0 V o-------------------------+

Vout = 5 V x R2 / (R1 + R2) = 5 x 2 / 3 = 3.33 V       (a LOW stays 0 V)
Speed: the divider's resistance seen by the pin is R1 || R2 = 667 ohm;  with 15 pF of p
  rise time (10 % to 90 %) = 2.2 x 667 x 15 pF = 22 ns          so the signal can be up
Disadvantages: it only works in one direction (5 V to 3.3 V), and loads the 5 V output
```

Up-shifting and the one-chip approach

For a 3.3 V board driving a 5 V device (a motor driver, an LED strip), use a **74HCT or 74AHCT buffer powered at 5 V**: its input threshold is 2.0 V, so 3.3 V is a clean HIGH, and its output is a full 5 V. The 74AHCT125 (four buffers, with output-enable pins) is the usual chip; a 74HCT04 works too. Remember the NMH of 0.64 V: keep the wire short. For the other direction in a single

chip, a 74LVC245 (eight-bit) or 74LVC125 powered at 3.3 V reads 5 V inputs safely and outputs 3.3 V. Ready-made modules with eight channels use these chips.

Both directions at once: the MOSFET shifter

An I2C bus is open-drain and works in *both* directions on the same wire, so it needs a bidirectional shifter. One N-channel MOSFET (BSS138 or 2N7000) per line does it:

```
        3.3 V (LV)                                        5 V (HV)
           |                                                 |
         [ 10 k ]                                          [ 10 k ]
           |                                                 |
LV line o-+---------------------[ S     D ]-------------------+--o HV line
                                [ BSS138 ]
                                [   G   ]
                                    |
                              to the 3.3 V supply (LV)

N-channel MOSFET: source to the LV line, drain to the HV line, gate to 3.3 V (LV).  Its

1. Nobody pulls low:  the LV line is at 3.3 V (pull-up), so VGS = 3.3 - 3.3 = 0: the MO
2. The 3.3 V side pulls its line to 0 V:  the source drops to 0, VGS = 3.3 V (well abov
3. The 5 V side pulls its line to 0 V:  the drain is 0 V, the body diode conducts and pu
   the MOSFET turns ON and takes the LV line all the way to 0 V.
Both directions work with no direction pin.  Ready-made 4 and 8 channel modules for 3.3
(the pull-up formula of section 5 applies to each side).
```

NEVER ASSUME 5 V TOLERANCE

The pins of a Raspberry Pi, a Pico and an ESP32 are **not** 5 V tolerant (about 3.6 V is the absolute maximum). A 5 V signal that appears to work may be slowly damaging the input. When in doubt, measure the signal with a meter before connecting it, and check the datasheet's absolute-maximum table. The reverse is easier: a 5 V Arduino input usually reads 3.3 V as HIGH, since the ATmega328P's threshold is 3.0 V at 5 V, but with only 0.3 V of margin.

11. Which board for logic work?

JOB	BEST BOARD	WHY
A first robot, 74HC chips, bumper switches and LED outputs, simple code	Arduino Uno or Nano	5 V logic matches 74HC (and 74HCT) at 5 V with no shifting; rugged pins; `shiftOut()` and the SPI library drive a 74HC595 in two lines; tens of tutorials
The same chips on a 3.3 V robot	Raspberry Pi Pico	Run the 74HC chips from 3.3 V (they work from 2 V to 6 V) so no shifter is needed; the programmable I/O blocks (PIO) can shift data into a 595 or from a 165, count pulses and make exact LED-strip timing in hardware
Counting fast encoder pulses	Raspberry Pi Pico (PIO) or ESP32 (pulse-counter peripheral)	Hardware counting avoids missed pulses when the program is busy; otherwise a 74HC161 divide-by-16 in front of any board
Logic plus Wi-Fi, sensors on a bus	ESP32	3.3 V, many pins, a hardware pulse counter; the 0.17 V NM_H in section 1 is a reason to buffer 5 V devices with a 74HCT or 74AHCT chip
A full Raspberry Pi running Linux	Raspberry Pi with an external 595 / 165	Its pins are 3.3 V only and cannot make precise timing in software; offload to a 74HC595 on hardware SPI and level-shift anything that is 5 V
Timing or blinking with no processor at all	None: a 555 and a 74HC74	Cheap, instant on, no code to hang: a safe place for the watchdog or a fault-lamp of a robot

Basic Electronics 10 uses these chips from code and Basic Electronics 11 compares the boards in full.

12. Build it: a transistor NAND, a 555 blinker with a flip-flop, and a PWM motor-speed controller

Three small builds. **Build 1** is a NAND gate made from two transistors, measured against a real 74HC00. **Build 2** is a 555 blinker clocking a 74HC74 as a divide-by-two. **Build 3** is the 555 as a pulse-width-modulation (PWM) speed control for a small DC motor, switched by the logic-level MOSFET of Basic Electronics 6.

POWER AND STATIC

Builds 1 and 2 run from 5 V (a breadboard power module set to 5 V, a USB supply, or the 5 V pin of an Arduino); measure it with the meter before you connect the chips. A 74HC chip is damaged above 7 V and a TLC555 above 18 V. Put a 100 nF capacitor across the supply rails next to each chip, tie every unused input (Build 2 shows how), and connect the power last. Touch a grounded metal object before handling the CMOS chips. Build 3 uses a separate battery pack for the motor, joined only at the ground rail.

Parts list

PART	VALUE	USED IN
Q1, Q2	2N3904 NPN transistor (TO-92)	1
U3	74HC00 quad NAND, DIP-14	1 (comparison)
U1	TLC555 (or LMC555) CMOS timer, DIP-8	2, 3
U2	74HC74 dual D flip-flop, DIP-14	2
Q3	IRLZ44N logic-level N-channel MOSFET (TO-220; pins gate, drain, source from the left, label facing you)	3
Rc	1 kΩ (brown-black-red)	1; R1 in builds 2 and 3
Ra, Rb (base resistors)	10 kΩ (brown-black-orange), two	1
R2 (build 2)	68 kΩ (blue-grey-orange)	2
R3 (LED1), R4 (LED2)	2.2 kΩ (red-red-red) and 1 kΩ (brown-black-red)	2
C (build 2)	10 µF electrolytic, 16 V (+ leg to the timing node)	2
C3	10 nF ceramic (pin 5 bypass), two	2, 3
C1, C4	100 nF ceramic, decoupling (one per chip)	2
LED1, LED2	Red, 5 mm, two	2, 3
P1	100 kΩ linear potentiometer	3
R1 (build 3), Rd	1 kΩ, two	3
C (build 3)	4.7 nF ceramic or film	3
Rg, Rpd	100 Ω (brown-black-brown) and 100 kΩ	3

PART	VALUE	USED IN
D1, D2	1N4148 signal diode, two	3
D3	1N4001 (flyback diode)	3
Motor	Small 3 to 6 V hobby DC motor (about 150 mA running, 600 mA stalled) and a 4 × AA battery holder (6 V)	3
Supply	5 V (module, USB or board)	all
Breadboard, jumpers, multimeter	A meter with a duty-cycle or frequency range is useful but not required	all

Build 1: a NAND gate from two transistors

Two NPN transistors in series with a collector resistor, as in section 3. The inputs are two jumper wires that you plug into the + rail (1) or the − rail (0).

```
+5 V o------+
            |
        [ Rc 1 k ]
            |
            +----------------o OUT        (the gate output: measure it here)
            |
            C
A o-[ Ra 10 k ]-B  Q1  (2N3904)
            E
            |
            C
B o-[ Rb 10 k ]-B  Q2  (2N3904)
            E
            |
0 V o-------+

Check the transistors will saturate (both inputs HIGH = 5 V):
  Base current  Ib = (5 V - 0.7 V) / 10 k = 0.43 mA          Collector current  Ic = (5
  Forced gain = Ic / Ib = 4.8 / 0.43 = 11, far below the real gain (100 or more), so bo
  Each VCE(sat) about 0.1 V, so OUT = 0.1 V + 0.1 V = 0.2 V        Q1's base sits at (0
If either input is LOW (0 V) its transistor is OFF, no current can flow through the ser
Power wasted when OUT is LOW:  5 V x 4.8 mA = 24 mW, constantly (the weakness of sectio
```

```
Breadboard layout. Top rails: + rail = 5 V, - rail = 0 V. Columns have rows a-e joined a
TO-92 transistors: flat face towards you, legs down, left to right = emitter, base, coll
Transistors in row c:
   Q1:  emitter column 10 row c,   base column 11 row c,   collector column 12 row c
   Q2:  emitter column 16 row c,   base column 17 row c,   collector column 18 row c
Series connection:  jumper column 10 row a -> column 18 row a        (Q1 emitter to Q2 c
Ground:             jumper column 16 row a -> top - rail            (Q2 emitter)
Collector resistor: Rc (1 k): column 12 row a -> column 15 row a;   jumper column 15 row
Base resistors:     Ra (10 k): column 11 row b -> column 5 row b;    input A = the free
                    Rb (10 k): column 17 row b -> column 22 row b;   input B = the free
                    (plug the other end of each input jumper into the + rail for 1 or th
Measure:   OUT = red probe column 12 row e, black on the - rail;   Q1 base = column 11 r
Connect the power LAST.

Compare with a real gate (74HC00, DIP-14 in row e columns 30 to 36, row f 36 to 30, notc
   power:  pin 14 -> + rail,  pin 7 -> - rail,  100 nF across the rails next to it;
   gate 1: A = pin 1, B = pin 2, output = pin 3.   Tie the unused gates' inputs (pins 4,
   Wire pin 3 to a 1 k + LED to ground to see the output:  LED current = (about 4.9 V -
```

1. Set A = 0, B = 0 (both jumpers on the − rail): OUT reads 5.0 V. Then A = 0, B = 1 and A = 1, B = 0: OUT still 5.0 V.
2. Set A = 1, B = 1: OUT drops to about 0.2 V; the voltage across R_c is about 4.8 V (4.8 mA); Q1's base reads about 0.8 V and Q2's about 0.7 V.
3. Leave input A unconnected (floating) with B = 1: OUT wanders: a floating input is neither HIGH nor LOW. Touch the loose wire with a finger and watch OUT follow your hum.
4. Wire the 74HC00 as above and repeat the four combinations. OUT is 5 V or 0 V (the 74HC output goes almost to the rails; the transistor version can only reach 0.2 V).
5. *Threshold test:* hold B = 1, connect input A of the 74HC00 to the wiper of a 10 kΩ pot across the supply and turn it while watching the LED. The output flips at about 2.5 V (half the supply). Do the same with the transistor gate: it flips at about 0.65 V (one base-emitter drop): a much worse noise margin.

Build 2: a 555 blinker clocking a flip-flop

The TLC555 oscillates at about 1 Hz; its output drives LED1 and also the clock of one 74HC74 flip-flop wired as a toggle, whose Q output drives LED2 at half the rate.

```
 +5 V o------+---------------+---- U1 pin 8, pin 4;   U2 pin 14, pin 1 (1RD-bar), pin 4
             |
         [ R1 1 k ]                      U1 = TLC555:      pin 1 GND,  pin 2 TRIG,  pin
             |                                             pin 5 CTRL (10 nF to GND),  p
             +--------- pin 7
             |
         [ R2 68 k ]
             |
             +--------- pin 6  and  pin 2 (joined)
             |
         [ C 10 uF ]  (+ at the top)
             |
 0 V o-------+

 U1 pin 3 (OUT) ----+---- [ R3 2.2 k ] ---- LED1 ---- 0 V            t_high = 0.478 s, t
                    |
                    +---- U2 pin 3 (1CP, the clock)

 U2 = 74HC74, flip-flop 1:   pin 2 (1D) wired to pin 6 (1Q-bar)          (toggle: D = th
                             pin 5 (1Q) ---- [ R4 1 k ] ---- LED2 ---- 0 V
                             pins 1 and 4 (RD-bar, SD-bar): to +5 V       (inactive)
 Unused flip-flop 2:         pin 13 (2RD-bar) and pin 10 (2SD-bar) to +5 V;   pin 12 (2D
 Decoupling:                 100 nF across the supply at U1 and at U2

 Frequency check:  T = 0.693 x (R1 + 2 R2) x C = 0.693 x 137 k x 10 uF = 0.949 s      f(
```

```
Breadboard layout. Top rails: + rail = 5 V, - rail = 0 V. C1 (100 nF) across the top rail
U1 (TLC555) straddles the centre gap, notch towards column 1: pins 1 to 4 in row e, colu
   U1 pin 1 = column 10 row e     pin 2 = column 11 row e     pin 3 = column 12 row e
   U1 pin 8 = column 10 row f     pin 7 = column 11 row f     pin 6 = column 12 row f
Power:        jumper column 10 row g -> top + rail          (pin 8, VCC)
              jumper column 10 row a -> top - rail          (pin 1, GND)
              jumper column 13 row a -> top + rail          (pin 4, RESET-bar held HIGH)
Control:      C3 (10 nF): column 13 row g -> column 15 row g;   jumper column 15 row h -
Join 2 and 6: jumper column 11 row b -> column 12 row h          (pin 2 to pin 6: the ti
R1 (1 k):     column 11 row g -> column 17 row g;   jumper column 17 row h -> top + rail
R2 (68 k):    column 11 row i -> column 18 row i;   jumper column 18 row j -> column 12
C (10 uF):    + leg column 11 row d, - leg column 14 row d;   jumper column 14 row b -> t
LED1:         R3 (2.2 k): column 12 row b -> column 22 row b;   LED1 anode column 22 row
              jumper column 23 row d -> top - rail

U2 (74HC74): pins 1 to 7 in row e columns 30 to 36, pins 14 to 8 in row f directly oppos
```

```
    pin 1 = column 30 e   pin 2 = column 31 e   pin 3 = column 32 e   pin 4 = column 33 e
    pin 14 = column 30 f  pin 13 = column 31 f  pin 12 = column 32 f  pin 11 = column 33
  Power:        jumper column 30 row g -> top + rail (pin 14);   jumper column 36 row a ->
  Inactive set/reset of flip-flop 1:   jumper column 30 row a -> top + rail (pin 1);   jump
  Toggle:       jumper column 31 row b -> column 35 row b          (pin 2, 1D, to pin 6,
  Clock:        jumper column 12 row c -> column 32 row c          (U1 pin 3 to U2 pin 3,
  LED2:         R4 (1 k): column 34 row d -> column 38 row d;   LED2 anode column 38 row b
                jumper column 39 row c -> top - rail
  Unused flip-flop 2:   jumper column 31 row g -> top + rail (pin 13);   jumper column 34
                        jumper column 32 row g -> top - rail (pin 12);   jumper column 33
  Connect the power LAST.
  Measure:   OUT = red probe column 12 row a;   timing node = column 11 row c;   Q = colum
             (if your board has bottom power rails, use them for the lower-half jumpers an
```

1. Power on. LED1 flashes about once a second, about equally on and off. LED2 flashes at half the rate, and changes state only at the moment LED1 *turns on* (the rising edge).
2. Count 20 flashes of LED1 with a stopwatch: about 19 s for the calculated 0.949 s each (anywhere from 15 to 25 s is normal with a 20 % capacitor).
3. Meter on the timing node (pin 2/6): about 2.5 V steady on a digital meter, the average of a signal swinging between 1.67 V and 3.33 V. Meter on OUT: about 2.5 V average (a rapidly switching signal reads its average).
4. Change C to 1 µF: the blinker runs ten times faster (10.5 Hz, too fast to count but a visible flicker); LED2 at 5.3 Hz.
5. Disconnect pin 2 (1D) from pin 6 and tie it to the + rail: LED2 stays on after the first clock edge, because D is always 1: a flip-flop remembers D at the clock edge.

flip-flop is not connected to anything, but the supply current of the chip can rise and the chip warm slightly; floating inputs do harm even with no visible effect.

Build 3: PWM motor-speed control with the 555

Pulse-width modulation switches a motor fully on and fully off, thousands of times a second; the fraction of the time that it is on (the **duty cycle** D) sets the average voltage, and so the speed. The 555's duty cannot go below 50 % with the plain circuit, so two diodes *steer* the capacitor's charge and discharge currents through different parts of the potentiometer: moving the wiper trades charge time against discharge time while the total stays almost constant, so **the frequency stays near-constant and only the duty changes**.

```
+5 V o-----+-------------------------------------------------+---- U1 pin 8 and pin 4 (re
           |
       [ R1 1 k ]
           |
           +--------------------- U1 pin 7 (DISCH): call this node M
           |
        D1 | anode at M, cathode towards the pot's end A            D2:  anode on the
           |                                                        |
           +--------[ A    P1 100 k    B ]--------[ Rd 1 k ]--------+---- joins M again
                              | W (wiper)
                              +------------------ U1 pin 6 and pin 2 (joined): the timin
                              |
                         [ C 4.7 nF ]
                              |
0 V o-------------------------+

Charging path (OUT HIGH):  +5 V -> R1 -> D1 -> A end of P1 -> wiper -> K (capacitor).
Discharge path (OUT LOW):  K -> wiper -> B end of P1 -> Rd -> D2 -> M -> pin 7 (pulled

With the wiper a fraction p of the way from A to B  (R_AW = p x 100 k):
  charge resistance      Ra = 1 k + p x 100 k        discharge resistance   Rb = 1 k +
  Ideal (no diodes):  t_high = 0.693 Ra C,  t_low = 0.693 Rb C,  T = 0.693 x 102 k x 4.

The diodes' 0.5 V drop changes the maths: the capacitor charges toward Vcc - 0.5 V and
  t_high = 0.887 Ra C       t_low = 0.571 Rb C         (from  ln((5-0.5-1.667)/(5-0.5-3
so the real duty is somewhat above the ideal:  D = 0.887 Ra / (0.887 Ra + 0.571 Rb),  a

   p      Ra       Rb       ideal D    ideal f     real D (diode-corrected)    real f
  0.25    26 k     76 k      25.5 %     3.01 kHz         about 35 %            about 3.
  0.50    51 k     51 k      50.0 %     3.01 kHz         about 61 %            about 2.
  0.75    76 k     26 k      74.5 %     3.01 kHz         about 82 %            about 2.
Duty goes from about 1 % to about 99 % over the pot's travel. The exact value depends o
so treat the real column as +/- 5 points. A 3 kHz PWM frequency is a little audible (a

Output stage (Basic Electronics 6): OUT (pin 3) -> Rg 100 ohm -> MOSFET gate;  Rpd 100
MOSFET source to 0 V, drain to the motor's lower terminal;  motor's upper terminal to t
Pack - to the circuit's 0 V (one common ground).   Average motor voltage = D x 6 V  (ig
Gate drive:  the average gate current is  Qg x f = 48 nC x 3 kHz = 0.14 mA   (Qg = 48 n
```

```
Breadboard layout (clear the board first). Top rails: + rail = 5 V, - rail = 0 V. The mo
U1 (TLC555): same positions as Build 2:   pin 1 = column 10 e,  pin 2 = column 11 e,  pi
                                          pin 8 = column 10 f,  pin 7 = column 11 f,  pi
```

```
Power:        jumper column 10 row g -> top + rail;   jumper column 10 row a -> top - ra
Control:      C3 (10 nF): column 13 row g -> column 15 row g;   jumper column 15 row h -
Timing node K (pins 2 and 6):   jumper column 11 row b -> column 12 row h
Capacitor:    C (4.7 nF): column 11 row c -> column 14 row c;   jumper column 14 row a -
Node M (pin 7, column 11 rows f to j):
   R1 (1 k):  column 11 row g -> column 17 row g;   jumper column 17 row h -> top + rail
   D1:        anode column 11 row h, cathode (banded end) column 16 row h;   jumper colu
   D2:        anode column 30 row h, cathode (banded end) column 34 row h;   jumper colu
Potentiometer P1: legs in row j: end A = column 24 row j,  wiper = column 25 row j,  end
   (A is the end joined to D1;  B the end joined to Rd)
   wiper to K:  jumper column 25 row i -> column 11 row d
   Rd (1 k):    column 26 row i -> column 30 row i
Output stage (MOSFET Q3 legs in row c: gate column 40, drain column 41, source column 42
   Rg (100 ohm): column 12 row b -> column 16 row b;   jumper column 16 row c -> column
   Rpd (100 k):  column 40 row d -> column 44 row d;   jumper column 44 row e -> top - r
   Source:       jumper column 42 row a -> top - rail
   Motor:        one wire in column 41 row a (drain), the other in column 46 row a;   ju
   D3 (1N4001):  anode column 41 row e, cathode (banded end) column 46 row e
   Battery pack -: black lead to the top - rail.
LED1 (brightness follows duty):  R3 (2.2 k): column 12 row d -> column 22 row d;   LED a
Connect the 5 V LAST, then the motor pack.
Measure:   pin 3 average voltage = red probe column 12 row a;   drain voltage = red prob
           (black probe on the - rail for all)
```

1. Turn P1 fully towards end A (p = 0): LED1 almost off, the motor is stopped. Pin 3 average about 0.05 to 0.1 V (D about 1 to 2 %).
2. Turn it a quarter of the way (p = 0.25): LED1 dim; pin 3 reads about 1.7 V (D about 35 %); the motor is about to start (hobby motors need about 2 V average to overcome friction; the average at this position is 6 V × 35 % = 2.1 V).
3. Half way (p = 0.5): pin 3 about 3.0 V (D about 61 %), the motor runs at medium speed; three quarters: about 4.0 V (D about 82 %); fully towards B: LED1 full and the motor at full speed (pin 3 about 4.9 V).
4. Speed in relation to duty: motor speed is close to proportional to the average voltage D × 6 V above the stall point (the motor acts like a low-pass filter on the pulses: its winding inductance of a few mH and its inertia average the 3 kHz).
5. Remove D3 for a moment (power off first): the drain terminal shows spikes above 6 V when the MOSFET switches off the inductive load (the flyback voltage of Basic Electronics 5). With D3 in place the drain never exceeds about 6.7 V.

6. Compare the frequency at each position with a frequency meter or the simulator: it changes from about 3.2 kHz to 2.6 kHz as p goes from 0.25 to 0.75, not exactly constant, as the table predicts.

Simulators

Falstad Circuit Simulator (falstad.com/circuit): gates, flip-flops, counters and the 555 are in the *Draw* menu under the *Logic Gates* and *Digital Chips* groups (and the *Circuits* menu has ready-made 555 astable and flip-flop examples, a good place to start). For Build 1, draw two NPN transistors from *Active Components*, add logic-level inputs via *Inputs and Sources* (a logic input element) and use a 5 V source; Falstad's transistors are ideal enough to give the 0.2 V low and 5 V high of the calculation. For Build 2, drop a 555 chip with R1, R2 and C; a D flip-flop with Q-bar joined to D divides by two, and the scope shows both traces. For Build 3, model the motor as a 40 Ω resistor in series with a 5 mH inductor (6 V / 0.15 A = 40 Ω), put D3 across both, and use a logic-level N-channel MOSFET (a plain MOSFET with a 5 V gate is enough); put a slider on the potentiometer resistance and watch the duty cycle on the scope.

Tinkercad Circuits (tinkercad.com): the component list has a 555 timer, a 74HC595 shift register, LEDs, push buttons, a DC motor, a power supply, a multimeter and an oscilloscope, and an Arduino Uno. Tinkercad's list of logic ICs has changed over time, so search it for the 74-series chips (hex inverter, quad NAND and flip-flop); if one is missing, build the same function from the transistors or use Falstad for that chip. Its 555 follows the same formulas; set R_1, R_2 and C and measure the output with the oscilloscope, which gives the period and duty directly. A 74HC595 with an Arduino and the `shiftOut()` function is a good simulation of the three-pin expansion of section 8.

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
1: transistor NAND, OUT for A,B = 00, 01, 10	5.0 V	4.98 to 5.00 V (no load)	5.0 V
1: OUT for A = B = 1	0.2 V	0.10 to 0.30 V	0.1 to 0.2 V
1: voltage across R_c when A = B = 1	4.8 V (4.8 mA)	4.7 to 4.9 V	4.8 V
1: Q1 base / Q2 base when A = B = 1	0.8 V / 0.7 V	0.75 to 0.85 V / 0.65 to 0.75 V	0.8 V / 0.7 V

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
1: 74HC00 OUT high / low	5 V / 0 V	4.9 V (with the 1 k + LED) / 0.00 V	5 V / 0 V
1: 74HC00 switching threshold	about 2.5 V	2.2 to 2.8 V (guaranteed outside 1.5 to 3.5 V)	about 2.5 V
1: transistor gate switching threshold	about 0.65 V	0.55 to 0.75 V	0.65 V
2: 555 f / t_{high} / t_{low}	1.053 Hz / 0.478 s / 0.471 s	0.85 to 1.3 Hz (C is ±20 %)	1.053 Hz (exact values entered)
2: duty of OUT	50.4 %	49 to 52 %	50.4 %
2: timing node (pin 2/6) average / swing	2.5 V / 1.67 to 3.33 V	2.4 to 2.6 V	swing 1.67 to 3.33 V on the scope
2: OUT high / low	5 V / 0 V	4.5 to 5.0 V (with LED) / 0.05 to 0.15 V	5 V / 0 V
2: LED1 current / LED2 current	1.2 mA / 2.9 mA	1.0 to 1.4 mA / 2.6 to 3.1 mA	1.2 mA / 2.9 mA
2: LED2 (Q) f and duty	0.527 Hz, 50 %	half the LED1 rate; 50 % to within 1 %	0.527 Hz
3: ideal frequency	3.01 kHz	2.6 to 3.3 kHz, falling as p rises	3.01 kHz (ideal diodes)
3: duty at p = 0.25 / 0.5 / 0.75	25.5 / 50.0 / 74.5 % (ideal); 35 / 61 / 82 % (with 0.5 V diode drops)	35 / 61 / 82 % (± 5 points)	ideal diodes give 25.5 / 50 / 74.5 %; add a 0.5 V drop to match
3: pin 3 average voltage at p = 0.25 / 0.5 / 0.75	1.7 / 3.0 / 4.0 V (real duty x 4.9 V)	1.6 to 1.9 / 2.8 to 3.2 / 3.8 to 4.3 V	(from the duty)
3: motor average voltage at the same positions	2.1 / 3.7 / 4.9 V (6 V x real duty)	within about 0.5 V (the flyback current adds a little)	(from the duty)
3: gate average current	0.14 mA (Q_g f)	not measured directly	—

Real readings spread because the capacitor in Build 2 is a ±20 % part, diodes differ, the 555 has a timing error of a few percent and an LED's voltage varies between colours and batches. What does not vary is the structure: the transistor NAND is LOW only when both inputs are HIGH; the 74HC switches near half the supply and the transistor gate near 0.65 V; LED2 changes only on the rising edge of LED1 and runs at exactly half its rate; and in Build 3 the duty (and so the speed) rises as the wiper moves away from the D1 end while the frequency stays within about 20 % of 3 kHz.

13. Challenges

CHALLENGE 1: LEVELS, MARGINS AND PULL-UPS

(a) Using the table in section 1, calculate NM_L and NM_H for a 74HC output driving a 74HC input at 3.3 V, and for the same chip at 5 V driving a 74LS input. Then decide whether each of these connections works: an ESP32 pin driving a 74HC at 5 V; an ESP32 pin driving a 74HCT at 5 V (give the noise margins); a 74LS output driving a 74HC input at 5 V. (b) A 5 V Arduino output must feed a 3.3 V Pico input. Use R_1 = 1 kΩ and choose R_2 from the E12 series so that the output is as close as possible to 3.3 V without exceeding it; give the output voltage and the 10 % to 90 % rise time with 15 pF of load, and with a 100 pF wire instead. (c) Four open-drain sensors share one wire with a pull-up to 5 V. Each can sink 3 mA at 0.4 V. The wire capacitance is 200 pF and it must rise (30 % to 70 %) in 1 µs. Find the minimum and maximum pull-up, choose a standard value and give the rise time and the current when the wire is pulled low. A fifth, push-pull device is mistakenly added, with an output resistance of 65 Ω in each state: while it drives HIGH, a sensor holds the wire LOW with an on-resistance of about 130 Ω (0.4 V / 3 mA). Find the current and the voltage on the wire, and say whether that voltage is a valid LOW.

View solution

CHALLENGE 2: GATES, LATCHES AND SHIFT REGISTERS

(a) Build an XOR from four NAND gates and check all four input combinations of A and B by tracing each gate; state how many 74HC00 chips you need and what you must do with the unused pins. (b) For the NAND SR latch of section 7, trace the outputs Q and Q-bar through this input sequence (S-bar, R-bar): (1,1), (0,1), (1,1), (1,0), (1,1), (0,0), (1,1). Start with Q = 0. Mark the step that is forbidden and say what is wrong with the last step. (c) A robot needs 22 LED outputs and 12 bumper inputs from a board with only 6 spare pins. Choose the number of 74HC595 and 74HC165 chips, the pins used, the time to update all LEDs and to read all inputs at a 1 MHz clock, and check the 70 mA supply limit if every LED needs 6 mA (and say how many LEDs may be on at once if each needs 10 mA).

View solution

CHALLENGE 3: THE 555 AND PWM

(a) Design a 555 astable at 1 kHz with about 60 % duty using C = 10 nF and E12 resistors: find R_1 and R_2 (use $R_2 = 2\,R_1$ as a start), and the frequency and duty you actually get. Then find the frequency and times if the capacitor is 20 % too large. (b) Design a monostable that holds a beeper on for 2 s after a trigger, with C = 10 µF; find R from the E12 series and the pulse length, and the range for a ±20 % capacitor. (c) For Build 3 (R1 = Rd = 1 kΩ, P1 = 100 kΩ, C = 4.7 nF), find the ideal duty and frequency at p = 0.2, then the diode-corrected duty and frequency using $k_h = 0.887$ and $k_l = 0.571$, and the average motor voltage on a 6 V pack for each. A motor needs an average of 2.0 V to start: show that it will not turn at p = 0.2 but will at p = 0.3 (find the corrected duty and average voltage at p = 0.3 too), and say what the ideal formula would wrongly have predicted for p = 0.3.

View solution

QUICK REFERENCE

- **Levels:** output V_{OH} must be at least the input's V_{IH}, V_{OL} at most V_{IL}; $NM_L = V_{IL} - V_{OL}$, $NM_H = V_{OH} - V_{IH}$, both must be positive. 74HC at 5 V: 1.5 / 3.5 V in, 0.1 / 4.4 V out; 74HCT in: 0.8 / 2.0 V
- **Gates:** AND, OR, NOT, NAND, NOR, XOR; NAND alone makes all of them (NOT = NAND(A,A); OR = NAND of the inverted inputs); bubble or bar = active low or inverted
- **CMOS:** complementary P and N MOSFETs, almost no current at rest; $P = C_{pd}\,V^2\,f$; propagation delay about 8 ns at 5 V, longer at 3.3 V
- **74HC rules:** tie every unused input to VCC or GND; 100 nF at every chip; never tie two push-pull outputs together; supply 2 to 6 V; fan-out is limited by speed, not DC current
- **Pull-ups:** 10 kΩ typical for a button (0.5 mA pressed at 5 V); open-drain wired-AND: R_p between $(V_{CC} - V_{OL})/I_{OL}$ and $t_r/(0.8473\ C)$
- **Schmitt trigger (74HC14):** V_{T+} about 2.4 V, V_{T-} about 1.4 V at 5 V; RC debounce delay = $-RC\ \ln(1 - V_{T+}/V_{CC}) = 0.65\ RC$
- **Memory:** NAND SR latch (inputs active low, 00 forbidden); D flip-flop copies D on the rising clock edge (74HC74; set/reset tied high); Q-bar to D divides by 2
- **Counters and shift registers:** 74HC161 counts to 15, $f/2^n$ on Q_n; 74HC595 (DS, SHCP, STCP: 3 pins, 8 outputs, 6 mA each, 70 mA total); 74HC165 (PL-bar, CP, Q7: 3 pins, 8 inputs); chains multiply the outputs, not the pins

- **555 astable:** $t_{high} = 0.693\ (R_1 + R_2)\ C$, $t_{low} = 0.693\ R_2\ C$, $f = 1.44 / ((R_1 + 2R_2)\ C)$, duty = $(R_1 + R_2) / (R_1 + 2R_2)$; **monostable:** $t = 1.1\ R\ C$; use a CMOS 555 for 3.3 V and 5 V logic
- **PWM:** average voltage = $D \times V_{supply}$; steering diodes keep f nearly constant while D changes; MOSFET gate current average = $Q_g\ f$
- **Level shifting:** 5 V to 3.3 V: divider $(R_2/(R_1+R_2) = 2/3)$, 74LVC, or MOSFET; 3.3 V to 5 V: 74HCT or 74AHCT at 5 V; I2C: BSS138 shifter with a pull-up on each side; Pico, ESP32 and Raspberry Pi pins are not 5 V tolerant

Next, Basic Electronics 9 moves from signals to power: DC motors, servos and steppers at circuit level, H-bridge and motor-driver chips, relays, and the batteries, regulators and current budgets that keep a robot from browning out. The PWM idea of Build 3 returns there in a full H-bridge.

Basic Electronics

Course 1 · Chapter 9 · Motors, Relays and Power

So far the loads have been LEDs, small chips and a motor that was simply switched on and off. A robot is different: its motors need amps, they turn the other way when you ask them to, they spike and stall, and they share a battery with a microcontroller that resets if the supply dips for a few milliseconds. This chapter treats the *actuators* (DC motors, hobby servos, stepper motors, relays) and the *power system* (batteries, regulators, capacitors, grounding, fuses, wire) as one subject, because in practice they cannot be separated: most "mysterious" robot faults are a motor and a microcontroller fighting over the same few volts. It uses the transistor switch and the flyback diode of Basic Electronics 5 and Basic Electronics 6, the inductor behaviour of Basic Electronics 4, the PWM of Basic Electronics 8, the divider and power formulas of Basic Electronics 1 and Basic Electronics 3, and the grounding ideas of Basic Electronics 7. The next chapter, Basic Electronics 10, drives all of this from code.

1. The DC motor as a circuit

A brushed DC motor is a coil of wire (the armature) turning inside a magnetic field. For the circuit designer it behaves as two things in series: a **resistor R** (the copper winding, typically 1 to 30 Ω in hobby motors) and a **voltage source E** that opposes the supply. That source is the **back-EMF**: any conductor moving through a magnetic field generates a voltage, and the faster the shaft turns, the larger it is. A small inductance L (about 1 to 5 mH) sits in series as well, and matters for PWM (section 2). One motor, three numbers:

```
Motor model (steady speed):

  +  o---[ R ]---( E )---o  -          V = I x R + E          E = Ke x w          T = K

  V   the voltage across the motor terminals (V)          I   the motor current (A)
  R   the winding resistance (ohm)                         w   the shaft speed (rad/s)
  Ke  the back-EMF constant (V per rad/s, i.e. V.s/rad)    Kt  the torque constant (N.m
  T   the torque the magnetic field produces (N.m)

  In SI units  Ke = Kt  (same number: V.s/rad = N.m/A).  Datasheets often give them in
  so convert first:   Ke [V per krpm] = Ke [V.s/rad] x 1000 x 2 pi / 60 = Ke x 104.72

  Three things follow from these equations alone:
```

```
1. Torque is set by CURRENT (T = Kt I), speed is set by VOLTAGE (w = (V - I R) / Ke
2. When the shaft is held still (E = 0), the current is limited only by the winding
3. Run with no load and the motor speeds up until the back-EMF almost cancels the s
   "no-load current" I0 (friction and wind losses).  No-load speed  w0 = (V - I0 R)
```

Worked example: one small hobby motor

Motor M1 (the same 6 V hobby motor as Basic Electronics 6) has, from its datasheet: nominal 6 V, winding resistance R = 5.0 Ω, no-load speed 8000 rpm and no-load current I_0 = 0.10 A. Everything else can be calculated.

```
No-load speed:   w0 = 8000 x 2 pi / 60 = 837.8 rad/s
Ke = (V - I0 R) / w0 = (6 - 0.10 x 5.0) / 837.8 = 5.5 / 837.8 = 0.006565 V.s/rad      =
Kt = Ke = 6.565 mN.m per A

Stall current:   I(stall) = V / R = 6 / 5.0 = 1.20 A        (6 V x 1.2 A = 7.2 W, all o
Stall torque:    T(stall) = Kt (I(stall) - I0) = 6.565 mN.m/A x (1.20 - 0.10) A = 7.22 m

Running at a load of 0.25 A:
  speed   w = (V - I R) / Ke = (6 - 0.25 x 5) / 0.006565 = 723.5 rad/s = 6909 rpm
  torque  T = Kt (I - I0) = 6.565 x (0.25 - 0.10) = 0.985 mN.m
  power   P(out) = T w = 0.000985 x 723.5 = 0.713 W          P(in) = V I = 6 x 0.25 = 1
  efficiency = 0.713 / 1.50 = 47.5 %      (loss in the winding I^2 R = 0.0625 x 5 = 0.3

Relations that save calculating every point:
  Speed falls LINEARLY with torque:       w = w0 x (1 - T / T(stall))          (the st
  Current rises linearly with torque:     I = I0 + T / Kt
  Maximum MECHANICAL power is at half the no-load speed and half the stall torque:
       I = (V + I0 R) / (2 R) = (6 + 0.5) / 10 = 0.65 A     speed 4000 rpm     P(out, m
       but its efficiency is only 6 x 0.65 = 3.9 W in, 1.51 W out = 38.8 %
  Maximum EFFICIENCY is at a much lighter load:   I = sqrt(I0 x I(stall)) = sqrt(0.10 x
  So a motor sized to give its job at a quarter of its stall torque runs cool, efficien
```

SIZE FOR THE STALL, RUN AT A FRACTION

Everything that carries motor current (the driver, the wires, the fuse, the battery) must survive the *stall current*, even if the motor normally draws a fifth of it, because a wheel against a wall, a jammed gearbox or simply the instant of switching on (the shaft is still, so E = 0) all draw V/R. For M1 that is 1.2 A against 0.25 A running: a driver rated “1 A continuous” would be marginal. The datasheet of the

motor gives the stall current; the driver must have either a rating above it or a current limit, or the program must stop the motor within a second or two of a stall.

Reading a motor datasheet

DATASHEET LINE	WHAT IT TELLS YOU	M1 (6 V HOBBY MOTOR)
Nominal (rated) voltage	The voltage at which the other figures are quoted; 50 to 150 % of it is usually safe for short use; the PWM average sets the real voltage	6 V
No-load speed and current	w_0 and I_0; gives K_e with R. A large I_0 means a poor motor (friction)	8000 rpm, 0.10 A
Stall torque and stall current	The worst case for the supply and the driver; the torque at which the motor gives up	7.2 mN·m, 1.2 A
Terminal resistance R	Winding resistance, measured with the shaft still; V/R = stall current. Check it with a meter on the 200 Ω range (it varies as you turn the shaft by hand, since brushes contact different coils: take the average)	5.0 Ω
Torque constant K_t (mN·m/A) or speed constant (rpm/V)	Torque per amp; the reciprocal of K_e in matching units: 1 / (0.6875 V per krpm) = 1455 rpm per V	6.57 mN·m/A, 1455 rpm/V
Terminal inductance L	Sets the electrical time constant L/R and the PWM ripple (section 2)	2 mH (τ = L/R = 0.4 ms)
Rated (continuous) torque and current	What the motor can do for hours without overheating; usually a quarter to a third of stall	about 0.25 A
Maximum efficiency and the current at which it occurs	Where to aim the gearing	50 % at 0.35 A

Gear motors: trading speed for torque

A hobby motor spins at thousands of rpm with a few mN·m of torque; a wheel needs the opposite. A gearbox of ratio N divides speed by N and multiplies torque by N, minus the friction of the gears (efficiency η_g, 0.6 to 0.9 depending on type: a worm drive is about 0.4 to 0.6, spur gears about 0.8 to 0.9 per stage pair). The motor sees only the output torque divided by N, so its

current for a given job falls with N, and the stall current of the motor is unchanged (the gearbox does not change V/R).

```
Gearbox ratio N, efficiency eg:     speed(out) = speed(motor) / N          torque(out) =

M1 with a 50:1 gearbox (eg = 0.70), motor at 0.25 A as above:
    speed  = 6909 / 50 = 138 rpm
    torque = 0.985 mN.m x 50 x 0.70 = 34.5 mN.m = 0.35 kg.cm            (a pull of 0.35
    stall torque at the output = 7.22 x 50 x 0.70 = 253 mN.m     (and the gears must su
    power out = 0.713 x 0.70 = 0.50 W      (same motor power, less gear loss: speed and
Choosing N for a wheel:   wheel speed (m/s) = (output rpm / 60) x pi x diameter.    138
```

2. PWM speed control and its frequency

Basic Electronics 8 built a PWM generator from a 555. Every microcontroller has hardware PWM on several pins. The motor is switched fully on and fully off at a fixed frequency f; the fraction of each period that it is on is the **duty cycle** D. Because the winding inductance keeps the current smooth, the motor responds to the *average* voltage:

```
Average voltage  V(avg) = D x V(supply)       so      speed (no load) = (D x V - I0 R) /

Why the current is smooth: the inductance L resists changes in current.  While the swit
path, section 3).  Peak-to-peak ripple (assumes the motor's back-EMF is steady and igno

       dI(p-p) = V x D x (1 - D) / (L x f)          largest at D = 0.5:     V / (4 L f)

M1 (L = 2 mH, R = 5 ohm, electrical time constant L/R = 0.4 ms) at 6 V, D = 0.5:
       f =    490 Hz  (default Arduino analogWrite on pins 5 and 6 is 980 Hz; on pins 3
       f =  1 000 Hz:   0.75 A p-p      f = 5 000 Hz:   0.15 A p-p      f = 20 000 Hz:
The motor's average current is only about 0.1 to 0.3 A, so 1.5 A of ripple means the cu
bad (heating, torque ripple, a whine at the PWM frequency).  By 20 kHz the ripple is a
```

Choosing the frequency

- **Above hearing.** The motor and its wires act as a loudspeaker for the PWM frequency; anything from about 1 kHz up to 15 kHz sings. 20 kHz or more is silent to most adults. This is the main reason motor drivers are driven at 20 to 30 kHz.

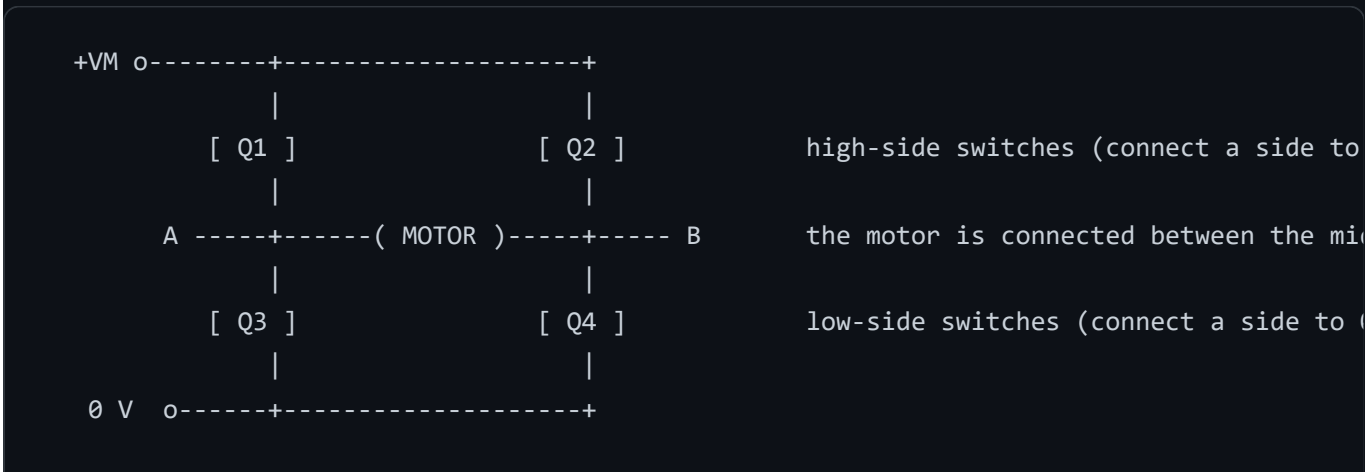
- **Well above 1 / (2 π τ)** where τ = L/R. For M1 τ = 0.4 ms (400 Hz): any f above about 5 kHz keeps ripple small.
- **Not too high.** Each edge costs switching loss in the driver (P_{sw} about ½ V I (t_{rise} + t_{fall}) f, an estimate), and the driver chip has a maximum PWM frequency (the L298N is slow: use at most about 5 to 10 kHz; the TB6612 handles 100 kHz). At 31 kHz with 100 ns edges the loss is tiny; at 31 kHz with a slow L298N it is not.
- **A coarser step at higher frequency.** A timer counting 8 bits (256 steps) at 16 MHz gives 16 MHz / 510 = 31.4 kHz (phase-correct mode). Higher frequency means fewer counts per period: at 100 kHz on a 16 MHz timer there are only 160 steps. 31 kHz with 256 levels is a good compromise on an Arduino Uno; a Raspberry Pi Pico and an ESP32 have 16-bit hardware PWM and give 20 kHz with thousands of levels.

Dead zone and a gentle start

A real motor does not move until the voltage overcomes stiction (static friction), typically 15 to 30 % of rated voltage: for M1, if starting needs 0.2 A, D × 6 V must exceed 0.2 × 5.5 Ω = 1.1 V (the 5.5 Ω includes the driver's 0.5 Ω), which is D = 18 %. Below that the motor hums and draws current without moving. A program maps 0 to 100 % of the command onto 18 to 100 % of the duty. Also, *ramp* the duty up over 50 to 300 ms rather than jumping from 0 to 100 %: at the instant of a jump the back-EMF is zero and the motor draws its full stall current (section 10 shows what that does to the supply).

3. The H-bridge: four switches, two directions

One transistor switches a motor on and off in one direction (Basic Electronics 6). To reverse a motor you must reverse the voltage across it. The **H-bridge** does this with four switches arranged as the letter H, the motor being the crossbar:



```
Q1 and Q3 are the left leg, Q2 and Q4 the right leg.  Each leg's top and bottom switche

Q1 Q2 Q3 Q4   Result
 1  0  0  1    FORWARD:  current +VM -> Q1 -> A -> motor -> B -> Q4 -> 0 V       (A is
 0  1  1  0    REVERSE:  current +VM -> Q2 -> B -> motor -> A -> Q3 -> 0 V       (B is
 0  0  1  1    BRAKE:    both motor ends joined to 0 V through Q3 and Q4; the spinning
                         stops it quickly (a "short brake"; Q1 and Q2 on instead gives
 0  0  0  0    COAST:    all four off, the motor spins down freely (current may flow th
 1  0  1  0    SHOOT-THROUGH: +VM -> Q1 -> Q3 -> 0 V, a short circuit.  Same for Q2 and
```

The bridge is just the four-way version of the switch of Basic Electronics 6: each of the four is a MOSFET with its gate driven by a signal. The control logic usually takes two inputs per motor (IN1 and IN2, which become the left-leg and right-leg commands) and one PWM input. The tables of the chips in section 4 look like this.

Shoot-through and dead time

A MOSFET does not turn off the instant its gate is pulled low: the gate capacitance (C_{iss}, Basic Electronics 6) has to discharge through the gate resistor. If Q1 is commanded off while Q3 is commanded on in the same instant, for a few hundred nanoseconds *both* are conducting and the supply is shorted through two tiny resistances. With $R_{DS(on)}$ of 25 mΩ each at 12 V that is 12 / (0.025 + 0.025) = 240 A (limited in practice by the supply, wires and the transistors' own heating, but every pulse stresses them, and the supply droops, upsetting the logic too). The cure is **dead time**: after switching one transistor off, wait a short time before turning its partner on.

```
Estimate of the turn-off time of a MOSFET driven through a gate resistor (illustrative;
    t(off) is about  Rg x Ciss x ln(Vdrive / Vth)

Example: Rg = 100 ohm, Ciss = 3.3 nF (IRLZ44N), Vdrive = 5 V, Vth = 2 V:   t(off) = 100
Dead time = about 2 x t(off) = 0.6 us   (a safety factor, because Ciss and the real tra
At a PWM frequency of 20 kHz (period 50 us) that costs 0.6 / 50 = 1.2 % of the duty ran
Integrated drivers generate the dead time themselves (section 4); with a hand-built bri

The conduction loss of a bridge: P = I^2 x (Rds(high) + Rds(low)).   Two 25 mohm MOSFET
A bridge made of diodes or bipolar transistors that drops 2 V at 3 A would waste 6 W: t
```

Freewheeling, body diodes and braking

A motor winding is an inductor; the current through it cannot stop instantly (Basic Electronics 4). When PWM switches one transistor off, the current must find another path. Every MOSFET has a **body diode** (a parasitic diode between source and drain, anode on the source for an N-channel part) which provides it automatically; this is the flyback protection of Basic Electronics 5 built in, so a bridge needs no external flyback diodes. Follow the current in the forward state with Q1 as the PWM switch and Q4 held on:

```
Forward, PWM on Q1, Q4 held ON (motor current flows from A to B through the motor):

  Q1 ON  (the "on" part of the cycle):   +VM -> Q1 -> A -> motor -> B -> Q4 -> 0 V.
  Q1 OFF (the "off" part):               The motor current cannot stop; it must still
                                         Its only source is Q3's body diode (anode at
                                         0 V -> Q3 body diode -> A -> motor -> B -> Q4
  Loss in the freewheel path = 0.7 V x I (the diode drop).  If Q3 is switched ON during
  with the dead time) the current flows through the channel instead and the loss is onl

Two ways to apply PWM, both used in drivers:
  Sign-magnitude:   one input sets the DIRECTION (held), the other is the PWM; the off
                    side: "slow decay / brake").  Smooth, lowest ripple, and the motor'
  Lock-antiphase:   the two legs are driven with opposite signals, 50 % duty = stopped,
                    every cycle (a "fast decay"); the ripple is much larger, but one pi
```

High-side drive

The low-side transistors are easy: gate to a logic signal, source to 0 V. The high-side transistors are harder. An N-channel MOSFET turns on when its gate is some volts *above its source*, and the source of Q1 sits at the motor voltage when Q1 is on. To keep Q1 on at 12 V the gate must go to about 22 V. Three solutions: a P-channel MOSFET (turns on with the gate *below* the source; simple, but its $R_{DS(on)}$ is two to three times higher than a comparable N-channel); an N-channel with a **bootstrap capacitor**, which a driver chip (IR2104, DRV8871) charges while the low side is on and then lifts the gate with (this is why such bridges cannot hold 100 % duty for ever); or a **charge pump** inside the driver (the DRV8833, TB6612 and most modern chips do this internally).

4. Integrated motor drivers

You will rarely build a bridge from four loose MOSFETs. Driver chips contain the bridge, the dead time, the high-side drive, the flyback paths and often thermal and current protection. Four

common families, with typical datasheet values (the real datasheet of the exact chip is the authority):

DRIVER	TECHNOLOGY	MOTOR SUPPLY	CURRENT PER BRIDGE	VOLTAGE LOST (HIGH + LOW) AT 1 A	HEAT IN THE CHIP AT 1 A	LOGIC AND PWM	NOTES
L298N (the red module)	Bipolar (BJT) transistors	about 7 to 46 V	2 A (two bridges)	about 2.2 V (about 1.4 V upper plus 0.9 V lower)	2.2 W per bridge: heatsink needed	5 V; PWM up to about 5 to 10 kHz (slow)	No internal diodes (the module adds slow 1N4007s). Wastes 2 V: a 6 V motor on a 6 V supply gets 3.8 V. Fine for 12 V motors at a modest current
L293D	Bipolar, with clamp diodes	4.5 to 36 V	0.6 A (1.2 A peak), two bridges	about 2.6 V at 0.6 A	1.5 W at 0.6 A	5 V logic; also the chip in the Tinkercad Circuits parts list	Small and simple; DIP-16 for breadboards; poor efficiency
TB6612FNG	MOSFET	2.5 to 13.5 V	1.2 A continuous (3.2 A peak, short), two bridges	about 0.5 V (R about 0.5 Ω high plus low)	0.5 W	2.7 to 5.5 V logic: works with Uno, Pico, ESP32; PWM up to 100 kHz; STBY pin	The usual choice for small robots; built-in dead time and diodes
DRV8833	MOSFET	2.7 to 10.8 V	1.5 A RMS (2 A peak),	about 0.36 V (R about	0.36 W	PWM straight on the IN	Ideal for 3.3 V logic and 3 to 9 V

DRIVER	TECHNOLOGY	MOTOR SUPPLY	CURRENT PER BRIDGE	VOLTAGE LOST (HIGH + LOW) AT 1 A	HEAT IN THE CHIP AT 1 A	LOGIC AND PWM	NOTES
			two bridges	0.36 Ω total)		pins; nSLEEP; accepts 3.3 V and 5 V inputs; current limit by a sense resistor	motors; very efficient

```
Same job, 6 V supply, a 6 V motor drawing 1 A (the table's drops are taken at 1 A):

                    motor volts    heat in the driver    share of the supply's energy
    L298N           6 - 2.2 = 3.8 V      2.2 W                 3.8 / 6    = 63 %
    TB6612FNG       6 - 0.5 = 5.5 V      0.5 W                 5.5 / 6    = 92 %
    DRV8833         6 - 0.36 = 5.64 V    0.36 W                5.64 / 6   = 94 %

The motor's speed follows the voltage it actually gets; the L298N motor also loses torqu
bipolar driver is a large voltage, not a small battery.  Rule of thumb: for a battery-p

TB6612FNG truth table for ONE bridge  (PWMA = the speed pin, STBY must be HIGH or the c
    AIN1  AIN2  PWMA  STBY    Out (AO1, AO2)
     H     L     H     H      forward  (AO1 high, AO2 low)              H, L, PWM:  fo
     L     H     H     H      reverse
     H     H    H or L H      SHORT BRAKE (both outputs low)
     L     L    H or L H      STOP (outputs off, the motor coasts)
     x     x     x     L      STANDBY (everything off, minimal supply current)
```

5. Servos

A hobby servo is a small gear motor, a position sensor (a potentiometer on the output shaft) and a control circuit in one case. The circuit compares the commanded position with the potentiometer and drives the motor in whichever direction closes the gap: a complete feedback

loop with no code on your side. It has three wires: **ground** (brown or black), **supply** (red, 4.8 to 6 V for a standard servo) and **signal** (orange or yellow). The command is a pulse, repeated every 20 ms (50 Hz): the **width** of the pulse sets the angle.

```
Signal:   a pulse every 20 ms (50 Hz); the pulse is HIGH for 1.0 to 2.0 ms (many servos

  pulse width     angle (for 1.0 ms = 0 deg, 2.0 ms = 180 deg)        duty = pulse / 20
     1.00 ms          0 deg                                          5.0 %
     1.25 ms         45 deg                                          6.25 %
     1.50 ms         90 deg  (centre)                                7.5 %
     1.75 ms        135 deg                                          8.75 %
     2.00 ms        180 deg                                          10 %

  angle = (pulse - 1.0 ms) / 1.0 ms x 180 deg      pulse = 1.0 ms + angle / 180 deg x 1
  Resolution:  1 us of pulse width = 0.18 deg.   A 16-bit hardware timer at 50 Hz (abou
  A continuous-rotation servo uses the same pulse: 1.5 ms = stop, 1.0 ms = full speed o
```

Current, spikes and a separate supply

A servo draws almost nothing while holding still (about 10 mA for a 9 g micro servo such as the SG90), a few hundred milliamps while moving, and a large stall current (about 650 mA for a micro servo, 2.5 A for a large metal-gear servo such as the MG996R) when the arm meets an obstacle or the command asks for an impossible position. Three servos moving together at 0.25 A each draw 0.75 A, and the same three stalling draw about 1.95 A. The motor is switched hard by a bridge inside the servo, so the current changes quickly, and the supply wires carry noisy spikes. Consequences:

- **Do not power servos from the 5 V pin of an Arduino** (it is limited by the USB port to about 500 mA total and shares the voltage with the processor) or from a board's 3.3 V pin. Use a separate 5 V to 6 V supply (four NiMH cells give 4.8 V; a 5 V buck converter, section 9, works too).
- **Join the grounds** of the servo supply and the board; without a common ground the signal has no reference. This is the only connection between the two supplies.
- **Put 470 µF to 1000 µF across the servo supply** next to the servos, to supply the spikes locally.
- A 3.3 V signal (Pico, ESP32) usually works with a 5 V servo (the servo's input threshold is typically about 2 V), but check it; put a 220 Ω resistor in the signal line to protect the pin.

- Set the position *once* and stop the pulses (detach) if the servo need not hold against a load: a holding servo draws current and buzzes. A servo holding against the load gets warm.

```
 Servo supply (separate)                                    Board (Uno, Pico or ESP

   +5 V o--------+-------------------------------------> servo red (supply)
                 |
              [ C 470 uF ]     (the + leg to the supply +, the stripe/- leg below)
                 |
   0 V  o--------+-------------------------------------> servo brown or black (ground)
                 |
                 +-------------------------------------> board GND         the ONLY link

   board PWM pin o-------[ 220 ohm ]---------------------> servo orange or yellow (signa
```

6. Stepper motors at circuit level

A stepper has several coils instead of one, and no brushes. Energise the coils in a repeating sequence and the rotor jumps one step at a time, so the position is set by *counting steps*, not by feedback (open loop). A typical motor has 200 steps per revolution (1.8° per step; a 0.9° motor has 400). Holding position costs full coil current even at rest, and a missed step (the load exceeded the torque at that speed) goes undetected.

	UNIPOLAR	BIPOLAR
Wires	5 or 6: each of two coils has a centre tap, which goes to the supply	4: the ends of two coils (A and A-bar, B and B-bar)
Driving	Each half-coil switched to ground by one transistor per half: a simple ULN2003 transistor array does it	Each coil needs its current reversed: an H-bridge per coil, which is what the A4988 and DRV8825 contain
Torque	Lower: only half the coil is used at a time	Higher: all of the copper is used
Example	28BYJ-48 (5 V, geared, commonly quoted at 2048 steps per revolution in full-step mode, slow and weak, with a ULN2003 board)	NEMA 17 (e.g. 1.5 A per phase, 200 steps per revolution) with an A4988 or DRV8825
Identifying the wires	With the meter on ohms and the motor disconnected: two wires belonging to the same coil read a low resistance (a few ohms), wires of different coils read open. A unipolar centre tap	

	UNIPOLAR	BIPOLAR
	reads half the resistance of the full coil. Never guess the pairing and connect: reversed coil ends make the motor shudder instead of turn.	

Why a stepper wants a higher supply: the coil's L/R time

The rated current of a NEMA 17 motor (1.5 A) at its coil resistance (1.8 Ω) is only 2.7 V, so it looks as if a 3 V supply suffices. But each coil is an inductor (say 3.2 mH) and, when the controller switches the current on (or reverses it), the current rises as in Basic Electronics 4: $\tau = L/R = 3.2\ \text{mH} / 1.8\ \Omega = 1.78\ \text{ms}$. At each step the current must change direction or magnitude in a fraction of the step time, and at 2.7 V it would need about five time constants (9 ms): the motor could only manage about 100 steps per second. With a 12 V supply the same current is reached in $-\tau \ln(1 - 2.7 / 12) = 0.45\ \text{ms}$, and a **chopper driver** holds the current at the limit by switching the 12 V rapidly on and off (a tiny buck regulator per coil). That is why a 2.7 V motor is run from 12 V to 24 V: the supply voltage sets how fast current can change (and so the top speed), the current limit sets the torque.

Driver chips and the current limit

The A4988 and DRV8825 modules have two bridges, a chopper current limit set by a trimmer, and two logic inputs for motion: **STEP** (one rising edge = one step) and **DIR** (the direction). Three **MS** (mode) pins select **microstepping**: instead of energising the coils fully on or off, the driver feeds them with a sine-and-cosine pair of currents so the rotor can stop between full-step positions. With 1/16 microstepping a 200-step motor has 3200 steps per revolution (0.1125° per step) and runs much more smoothly and quietly.

```
Current limit (the driver chops the coil current at this value): set it by measuring th

  A4988  :   I(max) = Vref / (8 x Rs)        Rs is the sense resistor on the board: 0.0
             Vref = I(max) x 8 x Rs          for 1.0 A:  Vref = 1.0 x 8 x 0.068 = 0.544
  DRV8825:   I(max) = Vref / (5 x Rs)        Rs = 0.100 ohm:         Vref = I(max) x 5 
  Always set the limit at or below the motor's rated current per phase.  Too high: the 

Microstep pins (A4988; the DRV8825 uses the same idea with MODE0 to MODE2 and reaches 1/
    MS1 MS2 MS3        step size       steps per rev (200-step motor)
     L   L   L         full              200
     H   L   L         1/2               400
     L   H   L         1/4               800
     H   H   L         1/8              1600
     H   H   H         1/16             3200
```

```
Step frequency = (rpm / 60) x steps per revolution:   300 rpm at 1/16:   5 rev/s x 3200
hardware or an Uno's timer can, a slow loop in a program cannot).
Microstepping trade-off: the torque that pulls the rotor to the next micro-position is
So fine microsteps give smoothness, not strength or accuracy; the load-bearing steps ar
```

```
A4988 wiring (names on the module; the DRV8825 module has the same pin names except MOD

  Motor supply 8 to 35 V o------+--------------> VMOT          Logic 3.3 V or 5 V (the
                                |
                        [ C 100 uF, 35 V ]    (+ leg to VMOT; as close to the module a
                                |
  Motor supply 0 V      o-------+--------------> GND (next to VMOT)       board GND o--
  board pin o-----> STEP        board pin o-----> DIR        ENABLE: LOW = on (leave un
  MS1, MS2, MS3: leave unconnected for full step (internal pull-downs), or link to VDD
  Coil 1: 1A and 1B       Coil 2: 2A and 2B  (the two ends of each coil: one coil's two

NEVER connect or disconnect a motor while the supply is on: the coil's current collapse
Without the 100 uF capacitor, the first switching spike can also destroy the module.
```

7. Relays and solid-state relays

A **relay** is an electromagnet pulling a metal contact: a small current in a coil switches a separate, electrically isolated circuit. It is the oldest way to switch a load that is too big, too high-voltage or of the wrong kind (mains, AC, or a signal that must be isolated) for a transistor. The contacts are **COM** (common), **NO** (normally open, closed when the coil is powered) and **NC** (normally closed, open when powered).

```
Driving a relay coil from a logic pin (the low-side switch of Basic Electronics 6, here

  +V (coil supply) o-------+-----------------+
                           |                 |
                     [ COIL ]          D1 (1N4001): cathode (the stripe)
                           |             to +V, anode to the collector / drain
                           +-----------------+----- collector (or drain) of the switchi
                                                    emitter (or source) to 0 V;   base
  contact side:  load supply o----[ load ]----o NO       COM o----> load supply return

Coil current   I = V(coil) / R(coil).      A 5 V relay with a 70 ohm coil:   5 / 70 = 7
```

```
                                                A 12 V relay with a 400 ohm coil:  12 / 400 =
The coil is an inductor:  the diode D1 is not optional (Basic Electronics 5): without i
Side effect: the diode holds the current in the coil, so the contacts release slower (a
Pull-in is typically at 75 % of the rated coil voltage and drop-out at 10 to 20 %:  a c
```

Contact ratings: AC and DC are different

A typical 5 V hobby relay is stamped *10 A 250 VAC / 10 A 30 VDC*. The two figures are not alike because a DC circuit has no zero crossing: when the contacts open, an inductive DC load (a motor, a solenoid) can sustain an arc that welds or erodes them. As a rule, derate to about half for DC inductive loads, add a flyback diode (or a snubber) *across the load*, and use a bigger contact than the arithmetic suggests. Contacts also **bounce** for 1 to 5 ms on closing (the debounce problem of Basic Electronics 8, now in the output), and wear out: a mechanical relay is rated for about 100 000 switchings at full load, so do not PWM a motor through one.

Solid-state relays (SSR)

A solid-state relay has no moving parts: an LED inside lights a photo-detector which turns on the output (a triac for AC, a pair of MOSFETs for DC). The input is a current (typically 3 to 32 V DC, 10 to 15 mA), electrically isolated from the output.

	MECHANICAL RELAY	SOLID-STATE RELAY (AC TRIAC TYPE)	SOLID-STATE RELAY (DC MOSFET TYPE) / PLAIN MOSFET
Drop across the switch	Contact resistance 30 to 100 mΩ: 0.05 Ω at 8 A = 3.2 W	About 1.2 V: 8 A × 1.2 V = 9.6 W (heatsink needed)	$R_{DS(on)}$ of a few mΩ: a fraction of a watt
Load supply	AC or DC, any voltage up to the rating	AC only (a triac cannot turn off DC)	DC only; polarity matters (the + and − marks)
Switching speed	5 to 15 ms; wears out	Turns on at the next zero crossing of the mains; unlimited life	Microseconds, PWM possible
Isolation	Full (coil to contacts)	Full (opto-isolated)	SSR: full. A plain MOSFET: none
Off state	Perfect open circuit	Leaks a few mA (a mains lamp may glow faintly)	Very small leakage

	MECHANICAL RELAY	SOLID-STATE RELAY (AC TRIAC TYPE)	SOLID-STATE RELAY (DC MOSFET TYPE) / PLAIN MOSFET
Noise	Click and an arc that radiates noise; needs a flyback diode and suppression	Silent	Silent
Use it for	Occasional switching of anything up to its rating; mains; dissimilar voltages; a main battery cut-off	Frequent switching of mains heaters and lamps	Robot loads: motors, solenoids, strips (a MOSFET); isolated DC switching (the SSR)

MAINS VOLTAGE

A relay or SSR is the usual way a microcontroller switches a mains appliance. The contacts and the mains wiring are lethal. Do not wire mains on a breadboard; use a ready-made, enclosed relay module with screw terminals and an earthed case, keep the low-voltage side and the mains side physically separate, never work on live wiring, and if you are not experienced with mains, do not build this yourself. Everything in this chapter's builds is battery low-voltage.

Relay modules (a relay, a transistor, a flyback diode, an LED and often an opto-isolator on a small board) are active LOW in many cases (the board relay turns on when the input is pulled to ground), have a 5 V coil, and a jumper that decides whether the opto-coupler is powered from the same 5 V (convenient, not isolated) or from a separate supply (isolated). A 3.3 V board (Pico, ESP32) driving an opto-isolated active-low module works with the module's own 5 V supply; check that the input current (about 5 to 15 mA) is within the pin limit.

8. Batteries

The power source decides how long the robot runs and whether it browns out. Four numbers define a battery: its **nominal voltage** (and the range it covers while discharging), its **capacity** (mAh or Ah: the charge it holds), its **maximum discharge current** (often expressed as a multiple of capacity, the **C-rate**) and its **internal resistance**, which makes the voltage sag under load.

CHEMISTRY	CELL VOLTAGE (NOMINAL; FULL TO EMPTY)	TYPICAL AA / CELL FIGURES	INTERNAL RESISTANCE	NOTES FOR ROBOTS
Alkaline (AA, 1.5 V)	1.5 V; 1.6 to 0.9 V	About 2500 mAh at low current, much less at 1 A	0.15 to 0.3 Ω per cell, rising as it empties	Cheap and everywhere, but sags badly under motor currents; not rechargeable
NiMH (AA, rechargeable)	1.2 V; 1.4 to 1.0 V	1900 to 2500 mAh; handles 1 to 2 A	0.02 to 0.05 Ω per cell	Safe, forgiving, 4 cells = 4.8 V (a servo supply); a good first robot battery
Li-ion (18650, 3.6 or 3.7 V)	3.6 or 3.7 V; 4.2 to 3.0 V	2000 to 3500 mAh; often 10 A or more for a good cell	20 to 60 mΩ	Large energy; needs a protection circuit and a proper charger
LiPo (pouch pack, 3.7 V per cell)	3.7 V; 4.2 to 3.0 V; 2S = 7.4 V (8.4 V to 6.0 V), 3S = 11.1 V (12.6 V to 9.0 V)	500 to 5000 mAh, 20C to 100C discharge	5 to 30 mΩ per cell	The lightest, strongest pack for drones and fast robots; the most dangerous (below)
LiFePO4 (3.2 V per cell)	3.2 V; 3.65 to 2.5 V	1500 to 6000 mAh	Low	Safer chemistry, flat discharge curve, 4 cells = 12.8 V; heavier than LiPo

```
Capacity and C-rate:       a cell of capacity Q (Ah) at "1C" delivers Q amps (dischargi
  2200 mAh pack:   1C = 2.2 A;   a 25C pack can deliver 25 x 2.2 = 55 A for short perio
  Needed C-rate = I(peak) / Q:   3 A from a 1.8 Ah pack = 3 / 1.8 = 1.67C: any decent c

Runtime:   t = (Q x usable fraction) / I(average)       usable fraction about 0.8 (do n
  2.2 Ah pack at 0.76 A average:  2.2 x 0.8 / 0.76 = 2.3 hours.   (Capacity is quoted a

Voltage sag under load:   V(terminal) = V(open circuit) - I x R(internal)       (this,
  a 2S LiPo (R about 60 mohm for the pack) at 10 A:   10 x 0.06 = 0.6 V lost;   four ha

LiPo / Li-ion state of charge, open circuit, per cell (rough guide only):  4.20 V full;
  Under load the voltage reads lower than at rest, so use the pack's rest voltage, or a
```

Lithium cells: protection and safety

- **Charge limits.** A lithium cell is charged at constant current up to 4.20 V, then at constant voltage until the current falls to about C/10: the job of a lithium charger (TP4056 for a single cell, a balance charger for LiPo packs). Charging past 4.2 V, or a cell below 3.0 V (or one that has been deeply discharged and then charged), can overheat and ignite it. Never charge with a bench supply set to "about right", and never charge a swollen or damaged pack.
- **Discharge limit.** Do not go below 3.0 V per cell (3.3 V while under load is a sensible cut-off). Use a **protection circuit module (PCM / BMS)** on single cells and packs, or a low-voltage alarm on LiPo packs: it cuts off at the over-discharge, over-charge and short-circuit limits. Bare, unprotected cells are for builders who add their own protection.
- **Short circuits.** A 25C 2200 mAh pack can push 55 A into a short: wires glow, insulation melts, the pack catches fire. Fuse the battery (section 12) close to the positive terminal, and never rest a loose lead where it may touch the other lead.
- **Physical damage.** A punctured, crushed or swollen lithium pack can ignite after a delay. Retire it, discharge it (a bulb or a resistor) and take it to a battery recycling point; do not put it in household bins.
- **Storage and charging.** Charge lithium packs on a non-flammable surface (a ceramic tile, a metal tray or a LiPo bag), not unattended, not on a bed or sofa. For storage over a week leave them at about 3.8 V per cell.

9. Regulators: linear against switching

A battery's voltage changes as it discharges and does not suit a 5 V or 3.3 V chip. A **voltage regulator** turns a variable, higher input into a fixed output. There are two very different kinds.

Linear regulators: simple, quiet, and wasteful

A linear regulator is a transistor acting as a variable resistor, continuously adjusted to keep the output at the target. The extra input voltage is burnt as heat in that transistor; the current in is the same as the current out (plus a little for the regulator itself, the **quiescent current**). It needs the input to exceed the output by at least the **dropout voltage**, or it can no longer regulate.

```
Power lost in a linear regulator:    P = (Vin - Vout) x I(out)  (+ Vin x I(quiescent))

  7805 (TO-220, 5 V, 1 A, dropout about 2 V, so Vin at least 7 V; theta about 65 C/W in
      9 V in, 0.5 A out:   P = (9 - 5) x 0.5 = 2.0 W,  rise 2.0 x 65 = 130 C   (without
      7.4 V in, 0.333 A out (Build B):  P = 2.4 x 0.333 + 7.4 x 0.005 = 0.84 W,  rise 0
```

```
AMS1117-3.3 (SOT-223, dropout about 1.1 V, Iq several mA, theta about 100 C/W on a sm
    5 V in, 0.5 A out:   P = (5 - 3.3) x 0.5 = 0.85 W, rise 85 C: too hot for a small
LDO (low dropout) parts, for example the MCP1700 or AP2112K (3.3 V, dropout 0.2 to 0.
    (a single Li-ion cell, 4.2 to 3.3 V, can feed a 3.3 V logic board through an LDO
Pick a linear regulator when:  the current is small (below about 100 to 200 mA), Vin
```

Switching regulators: efficient, noisy

A **buck converter** (step-down) switches the input on and off at 100 kHz to several MHz through an inductor and a diode (or synchronous MOSFET); the inductor smooths the pulses (Basic Electronics 4), and the output is the average, just as in PWM motor control. A **boost converter** steps up, a **buck-boost** or SEPIC can produce an output either above or below the input (for a single Li-ion cell feeding 3.3 V or 5 V).

```
Ideal buck:   Vout = D x Vin        (D = duty of the switch)       Ideal boost:   Vout
  7.4 V to 5 V:   D = 5 / 7.4 = 0.676                          3.7 V to 5 V (boost):
Inductor ripple (buck):   dI(p-p) = (Vin - Vout) x D / (L x f)       22 uH at 500 kHz,

Efficiency:  eta = Pout / Pin = 85 to 95 % for a good converter at 20 to 100 % load;
  5 V, 0.35 A from 7.4 V at 90 %:    Pin = 1.75 / 0.9 = 1.94 W,   Iin = 0.263 A,   loss

Noise: the switching edges put ripple (tens of mV) at the switching frequency on the ou
  follow the converter with an LC filter or a linear (LDO) regulator ("post-regulation"
```

	LINEAR (7805, AMS1117, LDO)	SWITCHING (BUCK, BOOST, BUCK-BOOST MODULE)
Efficiency	Vout / Vin: poor if the drop is large	85 to 95 %, nearly independent of the voltage drop
Heat	(Vin − Vout) × I; needs a heatsink above about 1 W	Small; the module stays cool
Noise	Very quiet, fast response, no external inductor	Ripple and spikes at the switching frequency; layout matters
Step up?	No	Boost and buck-boost types can
Parts	One chip and two capacitors	A ready-made module; or a chip with an inductor, diode and capacitors

	LINEAR (7805, AMS1117, LDO)	SWITCHING (BUCK, BOOST, BUCK-BOOST MODULE)
Typical robot use	3.3 V for sensors from a 5 V rail; small loads	5 V from a 7.4 V or 11.1 V pack for the board, servos and sensors; battery-powered builds

10. Current budgets and brown-outs

A **current budget** lists every load on each supply rail with its typical and its peak current, then checks the sum against what the battery, the regulator and the wires can deliver. It is a table, ten minutes of work, and it prevents most of the faults of a first robot. Always include the *peak* column: the stall current of every motor, the servo stall, the Wi-Fi bursts of an ESP32 (about 240 mA average when transmitting, with spikes to 500 mA), and the inrush of any capacitive load.

```
Budget for the small robot of Build B:  5 V rail (from a buck converter), mA
    Arduino Uno board (microcontroller, regulator, LEDs)
    TB6612FNG logic supply
    3 distance sensors (15 mA each)
    1 micro servo (moving: about 250 mA; stalled: about 650 mA)
    5 V rail total
  Motor rail (battery directly), mA   two M1 motors (0.25 A each running, 1.2 A each st
  Battery current (2S LiPo 7.4 V):   motors 0.50 A + buck input 5 V x 0.345 A / 0.9 / 7
                                     peak:  2.40 A + 5 V x 0.745 A / 0.9 / 7.4 V = 0.56
  Choose:  a buck converter rated 1 A or more (970 mA with margin); a fuse at 5 A (abov
  Runtime:  2.2 Ah x 0.8 / 0.76 A = 2.3 hours.
```

Brown-outs: when the supply dips

Microcontrollers watch their supply: if it falls below a threshold (a **brown-out detector**: about 2.7 V on the ATmega328P of the Uno at its usual setting, about 2.4 to 2.8 V on an ESP32), they reset rather than run on a voltage too low for the logic to work reliably. A robot whose motors share a battery with its processor can brown out each time a motor starts: the motor's stall current drags the battery voltage down across its internal resistance, and the processor sees the dip.

```
Scenario: an ESP32 robot, the board on a 3.3 V AMS1117 (dropout about 1.1 V) from the s
  Battery: four half-used alkaline AAs:   open circuit 5.2 V;  internal resistance 4 x (
  ESP32 with Wi-Fi: 0.24 A.
```

```
At the moment both motors start (E = 0): the two 5 ohm windings in parallel are 2.5 o
   V = 5.2 - 1.3 x (V / 2.5 + 0.24)     V x (1 + 0.52) = 5.2 - 0.312 = 4.888      V =
The regulator's output cannot exceed V - dropout = 3.22 - 1.1 = 2.1 V.  That is below
The motors then stop, the voltage recovers, the board boots, starts the motors again:

The cures, in order of effectiveness:
1. Separate supplies (or a separate regulator input): the logic on its own cell or it
2. A buck-boost or a boost regulator for the logic (it can keep 3.3 V while its input
3. Limit the start-up current (a soft start): ramp the PWM from 0 to 100 % over 200 m
   V = 5.2 - 1.3 x (0.4 + 0.24) = 4.37 V, the regulator gives about 3.27 V: no reset.
4. A bulk capacitor and a Schottky isolation diode: the diode (anode on the battery,
   supplies the logic for t = C x dV / I:   1000 uF, 1.0 V of allowed droop, 0.24 A:
   a spike, not a stalled motor that stays on for a second.  (A diode costs 0.3 V too
```

11. Bulk capacitors, decoupling and grounding

The capacitors of Basic Electronics 4 do three jobs on the motor side, at three different sizes.

- **Bulk capacitor** (100 µF to 1000 µF, electrolytic, rated at least twice the supply) across the supply of every motor driver and servo, as close as possible to it: a local reservoir that supplies the sudden current of switching edges and start-ups, so the wires (and the battery) need not. The ripple on a supply of capacitance C when the load draws a pulse of charge is dV = I × dt / C: a 1.2 A start-up pulse of 50 µs on 470 µF gives dV = 1.2 × 50e−6 / 470e−6 = 0.13 V.
- **Decoupling capacitor** (100 nF ceramic) across the supply pins of every chip, as in Basic Electronics 4 and Basic Electronics 8.
- **Motor noise capacitor.** A brushed motor sparks at its commutator, which radiates noise that disturbs the radio, the sensors and the microcontroller. Solder a 100 nF ceramic capacitor across the motor terminals, and (optionally) one from each terminal to the metal case (each 100 nF). The leads must be short, soldered on the motor itself.

Grounding: separate the high currents from the sensitive ones

Wires have resistance (section 12) and inductance, so a current through a ground wire creates a small voltage drop along it. If the motor's return current (an amp or more, pulsing at the PWM rate) shares a piece of wire with a sensor's ground, that drop appears *in series* with the sensor's signal as noise (the supply ripple and ground noise of Basic Electronics 7). The remedy is a **star**

ground: every return path goes back to one point, which is the battery's negative terminal (or one point on the ground rail), instead of being daisy-chained through each other.

```
  Star ground:  every load returns to the battery minus by its OWN wire (short and thick

                        +--------- motor driver GND (thick wire, carries all motor cur
                        |
    battery -  o--------*star----- regulator GND (the logic rail's return)
                        |
                        +--------- servo supply GND (a separate wire; its 470 uF capac
                        |
                        +--------- sensor / ADC reference GND (a thin wire, nothing el

    battery + o----[ FUSE ]----+----------> motor driver VM (with its bulk capacitor)
                               |
                               +----------> regulator input ---> logic rail (decoupled)

    Wrong way:  battery - ---> motor driver GND ---> (the same wire continues) ---> micro
    microcontroller's and the sensor's "0 V" by tens of millivolts at each PWM edge.  An
```

12. Protection and wiring: fuses, reverse polarity, wire gauge

Fuses and polyfuses

A fuse protects the *wire and the battery*, not the load: it must blow before the wire overheats and before a short-circuit current from a strong battery melts something. Choose it above the largest legitimate peak (the budget of section 10: 2.96 A, so 5 A) and below the rating of the thinnest wire it protects (22 AWG: about 7 A for chassis wiring). Use a fast-blow type for a lithium pack, close to the battery's positive terminal. A **polyfuse** (resettable PTC fuse) is a part whose resistance rises sharply when heated by an overcurrent: it has a *hold current* (it stays conducting up to this, say 1.1 A) and a *trip current* (about twice the hold current, say 2.2 A); once tripped it stays high-resistance until the current is removed and it cools. It is slow (seconds), has some resistance (0.1 to 0.3 Ω) and is good on a USB port or a sensor supply; it will not protect against the 55 A short of a LiPo.

Reverse-polarity protection

Connecting a battery the wrong way round destroys chips and electrolytic capacitors in an instant, and is easy to do with a bare two-wire connector. A **series diode** (Basic Electronics 5)

blocks reverse current but drops 0.3 to 0.7 V and burns $V_F \times I$: a 1N5819 Schottky at 3 A (0.45 V) wastes 1.35 W. A **P-channel MOSFET in the positive rail** (Basic Electronics 6) does the same with a tiny loss, as follows.

```
P-channel MOSFET reverse-polarity protection (in the positive supply line):

  battery +  o---------+ drain                        the MOSFET is placed so that its
                       |                              points TOWARDS the load: the rig
                   [ P-FET ]  gate ----+
                       |               |
                  source +-------------+--------> to the load (+)        gate to 0 V: t
                       |                                                (and, if the b
  battery -  o---------+---------------------------------> load (-)       from gate to

  Right way round:  the body diode conducts, the source rises to the battery voltage, V
  Reversed:         the load's ground rail is now joined to the battery's + terminal an
                    (drain to source) is reverse biased, the source stays near 0 V, Vgs
  Loss:  a P-channel part with Rds(on) = 30 mohm at 3 A:   V = 0.09 V,  P = 3^2 x 0.03
```

Wire gauge and voltage drop

Copper has resistance, so a long thin wire between the battery and the motor driver drops volts and wastes power. Choose the wire for both: heating (the current rating of the insulation) and the voltage drop over the length. The round-trip length (out and back) is what counts.

WIRE (AWG)	RESISTANCE (MΩ PER METRE)	SAFE CURRENT IN CHASSIS WIRING	DROP FOR 2 A OVER 1 M OUT AND 1 M BACK	TYPICAL USE
26 (breadboard jumper)	about 133, plus 50 to 100 mΩ per contact	Under 1 A	0.53 V and more (the contacts add)	Signals only
24	84.2	3.5 A	0.34 V	Sensors, small servos
22	53.0	7 A	0.21 V	Signal and light power wiring, small motors
20	33.3	11 A	0.13 V	Motors and battery leads up to a few amps

WIRE (AWG)	RESISTANCE (MΩ PER METRE)	SAFE CURRENT IN CHASSIS WIRING	DROP FOR 2 A OVER 1 M OUT AND 1 M BACK	TYPICAL USE
18	20.9	16 A	0.084 V	Main battery wiring
16	13.2	22 A	0.053 V	Heavy motors, fast robots

```
Drop  V = I x R(wire) x 2 x length;     heating  P = I^2 x R(wire) x 2 x length

Example:  1.5 m each way of 22 AWG carrying 3 A:    R = 0.053 x 3 m = 0.159 ohm      V =
          P = 9 x 0.159 = 1.43 W lost in the wire.     The same run in 18 AWG: 0.0209 x
For a short robot the heating limit rarely matters (the table's currents are for long ru
or screw terminals for motor current, crimped connectors (XT30, XT60, JST-XH) for batter
the contacts heat up, become unreliable and are a fire risk.  Twisting each pair (+ and
```

13. Which board for motors and power?

JOB	BEST BOARD	WHY
One or two DC motors with speed and direction (this chapter's Build A), 5 V driver parts, first project	Arduino Uno or Nano	5 V logic matches the TB6612 and L293D, hardware PWM on pins 3, 5, 6, 9, 10 and 11, `analogWrite()` in one line, the timer can be raised to 31 kHz, and Tinkercad Circuits simulates it; rugged pins
Four wheels, encoders, closed-loop speed, or 3.3 V parts (DRV8833)	Raspberry Pi Pico	Eight PWM slices (16 outputs) with 16-bit resolution, programmable I/O (PIO) blocks to count encoder pulses and produce step pulses precisely, 3.3 V logic that suits the DRV8833 directly
Many servos, Wi-Fi control from a phone, a small walker	ESP32	16 LEDC PWM channels for servos, wireless built in; but its Wi-Fi bursts and its brown-out threshold make a separate, solid supply essential (section 10)
A Linux robot with a camera and a set of servos	Raspberry Pi with a PCA9685 servo board or a motor HAT	Software PWM on a Raspberry Pi jitters (a visible servo twitch); a PCA9685 over I²C generates sixteen exact 50 Hz channels itself. Power the servos from their own supply, never the Pi's 5 V pin

JOB	BEST BOARD	WHY
Stepper axes	Pico (PIO), or any board plus A4988/DRV8825	A steady step pulse train at kilohertz rates needs hardware timing

Basic Electronics 10 writes the code for these cases and Basic Electronics 11 compares the boards in full.

14. Build it: a motor with PWM speed and direction, and a power budget for a small robot

Build A drives M1 through a TB6612FNG driver from an Arduino Uno: a potentiometer sets the speed, a push button reverses the direction. The Uno is the most suitable board here, for the reasons in section 13. **Build B** works out and then measures the power system of a small robot: a current budget, a comparison of a 7805 linear regulator and a buck converter feeding the same load, and a brown-out test in the simulator.

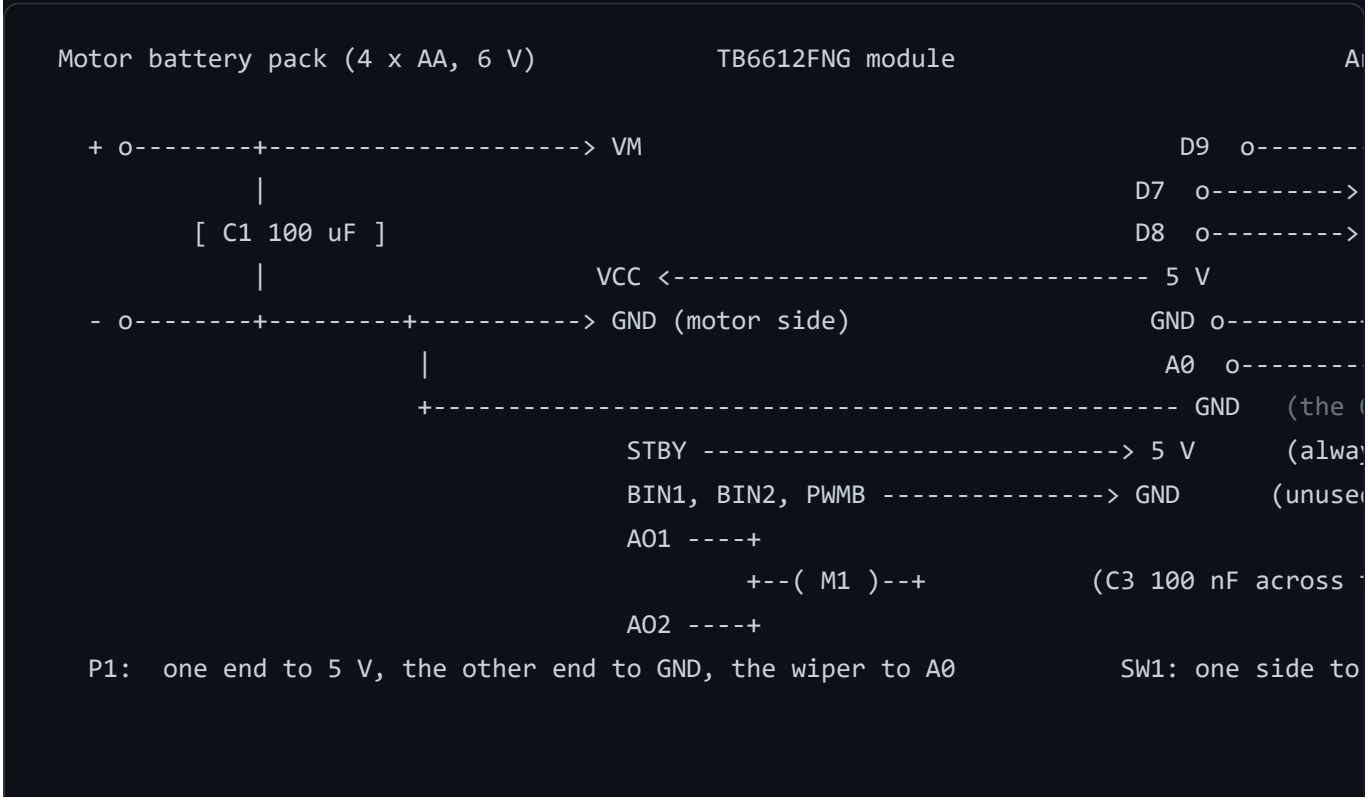
> **SAFETY AND ORDER OF CONNECTION**
>
> Use a separate battery pack for the motor (four AA cells, 6 V) and power the Uno from USB or its own supply; join only the grounds. Connect the motor pack last and disconnect it before any wiring change. Do not hold the motor shaft for more than a couple of seconds: stalled, it draws 1.1 A and the driver and the motor warm up. Use only alkaline or NiMH AA cells for Build A. Build B's bench supply should have its current limit set to 1 A; if you use a LiPo pack, fit a 5 A fuse and follow section 8. Never connect a lithium pack to anything you have not checked with the meter.

Parts list

PART	VALUE	USED IN
Board	Arduino Uno (or Nano) and a USB cable	A
U1	TB6612FNG breakout module (pins named VM, VCC, GND, STBY, AIN1, AIN2, PWMA, BIN1, BIN2, PWMB, AO1, AO2, BO1, BO2)	A
M1	Small 3 to 6 V hobby DC motor (about 0.1 A no-load, 1.2 A stalled) and a 4 × AA battery holder (6 V)	A
P1	10 kΩ linear potentiometer	A

PART	VALUE	USED IN
SW1	Push button (tactile switch)	A
C1	100 µF electrolytic, 16 V (motor supply bulk capacitor; + leg to the motor +)	A
C2, C3	100 nF ceramic: C2 from VCC to GND at the module, C3 across the motor terminals	A
U2	7805 regulator (TO-220), with 0.33 µF ceramic on the input and 0.1 µF on the output	B
U3	Adjustable buck converter module (LM2596 or MP1584 type), with input and output capacitors on the module	B
R_L	15 Ω, 5 W (or two 33 Ω 2 W in parallel, about 16.5 Ω) load resistor: 5 V / 15 Ω = 0.333 A, 1.67 W	B
Supply	7.4 V: a bench supply (current limit 1 A) or a 2S Li-ion/LiPo pack with a protection board and a fuse	B
Breadboard, jumpers, multimeter	The meter must have a 10 A (or at least 400 mA) current range; a second meter, or an oscilloscope, is useful	A, B

Build A: motor speed and direction with a TB6612FNG



```
 Current paths (forward, AIN1 = H, AIN2 = L, PWM on):   battery + -> VM -> high-side FET
 (reverse is the same through the other pair, with M1 supplied the other way round).  In
```

```
// Arduino Uno: pot sets the speed, the button reverses the motor (TB6612FNG on bridge A
const int PWMA = 9, AIN1 = 7, AIN2 = 8, POT = A0, BTN = 2;
bool forward = true;
bool lastBtn = HIGH;

void setup() {
  pinMode(AIN1, OUTPUT);  pinMode(AIN2, OUTPUT);  pinMode(PWMA, OUTPUT);
  pinMode(BTN, INPUT_PULLUP);
  TCCR1B = (TCCR1B & 0xF8) | 0x01;     // Timer 1 prescaler 1: pins 9 and 10 now run at
}

void drive(bool fwd, int duty) {
  digitalWrite(AIN1, fwd);               // forward: AIN1 high, AIN2 low;  reverse: the
  digitalWrite(AIN2, !fwd);
  analogWrite(PWMA, duty);               // 0 to 255: the duty cycle
}

void loop() {
  int duty = analogRead(POT) / 4;     // 0 to 1023 becomes 0 to 255
  bool btn = digitalRead(BTN);
  if (lastBtn == HIGH && btn == LOW) {   // the button has just been pressed
    analogWrite(PWMA, 0);  delay(300);        // stop the motor first: reversing at spee
    forward = !forward;
  }
  lastBtn = btn;
  drive(forward, duty);
  delay(20);                             // also debounces the button
}
```

Breadboard layout for the small parts (the module and the Uno connect by jumper wires as in the diagram; many modules come with header pins you can place straight into the board, in which case follow the pin names, not positions). Top rails: + rail = 5 V from the Uno, − rail = GND. Bottom rails: + rail = motor pack +, − rail = motor pack − and the common ground.

```
Grounds: jumper Uno GND -> top - rail;   jumper top - rail -> bottom - rail   (the one l
Potentiometer P1 (legs in row e): end A = column 5 row e,  wiper = column 6 row e,  end
   jumper column 5 row a -> top + rail;   jumper column 7 row a -> top - rail;   jumper
```

```
Button SW1 (across the centre gap, legs in rows e and f):   column 12 rows e/f  and  col
   jumper column 12 row a -> top - rail;   jumper column 14 row a -> Uno D2
Module power:   VCC -> top + rail;   GND -> top - rail;   VM -> bottom + rail;   STBY ->
Module signals: PWMA -> Uno D9;   AIN1 -> Uno D7;   AIN2 -> Uno D8
C2 (100 nF): column 20 row c -> column 22 row c;  jumper column 20 row a -> top + rail,
C1 (100 uF): + leg column 30 row c, - leg column 32 row c;  jumper column 30 row a -> bo
Motor: AO1 -> one motor wire, AO2 -> the other (C3 across the two motor terminals)
Connect the 5 V (USB) first, check that the module's power LED is lit (if it has one), c
Measure:   pin D9 average = red probe at the Uno D9 jumper; motor volts = red on AO1, bl
           battery current = meter in series with the pack + lead (10 A range);   all ot
```

1. Pot fully towards end B (GND), duty 0: the motor does not move; D9 reads 0 V.
2. Turn it a quarter of the way: duty 64/255 = 25 %, D9 about 1.25 V average, the motor voltage about 1.45 V, and the motor may still be at rest (stiction, section 2: it needs about 1.1 V to start). Turn it a little further and it starts to spin.
3. Half way: D9 about 2.5 V, motor volts about 2.95 V, speed about 3500 rpm (if you have a tachometer or a phone app). Three quarters: motor volts about 4.45 V, about 5700 rpm. Fully: motor volts 5.95 V, about 7900 rpm.
4. Press the button: the motor stops for 0.3 s and restarts in the other direction. Keep the meter on AO1 and AO2: the reading changes sign.
5. Hold the shaft briefly (no more than two seconds) at full speed: the current rises to about 1.09 A (6 V / 5.5 Ω, the 0.5 Ω being the driver) and the motor volts fall to about 5.5 V, the driver's drop. The chip warms; the speed does not return until you let go.
6. Remove C1 for a test (battery disconnected first): at start-up and at reversal, the 5 V reading of the logic may glitch as the supply wire dips, and the motor is noisier. With C1 fitted the supply is steadier.
7. Change the line `TCCR1B = ...` to a comment (power cycle the board): the PWM falls to 490 Hz and the motor whines audibly at low speed, and the motor current ripple (section 2) is about fifty times larger.

Build B: a power budget and a regulator comparison

B1, the budget. Copy the table in section 10 onto paper and replace the figures by your own: measure the current of each load in turn with the meter in series (the Uno from a 5 V supply through its 5 V pin with the meter in series, each sensor from the 5 V rail, and the motor on its own pack: running, then stalled for a second at most). Add the margins and check the sum against the regulator, the fuse, the wire and the battery.

B2, linear against switching. Feed the same load, R_L = 15 Ω from 5 V, from each regulator in turn, with 7.4 V at the input and the meter in series in the input lead.

```
Supply 7.4 V o----[ meter A (10 A range) ]----+----> regulator input (7805 pin 1, or th
                                              |
                                        [ C 0.33 uF ] (for the 7805)
                                              |
Supply 0 V o----------------------------------+----------------------> regulator GND (780

7805 (TO-220, front facing you, pins left to right: 1 INPUT, 2 GROUND, 3 OUTPUT):   OUT
Buck module: first connect the input only, set the output to 5.00 V with the trimmer us
Measure for each:  Vin (at the regulator), Iin (the meter in series), Vout (across R_L)
Compute:           Pin = Vin x Iin     Pout = Vout^2 / R_L     efficiency = Pout / Pin
```

B3, the brown-out, in the simulator only. The real test destroys cells; the model shows the effect safely. See the simulator notes below.

Simulators

Falstad Circuit Simulator (falstad.com/circuit). *H-bridge:* four MOSFETs from the *Transistors* group (P-channel at the top, N-channel at the bottom, so the gate drives are 0 V and 12 V logic levels), a 12 V source, and the motor modelled as a 5 Ω resistor in series with a 2 mH inductor (use Falstad's own DC motor element if your version lists one). Switch the gates by logic inputs (*Inputs and Sources*) so that top-left and bottom-right are on and the other two are off, and read the motor voltage on the scope: ±12 V. Then deliberately switch both transistors of one leg on: the source current jumps to hundreds of amps (the model has no wire resistance), which is shoot-through. Turn the PWM leg off and watch the current continue through the body diode of the other low-side FET (add a diode across each MOSFET, anode at the source of the N-channel). *Brown-out (B3):* a 5.2 V source in series with a 1.3 Ω resistor (the pack and wiring), a node N after the resistor, two 5 Ω resistors in parallel in series with a 2 mH inductor and a switch from N to ground for the motors, and from N a Schottky diode to a node L carrying a 1000 µF capacitor and a 22 Ω load (the logic). Close the switch and read N and L on the scope; open it after a moment to see the recovery. Without the switch closed, N = 5.2 V; closed, N falls to about 3.4 V (two 5 Ω windings: 5.2 / (1.3 + 2.5) = 1.37 A), and L, held by the capacitor, falls much more slowly at first. The 5 V regulator comparison is a calculation unless your Falstad version lists a regulator part.

Tinkercad Circuits (tinkercad.com): the component list has the Arduino Uno, a DC motor, a hobby servo, a potentiometer, a push button, a battery pack, a multimeter, an oscilloscope and an L293D motor driver (the TB6612 is not listed). For Build A, wire the L293D as follows and paste the same sketch: pin 1 (EN1) to D9, pin 2 (IN1) to D7, pin 7 (IN2) to D8, pin 3 (OUT1) and pin 6 (OUT2) to the motor, pin 8 (VS) to the 6 V battery +, pin 16 (VSS) to 5 V, pins 4, 5, 12 and 13 to GND, pins 9 (EN2), 10 and 15 to GND, with the battery − joined to the Uno GND. The L293D is a bipolar driver that drops roughly 1.5 to 2.5 V, so the motor voltage is lower than with the TB6612 (section 4); the simulated motor shows the shape of the behaviour, not the same numbers. Tinkercad runs the sketch's PWM at its own accuracy; the oscilloscope on D9 shows the duty cycle directly. A servo on D10 with the `Servo` library (`myservo.write(angle)`) reproduces section 5.

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
A: D9 average at pot 0 / 25 / 50 / 75 / 100 %	0 / 1.25 / 2.51 / 3.76 / 5.00 V (duty 0 / 25.1 / 50.2 / 75.3 / 100 %)	Within 0.1 V (the pot's end positions are never exact)	Same (the oscilloscope shows the duty)
A: D9 frequency	31.4 kHz with the Timer 1 line; 490 Hz without	31 kHz (a frequency meter or a scope)	Whatever the simulator models
A: motor average volts at the same positions	0 / 1.45 / 2.95 / 4.45 / 5.95 V (D × 6 V − 0.1 A × 0.5 Ω)	Within about 0.3 V (battery sag, motor and module differences)	1 to 2.5 V lower (L293D)
A: motor starts at a duty of	about 18 % (1.1 V)	15 to 30 %	Model dependent
A: speed at 25 / 50 / 75 / 100 %	1380 / 3560 / 5750 / 7930 rpm	±20 %	Model dependent
A: battery current at 25 / 50 / 75 / 100 %	25 / 50 / 75 / 100 mA (D × motor current of 0.10 A)	Within a factor of 1.5 (the motor's current rises a little with speed)	Model dependent
A: AIN1 / AIN2 forward, reverse	5 V / 0 V, 0 V / 5 V	4.9 to 5.0 V / 0.00 V	5 V / 0 V
A: stall current at full duty (held for under 2 s)	6 / 5.5 = 1.09 A; motor volts about 5.45 V	0.9 to 1.2 A	Model dependent

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
B2: 7805, Vin 7.4 V, R_L = 15 Ω	Vout 5.00 V, Iout 0.333 A, Iin 0.338 A, Pin 2.50 W, Pout 1.67 W, efficiency 66.6 %, loss 0.84 W	Vout 4.95 to 5.05 V; Iin 0.34 A; efficiency 65 to 67 %; case about 50 to 60 °C above ambient	(calculation)
B2: buck module at 88 %	Vout 5.00 V, Iin 0.256 A, Pin 1.89 W, loss 0.23 W	Iin 0.24 to 0.28 A; efficiency 85 to 93 %; module slightly warm; Vout ripple 20 to 60 mV p-p on a scope (the 7805 shows under 1 mV)	(calculation)
B2: input current saved by the buck	0.338 − 0.256 = 0.082 A (24 % less)	0.06 to 0.10 A	—
B3: node N with the motors off / on	5.2 V / 3.4 V	(not measured; do not test on real cells)	5.2 V / 3.4 V
B3: node L after closing the switch	Falls by 0.2 V per ms (I/C = 0.2 A / 1000 µF) from 4.9 V, then follows N minus 0.3 V	—	Falls and follows N minus the diode drop

Real readings spread because the motor, the pot and the batteries are all loose-tolerance parts. What does not vary is the structure: speed rises with the average voltage and starts only above the stiction threshold; reversing swaps the sign of the motor voltage; the stall current is limited only by the winding and driver resistances; the linear regulator's input current equals its output current, so its efficiency is V_{out}/V_{in} whatever the load, while the buck converter's input current is lower as the input voltage is higher; and a battery with internal resistance sags under a stalled motor by exactly $I \times R_{internal}$.

15. Challenges

CHALLENGE 1: THE MOTOR, PWM AND THE BRIDGE

(a) A 12 V motor has a winding resistance of 3.0 Ω, a no-load current of 0.15 A and a no-load speed of 6000 rpm at 12 V. Find K_e (in V·s/rad and V per krpm), K_t in mN·m/A, the stall current and the stall torque. At a load current of 0.8 A find the speed, the torque, the output power and the efficiency. Find the current and speed at maximum power and the maximum efficiency. Then fit a 30:1 gearbox of efficiency 0.65: find the output speed and torque at 0.8 A, and the output stall torque. (b) The winding

inductance is 2.5 mH: find the electrical time constant, the peak-to-peak current ripple at D = 0.5 at 1 kHz and at 20 kHz (compare with 0.15 A), and the average voltage and the no-load speed at D = 0.7. (c) In a 12 V H-bridge built from four MOSFETs of 20 mΩ each, find the shoot-through current if one leg conducts at the same time, the conduction loss at 6 A in the bridge, and the dead time (twice the turn-off time) for R_g = 47 Ω, C_{iss} = 3.3 nF, V_{drive} = 5 V, V_{th} = 2 V as a percentage of a 50 µs PWM period.

View solution

CHALLENGE 2: DRIVERS, SERVOS, STEPPERS AND RELAYS

(a) A 6 V motor draws 1 A on a 6 V supply. For an L298N (2.2 V lost at 1 A), a TB6612FNG (0.5 V) and a DRV8833 (0.36 V), find the motor voltage, the power in the driver and the share of the supply energy that reaches the motor, and say which suits a battery robot. (b) A servo takes 1.0 to 2.0 ms for 0 to 180° at 50 Hz. Find the pulse and the duty for 30° and for 120°, and the angle for a 1.72 ms pulse. Three servos move at 0.25 A each and stall at 0.65 A each: find the supply current in each case and say why the supply must be separate. (c) A NEMA 17 motor is rated 1.2 A per phase. Find the Vref for an A4988 (R_s = 0.068 Ω) and for a DRV8825 (R_s = 0.100 Ω); find the steps per revolution at 1/8 microstepping and the step frequency at 120 rpm; find the angle per step for a 0.9° motor at 1/16. (d) A 12 V relay has a 400 Ω coil: find the coil current and power; and compare the loss in a mechanical relay (contact resistance 0.05 Ω) and in a triac SSR (1.2 V drop) switching an 8 A load.

View solution

CHALLENGE 3: POWER, BROWN-OUTS AND WIRING

(a) A robot runs from a 2S LiPo (7.4 V, 1.8 Ah). Its motors draw 0.8 A at the battery, and a 5 V rail supplies 0.5 A. Compare a linear regulator (5 mA quiescent) and a buck converter at 88 %: find the input current, the power lost, the total battery current and the runtime at 80 % usable capacity; and find the temperature rise of the linear part at 65 °C/W and of an AMS1117 at 100 °C/W (and say whether it is acceptable). (b) The logic shares a battery of open-circuit 5.4 V, total resistance 1.15 Ω, with motors that draw a total of 3 A at start-up. Find the voltage at the logic, say whether a processor with a 2.7 V brown-out threshold resets, and find the voltage if the soft start limits the motor current to 1 A. Find the capacitance that holds the logic for 10 ms at 70 mA if the voltage may fall by 1.1 V, and say why that is not a cure for a stalled motor. (c) A 3 A load is 1.5 m from the battery. Find the drop and the loss in 22 AWG and in 18 AWG wire (out and back), and choose a fuse for the 3.8 A peak of the robot (a 22 AWG wire is rated 7 A). State the C-rate required of the 1.8 Ah pack for 3 A.

View solution

QUICK REFERENCE

- **DC motor:** $V = I R + K_e \omega$; $T = K_t I$; $K_e = K_t$ in SI; stall current V/R; no-load speed $(V - I_0 R)/K_e$; speed falls linearly with torque; maximum power at half stall torque; maximum efficiency $(1 - \sqrt{(I_0/I_{stall})})^2$; size every part for the stall current
- **Gear motor:** speed / N, torque × N × η_g; the motor's stall current is unchanged
- **PWM:** $V_{avg} = D \times V$; ripple $\Delta I = V D (1 - D) / (L f)$; use 20 kHz or more (silent, low ripple) if the driver allows; dead zone about 15 to 30 %; ramp the duty at start-up
- **H-bridge:** Q1+Q4 forward, Q2+Q3 reverse, Q3+Q4 brake, none = coast; never both switches of one leg (shoot-through = $V / (R_1 + R_2)$); dead time about $2 \times R_g C_{iss} \ln(V_{drive}/V_{th})$; body diodes give the freewheel path; loss $I^2(R_{high} + R_{low})$
- **Drivers:** L298N loses about 2.2 V (BJT; for 12 V and up); TB6612 about 0.5 V at 1 A (1.2 A continuous); DRV8833 about 0.36 V (1.5 A RMS, 10.8 V max); L293D 0.6 A; use a MOSFET driver below 12 V
- **Servo:** 50 Hz; 1 to 2 ms = 0 to 180° (1.5 ms centre); separate 5 to 6 V supply with 470 µF; common ground; stall 0.65 A (micro) to 2.5 A (large)
- **Stepper:** 200 steps per revolution (1.8°); bipolar with A4988 ($I = V_{ref} / (8 R_s)$, $R_s = 0.068\ \Omega$) or DRV8825 ($I = V_{ref} / (5 R_s)$, $R_s = 0.1\ \Omega$); supply 8 to 35 V with 100 µF at the module; microstep torque about sin(90°/m); never connect the motor live
- **Relay:** coil current V/R (70 Ω, 5 V: 71 mA); flyback diode across the coil; derate DC inductive contacts to about half; SSR: triac for AC only (1.2 V drop), MOSFET type for DC; no mains on a breadboard
- **Batteries:** LiPo 3.0 to 4.2 V per cell (3.7 V nominal, 2S 7.4 V); NiMH 1.0 to 1.4 V; runtime = Q × 0.8 / I; C-rate = I / Q; sag = $I R_{internal}$; protection, fuse, fire-safe charging; never below 3.0 V per cell
- **Regulators:** linear $P = (V_{in} - V_{out}) I$, efficiency V_{out}/V_{in}, needs dropout headroom (7805 2 V, AMS1117 1.1 V, LDO 0.2 V); buck $V_{out} = D V_{in}$, 85 to 95 %, noisy; boost $V_{out} = V_{in}/(1 - D)$
- **Budget and brown-out:** list typical and peak current per rail, add 30 %; start-up dip = $I_{start} \times R_{internal}$; below the processor's threshold (about 2.7 V on an Uno, about 2.4 V on an ESP32) it resets; cures: separate supplies, buck-boost, soft start, bulk capacitor ($t = C \Delta V / I$)
- **Capacitors and grounds:** bulk 100 to 1000 µF at every driver and servo, 100 nF at every chip, 100 nF across a brushed motor; star ground to the battery minus; thick wires for motor current
- **Protection and wire:** fuse above the peak and under the wire rating; polyfuse hold/trip; P-channel MOSFET for reverse polarity (loss I^2R) rather than a diode ($V_F I$); drop = $I \times R_{wire} \times 2 \times$ length (22 AWG 53 mΩ/m, 18 AWG 21 mΩ/m)

Next, Basic Electronics 10 puts a microcontroller board at the centre of these circuits: digital inputs and outputs, the ADC for sensors, hardware PWM for the motor of this chapter, interrupts

for encoders and bumpers, and how to read a sensor and drive a motor safely, with the most suitable board chosen for each example.

Basic Electronics

Course 1 · Chapter 10 · Microcontrollers in Circuits

Nine chapters have built the parts of a robot's electronics one at a time: voltage and current, passive parts, semiconductors, logic, motors and power. A microcontroller board is where they meet. Seen as a circuit part, it is a small chip that can *read* voltages (logic levels and analogue levels) and *produce* voltages (logic levels and PWM), with all the limits of any other part: a supply voltage it must stay inside, a few milliamps per pin, an input that expects a certain impedance and a certain range. This chapter is about those limits and about connecting the board to the circuits of Basic Electronics 1 to Basic Electronics 9 without damaging either side. It is not a programming course: each program is short, explained line by line only where the electronics matters, and every circuit is given twice, once for a real breadboard and once for a free browser simulator, with the readings you should expect.

Four boards are used, each where it is the most suitable one, and each time the reason is stated. The **Arduino Uno** (5 V logic, 10-bit ADC) is the board for 5 V sensors and for learning, and the one that Tinkercad Circuits simulates well. The **Raspberry Pi Pico** (3.3 V, 12-bit ADC, flexible 16-bit PWM on any pin, MicroPython) is the board for clean 3.3 V work and motor PWM. The **ESP32** (3.3 V, Wi-Fi and Bluetooth, many ADC channels and PWM channels) is the board when the robot must be commanded wirelessly. The **Raspberry Pi** (a Linux computer with GPIO pins and no ADC) is the board when the job is software-heavy (a camera, a network service, a screen) and the electronics is simple. Basic Electronics 11 compares them side by side and wraps up the course.

1. The board as one part in a circuit

Every board shows the same four faces to the outside world, whatever its name:

- **Power in.** A USB socket, a barrel jack or a pin (VIN, VSYS), feeding an onboard regulator (Basic Electronics 9) that makes the 5 V or 3.3 V the chip actually runs from. The regulator's spare current is what you may take from the 5 V or 3.3 V pin for sensors: a few tens to a few hundred milliamps, never motors or servos.
- **Ground.** Several GND pins, all one net. Every external supply that talks to the board by a signal must share this ground (the common ground of Basic Electronics 7 and Basic Electronics 9).

- **GPIO pins.** General-purpose input and output pins that are configured as a digital input, a digital output, an analogue input (ADC), a PWM output or a special function (serial, I2C, SPI).
- **Reset and boot behaviour.** At power-on and reset, GPIO pins are *inputs* (floating or pulled) until the program configures them, for tens to hundreds of milliseconds. Any circuit that must not act in that interval, such as a motor driver, needs a defined default (section 8).

The numbers that matter for each board

These are typical datasheet and manufacturer figures from memory; the exact limits depend on the exact chip, the exact board and the pin, so confirm them on the datasheet before you rely on them (Basic Electronics 11 returns to this).

	ARDUINO UNO (ATMEGA328P)	RASPBERRY PI PICO (RP2040)	ESP32 DEV BOARD	RASPBERRY PI (E.G. PI 4)
Logic level	5 V	3.3 V	3.3 V	3.3 V, **not** 5 V tolerant
Current per output pin	20 mA recommended, 40 mA absolute maximum; about 200 mA for the whole chip	4 mA by default, selectable up to 12 mA; about 50 mA total (check the datasheet)	About 20 mA recommended, 40 mA absolute maximum (varies with the pin and the pad setting)	16 mA per pin, about 50 mA across all pins (the commonly quoted figures)
ADC	10-bit, 6 channels, reference 5 V (or 1.1 V internal)	12-bit, 3 pins plus an internal temperature sensor; reference is the 3.3 V supply; effective resolution closer to 9 or 10 bits because of noise	12-bit, many channels; ADC2 cannot be used while Wi-Fi runs; nonlinear near 0 V and near full scale; needs attenuation set for a 0 to about 3.1 V range	None: add an external ADC chip (for example an MCP3008 over SPI, 10-bit)
PWM	6 pins, 8-bit, 490 or 980 Hz by default (timer registers change this)	16 channels on any pin, 16-bit counter with an adjustable frequency	LEDC peripheral: 16 channels, any output pin, resolution trades against frequency	2 hardware channels (4 pins), plus software PWM that jitters

	ARDUINO UNO (ATMEGA328P)	RASPBERRY PI PICO (RP2040)	ESP32 DEV BOARD	RASPBERRY PI (E.G. PI 4)
Interrupt inputs	External interrupts on D2 and D3 only; pin-change interrupts on all pins	Any GPIO	Any GPIO	Any GPIO, through the operating system (latency tens to hundreds of microseconds, not guaranteed)
Internal pull-up	20 to 50 kΩ	about 50 to 80 kΩ (typical)	about 45 kΩ (typical)	about 50 kΩ (typical)

DECOUPLE THE BOARD, NOT JUST THE CHIPS

A running processor draws current in sharp bursts, and a 100 nF capacitor across the supply at each chip (Basic Electronics 9) keeps those bursts off the rails. Boards already have these on the board, but a sensor or driver mounted on a breadboard still needs its own 100 nF at its supply pins, as close as the board allows. When a program works until a motor starts, the supply, not the code, is the first suspect.

2. Digital outputs and the pin-current budget

An output pin is a pair of tiny switches: one to the supply (drives the pin high) and one to ground (drives it low). It behaves as a voltage source with a small series resistance (about 25 to 40 Ω on an AVR pin), so it holds its voltage only while the current stays small. The first design rule is therefore a *current budget*: the current taken from any one pin, and from all pins together, stays within the figures of the table in section 1. The second is that the pin is a *signal* source: anything that needs more than a few milliamps is switched by a transistor or driver (Basic Electronics 6 and Basic Electronics 9) and the pin supplies only the base or gate current.

```
LED on a pin (series resistor, the Ohm's-law rule of Basic Electronics 1):     R = (V(p

  Uno (5 V pin), red LED with V(LED) = 2.0 V, target about 10 mA:    R = (5 - 2.0) / 0.(
  Pico (3.3 V pin), same LED, target about 4 mA (the default drive):  R = (3.3 - 2.0) /
       (the resistor dissipates I^2 R = 0.00394^2 x 330 = 5.1 mW: any 1/4 W resistor)
  The LED is a little dimmer at 3.3 V (4 mA) than at 5 V (9 mA), but still clearly visi

Pin-current budget for several LEDs (Uno, V(LED) = 2.1 V, 270 ohm each):     I = 2.9 / 
  Six of them together:    6 x 10.7 = 64 mA   (well under the 200 mA chip total)
  Twenty of them would need 215 mA: over the chip total, even though every single pin i
```

```
A load that needs more current (a relay coil, a motor, a strip) goes through a switch.
  NPN switch, forced gain 10:     I(base) = 71 / 10 = 7.1 mA          R(base) = (5 - 0.
  Same switch from a 3.3 V pin:    R(base) = (3.3 - 0.7) / 0.0071 = 364 ohm -> standard
       (more than the Pico's default 4 mA drive: use a logic-level MOSFET instead, whos
  The flyback diode across the coil is not optional (Basic Electronics 5 and Basic Elec
```

3. Digital inputs, pull-ups and pull-downs

An input pin is a very high impedance (about 1 µA leakage or less) that decides “high” above a threshold V_{IH} and “low” below V_{IL}, as in Basic Electronics 8. The point of an input is that, left unconnected, it *floats*: it picks up mains hum and your hand, and reads at random. A switch that connects the pin only to ground leaves it floating while open, so a resistor must hold it at a defined level. A **pull-up resistor** to the supply makes the pin read high when the switch is open and low when it is closed (the pin is “active low”). Every board has an internal pull-up that the program can turn on (`INPUT_PULLUP` on an Arduino, `Pin.PULL_UP` in MicroPython), which saves the external resistor.

```
Button to ground, pull-up resistor Rp to the supply Vcc:

  Vcc o-----[ Rp ]-----+----> input pin
                       |
                   [ switch ]
                       |
  0 V o----------------+

  Open:     the pin sits at Vcc (the input leakage of 1 uA across Rp = 10 k drops only
  Closed:   the pin is at 0 V, and the supply gives away  I = Vcc / Rp  while the butto

Choosing Rp is a compromise:
  too small  ->  wastes current when pressed (Vcc / Rp) and loads the contact more than
  too large  ->  slow edges (rise time about 2.2 x Rp x C(pin and wire)) and a pin easi

  Rp          I when pressed, 5 V      I when pressed, 3.3 V      rise time with 150 pF
  4.7 k           1.06 mA                  0.70 mA                  2.2 x 4700 x 150e-1
  10 k            0.50 mA                  0.33 mA                  3.3 us
  50 k (internal) 0.10 mA                  0.066 mA                 16.5 us
  For a switch on a short wire the internal pull-up (about 50 k) is fine.  For a long c
```

```
A button also bounces for 1 to 5 ms on closing (Basic Electronics 8): debounce it in so
```

4. The ADC: turning a voltage into a number

An **analogue-to-digital converter** compares the input voltage with a reference voltage V_{ref} and returns a whole number from 0 to $2^N - 1$ for an N-bit converter. The step size, one **LSB** (least significant bit), is the smallest change it can report. The two formulas you need are the code for a given input and the voltage for a given code.

```
N-bit ADC, reference Vref:     LSB = Vref / 2^N          code = floor( Vin / Vref x 2^N
  The quantisation error is up to +/- half an LSB (reading (code + 0.5) x LSB removes t

  Uno      N = 10, Vref = 5 V:       LSB = 5 / 1024 = 4.883 mV         reading 512:  51
  Pico     N = 12, Vref = 3.3 V:     LSB = 3.3 / 4096 = 0.806 mV       reading 2048: 20
  Uno with the 1.1 V internal reference:   LSB = 1.1 / 1024 = 1.074 mV  (4.5 times fine

Reading a TMP36 temperature sensor (output V = 0.5 V + 10 mV per degree C, so 0.750 V a
  Uno at 5 V:      one LSB = 4.883 mV = 0.488 C       at 25 C: Vin = 0.750 V   code = f
                   temperature = (153 x 5 / 1024 - 0.5) x 100 = (0.7471 - 0.5) x 100 =
  Uno at 1.1 V:    one LSB = 1.074 mV = 0.107 C       a range from -50 C to 60 C: the f
  The reference sets the scale: a 5 V reference that droops to 4.8 V under load reads e
  measure the board's real 5 V with a meter and put that figure in the program for bett
```

Fitting a larger voltage into the ADC: the divider

The ADC input must stay between 0 V and V_{ref} (and never above the board's supply, or the chip is damaged). A battery pack is higher, so a resistive divider (Basic Electronics 1) scales it down. Design it in three steps: choose the ratio from the full-charge voltage, then pick resistors so the source impedance the ADC sees is not too high, then add a small capacitor.

```
Divider:  V(pin) = V(bat) x R2 / (R1 + R2)      the ratio  k = R2 / (R1 + R2)  must sat
Battery voltage from the code:     V(bat) = code x LSB / k          one LSB of the batt
The ADC input prefers a source impedance of about 10 k or less (the Uno's datasheet rec
holds the input steady (it gives the ADC's sampling capacitor a reservoir) and filters

Example 1 (the plan for Circuit 2 below, Pico):   a 4-cell NiMH pack, 5.6 V full (1.4 V
```

```
        full 5.6 V:    pin = 2.80 V   code = floor(2.80 / 3.3 x 4096) = 3475          (
        low  4.4 V:    pin = 2.20 V   code = floor(2.20 / 3.3 x 4096) = 2730
        One LSB of the battery = 0.806 mV / 0.5 = 1.61 mV.   Source impedance 10 k || 10
Example 2 (Uno measuring a 3-cell LiPo, 12.6 V full):   R1 = 22 k, R2 = 10 k:   k = 10 /
        12.6 V:  pin = 3.9375 V   code = floor(3.9375 / 5 x 1024) = 806        one LSB o
Example 3 (Pico measuring the same 12.6 V pack):   R1 = 33 k, R2 = 10 k:   k = 10 / 43 =
        source impedance 7.7 k, one LSB of the battery = 0.806 mV / 0.2326 = 3.46 mV.
A pin with a divider whose output could exceed the supply (a battery plugged in with th
never connect the divider to the pack before checking the divider's output with a meter
```

NEVER PUT MORE THAN THE SUPPLY ON AN INPUT

The absolute maximum on a pin is about the supply plus 0.3 V (3.3 V boards: 3.6 V). A 5 V sensor output on a 3.3 V board pin, or a divider that you built backwards, kills the pin, and sometimes the chip. Section 7 shows the level-shifting divider; for any signal that could be higher than the pin's supply, measure it first.

5. PWM: duty cycle, average voltage and resolution

Basic Electronics 8 built PWM from a 555 and Basic Electronics 9 used it for a motor. A microcontroller makes it with a hardware **timer**: a counter running from the chip's clock f_{clk} counts from 0 up to a *top* value and the output is high while the counter is below a *compare* value. The frequency is the clock divided by the number of steps; the duty cycle is the compare value divided by the top.

```
Duty D = compare / top        Average voltage = D x V(high)         (the motor or an RC
  Uno   analogWrite(pin, 128):  D = 128 / 255 = 50.2 %      average = 0.502 x 5 V = 2.5

The steps, the frequency and the resolution are tied together:        steps = f(clk) /
  Pico  (125 MHz clock) at 20 kHz:    125e6 / 20e3  = 6250 steps    -> 12 bits (6250 le
  ESP32 (80 MHz clock for the LEDC timers) at 20 kHz:   80e6 / 20e3 = 4000 steps  -> 11
  Uno   Timer 1 in 8-bit phase-correct mode:     16e6 / 510 = 31.4 kHz with 255 levels
  The trade-off in one line: doubling the frequency halves the number of levels.  20 kH

PWM as a cheap digital-to-analogue converter:  a resistor and a capacitor (Basic Electr
  The ripple on the capacitor (peak to peak) is approximately   dV = V x D (1 - D) / (R
       at 490 Hz:      5 x 0.25 / (10000 x 1e-6 x 490)     = 0.255 V peak to peak     (
```

```
        at 31 372 Hz:   5 x 0.25 / (10000 x 1e-6 x 31372)   = 3.98 mV peak to peak     (
    So a fast PWM and a slow RC give a decent analogue output.  The price is a settling t
```

Mapping a command onto a motor

Two corrections from Basic Electronics 9 belong in any motor program: a **dead zone** (the motor will not move below about 18 % duty, so a command of 1 % must map to 0, and a command of 100 % to 100 %, with the rest spread linearly between 18 % and 100 %) and a **ramp** (the duty rises at a limited rate, such as 2 % every 20 ms, so a change from 0 to 100 % takes 1.0 s and the start-up current is not a stall pulse). For a command c between 0 and 1 above a small threshold, the duty is 18 + 82 × c per cent: half way (c = 0.5) is 59 %, which on a 4.8 V battery is an average of 0.59 × 4.8 = 2.83 V.

6. Interrupts: reacting to an edge without polling

A program that checks a pin in a loop (*polling*) can miss a pulse shorter than the loop. An **interrupt** makes the hardware stop the main program the instant a pin changes, run a short **interrupt service routine** (ISR), and return to exactly where it left off. It is the right tool for a pulse you must not miss: a wheel encoder, a bumper switch, a fast echo or a stop button. The cost is paid in the ISR's run time, and in three rules: keep it short, share data through a variable declared `volatile` (Arduino C++) or a plain global (MicroPython), and do not print or wait inside it.

```
A wheel encoder: one sensor pulse per revolution of a slotted disc on the MOTOR shaft,
  Motor at 6900 rpm = 115 rev/s       pulse rate = 115 x 12 = 1380 interrupts per secon
  An ISR of about 10 us (an Arduino's overhead plus a counter increment):       CPU loa

Speed by counting in a fixed window (here 100 ms):
  counts in the window = 1380 x 0.1 = 138          motor rpm = counts x 60 / (pulses pe
  One count of resolution = 60 / (12 x 0.1) = 50 rpm of the motor = 1 rpm of the output
  A rule of thumb:  count over a window that gives at least 50 to 100 counts, so one co

The shared-variable trap on an 8-bit Arduino: a 16- or 32-bit counter is copied in seve
       noInterrupts();   copy = counter;   interrupts();
Only the Uno's D2 and D3 are true external interrupt pins; on the Pico and the ESP32 an
counted at 1000 pulses per second is better handled by a Pico or an Arduino beside the
```

7. Level shifting: mixing 3.3 V and 5 V

Basic Electronics 8 introduced level shifting. In the other direction (5 V output to a 3.3 V input) you can use a resistive divider, which is the cheapest shifter and works for any signal slower than a few megahertz.

```
5 V signal -> 3.3 V pin:        signal o-----[ R1 ]-----+-----> 3.3 V input        V(o
                                                        |
                                                     [ R2 ]
                                                        |
                                                 0 V o------+

  R1 = 1 k, R2 = 2 k:     V(out) = 5 x 2000 / 3000 = 3.33 V       (the pin's V(IH) for
                           divider current = 5 V / 3 k = 1.67 mA          source impeda
                           speed: with 60 pF of pin and wire the time constant is 667 x
  R1 = 1.8 k, R2 = 3.3 k:  V(out) = 5 x 3.3 / 5.1 = 3.24 V,  0.98 mA, 1.16 k, slightly
  The usual use:  the 5 V echo pin of an HC-SR04 ultrasonic sensor into a Raspberry Pi

3.3 V signal -> 5 V pin:   usually no shifter is needed for an Uno:  its V(IH) is about
  or the margin matters, use a logic-level shifter module (a MOSFET-based bidirectional
  A divider cannot go UP in voltage, and cannot shift a bidirectional line such as I2C:
```

8. Driving a motor from a board, safely

A motor driven by a board adds three new ways to fail: the board may start up with the motor pins in an undefined state, the program may hang with the motor on, and the motor's own supply may dip enough to reset the board mid-move (Basic Electronics 9). The remedy is not clever code but a short list of habits, each of which costs a resistor or a line of code.

- **A default of off, in hardware.** Put a 100 kΩ pull-down resistor on every driver input (the PWM pin, the direction pins, and the standby pin held low means asleep), so that while the board is booting and its pins float the motor is not driven.
- **Separate supplies, common ground.** The motor runs from its own battery through the driver; the board runs from USB or a regulator; the only electrical link between them is the common ground and the signal lines.
- **Start low and ramp.** Power up the program with the duty at 0, and raise it by a few per cent per cycle.

- **A watchdog.** Most boards have a timer that resets the chip if the program has not "fed" it recently. If the program hangs, the watchdog resets the board and the pull-downs switch the motor off. (In MicroPython: `WDT(timeout=2000)`, then `wdt.feed()` every time round the loop.)
- **Watch the battery.** A divider into an ADC pin (section 4) lets the program cut the motor when the pack falls below its safe limit, and a fuse in the pack lead covers what the program cannot.
- **Watch the current.** A motor that stalls draws V/R (Basic Electronics 9). A small sense resistor in the motor's ground return turns that current into a voltage the ADC can read: a 0.22 Ω resistor gives 0.22 V per amp, so the 1.09 A stall of motor M1 reads 0.24 V (297 counts on a Pico) against 0.055 V (68 counts) at the normal 0.25 A, and the program stops the motor if the reading stays above about 0.8 A (0.176 V, 218 counts) for half a second. The resistor wastes up to 0.24 V of the motor's supply and dissipates $0.22 \times 1.09^2 = 0.26$ W at stall: use a 0.5 W part.
- **A physical way to stop.** A switch in the battery lead (a mechanical relay or a plain toggle) beats any software when something goes wrong.

9. The right board for each job

JOB	BEST BOARD	WHY (AND WHAT THE OTHERS LACK)
Reading a 5 V analogue sensor (TMP36, a 5 V line sensor array) and flashing outputs; the first board to learn on	Arduino Uno	5 V logic and 5 V ADC range match 5 V sensors without shifting; 6 ADC pins; Tinkercad Circuits simulates it well; a mistake rarely destroys it, since a 40 mA pin and a socketed chip are forgiving. The Pico and the ESP32 would need level shifting on 5 V sensors, and a Raspberry Pi has no ADC.
Motor PWM at 20 kHz with 3.3 V drivers and a clean ADC measurement of a battery	Raspberry Pi Pico	PWM at 20 kHz on any pin with 12-bit resolution (an Uno needs register tricks to leave its 490 Hz default); 3.3 V matches the TB6612 and DRV8833 logic; low cost; MicroPython and Wokwi make it quick to test. Not the best choice if you need wireless or lots of ADC channels.
A robot driven from a phone, with Wi-Fi or Bluetooth built in	ESP32	Radio on the chip, two cores (one for networking), many ADC and PWM channels. Its limits: ADC2 is unavailable while Wi-Fi is running; the ADC is noisy and nonlinear near the ends; 3.3 V only.

JOB	BEST BOARD	WHY (AND WHAT THE OTHERS LACK)
A camera, a screen, a database or an AI model, with a few simple switches or LEDs	Raspberry Pi	A full Linux computer: Python libraries, networking, USB devices. Not good at hard timing (an encoder at 1000 pulses per second) or at analogue input (no ADC), so a Pico or an Arduino usually handles the motors and sensors beside it.

10. Build it: four circuits, each on a breadboard and in a simulator

Each circuit below names its board and the reason, then gives the parts, the wiring (as pin-to-pin connections, because the hole positions on your own breadboard will differ), the program and the simulator steps. Menu names in the simulators change from time to time; where I am less sure of an exact name the text says so, and searching the part list by name always works.

> **BEFORE YOU POWER ANYTHING**
>
> Check the wiring against the list with the power *off*, then connect USB last. Use a meter to confirm every divider output is within the pin's range before the pin is connected. Do not connect a motor pack until the program is running and the motor driver inputs read 0 V.

Circuit 1 — Arduino Uno: a TMP36 temperature reading, a debounced interrupt button and an LED

Why the Uno: the TMP36 is a 5 V analogue sensor, which suits the Uno's 5 V supply and 5 V ADC range directly; D2 is a true interrupt pin; Tinkercad simulates this circuit with the sensor's temperature adjustable by hand.

Parts: Arduino Uno, TMP36 (TO-92 package), push button, red LED, 330 Ω resistor, 100 nF capacitor, breadboard and jumpers.

```
Wiring (breadboard):
  TMP36, flat face towards you, pins left to right:   1 = +Vs,  2 = Vout,  3 = GND
      pin 1 -> Uno 5V          pin 2 -> Uno A0          pin 3 -> Uno GND         100 nF
  Button:   one leg -> Uno D2,  the opposite leg -> Uno GND   (no resistor: the program
  LED:      Uno D8 -> 330 ohm -> LED anode (long leg);  LED cathode (short leg, flat si
  Use the breadboard's rails for 5 V and GND: Uno 5V to the top + rail, Uno GND to the
  Check the TMP36 orientation first: reversed, it gets hot in seconds.  Feel it with a
```

```
const int PIN_BUTTON = 2;
const int PIN_LED = 8;
const int PIN_TEMP = A0;
const float VREF = 5.0;               // measure the real 5 V pin with a meter and put i

volatile unsigned long lastPress = 0;
volatile unsigned int pressCount = 0;
volatile bool ledOn = false;

void onButton() {
    unsigned long now = millis();
    if (now - lastPress > 20) {        // ignore bounces for 20 ms
        pressCount++;
        ledOn = !ledOn;
        digitalWrite(PIN_LED, ledOn);
        lastPress = now;
    }
}

void setup() {
    Serial.begin(9600);
    pinMode(PIN_LED, OUTPUT);
    pinMode(PIN_BUTTON, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(PIN_BUTTON), onButton, FALLING);
}

void loop() {
    int code = analogRead(PIN_TEMP);
    float volts = code * VREF / 1024.0;
    float tempC = (volts - 0.5) * 100.0;

    noInterrupts();                   // copy the shared counter safely
    unsigned int presses = pressCount;
    interrupts();

    Serial.print("code ");
    Serial.print(code);
    Serial.print("  volts ");
    Serial.print(volts, 3);
    Serial.print("  tempC ");
    Serial.print(tempC, 1);
    Serial.print("  presses ");
    Serial.println(presses);
```

```
    delay(500);
  }
```

Simulator, Tinkercad Circuits (tinkercad.com, free account): create a new Circuit; from the component list drag in an Arduino Uno, a breadboard (small), a *Temperature Sensor [TMP36]*, a pushbutton, a red LED and a resistor (set it to 330 Ω in the part's properties). Wire as above (the sensor's pins are labelled when you hover; the pushbutton has four legs joined in pairs, so use two diagonally opposite legs). Open *Code*, choose *Text*, paste the program, press *Start Simulation* and open the *Serial Monitor*. Click the sensor and drag its temperature slider (the range is wide, from about −40 to 125 °C) to 25 °C; click the button in the simulation.

Circuit 2 — Raspberry Pi Pico: a battery monitor, a speed knob and 20 kHz motor PWM, with a watchdog

Why the Pico: the TB6612FNG accepts 3.3 V logic, the Pico makes 20 kHz PWM on any pin with 12-bit resolution, and its ADC measures a 5 V-class battery through a divider of two equal resistors. The program is the short list of habits of section 8: an off default, a ramp, a dead-zone map, a low-battery cut-off and a watchdog.

Parts: Raspberry Pi Pico with headers, TB6612FNG module, a small 6 V hobby motor (M1 of Basic Electronics 9), 4 × AA NiMH cells in a holder (4.8 V), 2 A fast-blow fuse in the pack's positive lead, two 10 kΩ resistors (divider), 100 nF, 100 µF, 10 kΩ potentiometer, four 100 kΩ resistors (pull-downs), a switch in the pack lead, breadboard.

```
  Wiring (Pico pin numbers are the physical pin positions; GP numbers are the names used
    Pack +   -> switch -> fuse -> TB6612 VM          and the same point -> 10 k (R1) -> n
    Node N   -> Pico GP26 (pin 31, ADC0);   100 nF from node N to GND, close to the Pico
    Pack -   -> TB6612 GND and Pico GND (pin 38)            (the common ground, one poin
    100 uF across TB6612 VM and GND (+ leg to VM), at the module;   100 nF across the mot
    TB6612 VCC -> Pico 3V3(OUT) (pin 36)
    TB6612 PWMA -> Pico GP15 (pin 20);   AIN1 -> GP14 (pin 19);   AIN2 -> GP13 (pin 17);
    100 k from each of PWMA, AIN1, AIN2 and STBY to GND  (the hardware default is OFF)
    Motor -> TB6612 AO1 and AO2
    Speed knob: pot end 1 -> Pico 3V3(OUT), pot end 2 -> GND, wiper -> GP27 (pin 32, ADC1
    Power the Pico from USB.   Connect the pack's switch LAST, after the program is runni
```

```
from machine import ADC, Pin, PWM, WDT
import time

R1 = 10000.0                # divider top resistor
R2 = 10000.0                # divider bottom resistor
VREF = 3.3                  # the Pico's ADC reference is its 3.3 V supply
CUTOFF_V = 4.4              # low-battery limit for 4 NiMH cells
DEAD = 18.0                 # duty (%) below which the motor does not turn
STEP = 2.0                  # ramp: at most 2 % per 20 ms (0 to 100 % in 1 s)

batt_adc = ADC(Pin(26))
knob_adc = ADC(Pin(27))

stby = Pin(12, Pin.OUT, value=0)    # asleep until we say so
ain1 = Pin(14, Pin.OUT, value=0)
ain2 = Pin(13, Pin.OUT, value=0)
pwm = PWM(Pin(15))
pwm.freq(20000)
pwm.duty_u16(0)

def read_fraction(adc, samples=16):
    total = 0
    for _ in range(samples):
        total += adc.read_u16()
    return total / samples / 65535      # 0.0 to 1.0 (full scale reads about 65520, a 0.(

def battery_volts():
    pin_volts = read_fraction(batt_adc) * VREF
    return pin_volts * (R1 + R2) / R2

def set_duty(percent):
    percent = max(0.0, min(100.0, percent))
    pwm.duty_u16(int(percent / 100 * 65535))

ain1.value(1)               # forward: AIN1 high, AIN2 low
ain2.value(0)
stby.value(1)
wdt = WDT(timeout=2000)     # reset the board if the loop stops for 2 s
speed = 0.0
tick = 0

while True:
    wdt.feed()
    batt = battery_volts()
```

```
    knob = read_fraction(knob_adc)

    if batt < CUTOFF_V or knob < 0.05:
        target = 0.0
    else:
        target = DEAD + (100.0 - DEAD) * knob

    if target > speed:
        speed = min(target, speed + STEP)
    else:
        speed = max(target, speed - STEP)
    set_duty(speed)

    tick += 1
    if tick % 25 == 0:        # print twice a second
        print("battery %.2f V  knob %.2f  target %.0f %%  duty %.0f %%" % (batt, knob, t
    time.sleep_ms(20)
```

Simulator, Wokwi (wokwi.com, free): start a new project for the *Raspberry Pi Pico* with *MicroPython* (the starter list names may differ slightly) and replace the sample `main.py` with the program above. Wokwi has no TB6612 that I can count on (check the part list; if one is offered, wire it as above), so represent the driver input by an LED: from GP15 through a 330 Ω resistor to an LED and to GND, and use *two potentiometers* (the + button opens the parts list; the Wokwi potentiometer part has three pins, a signal pin `SIG` and the two ends): one on GP27 as the speed knob, one on GP26 as the divider midpoint (with the ends on 3V3 and GND, the wiper reads 0 to 3.3 V, which stands in for the 0 to 6.6 V a 10 k / 10 k divider would pass). Start the simulation and watch the terminal text under the editor. Turn the GP27 pot and the LED's brightness follows the duty (the LED averages the 20 kHz pulses, as your eye would). If your Wokwi version offers a logic analyzer part, connect it to GP15 to see the pulse widths.

Circuit 3 — ESP32: a potentiometer commanding a hobby servo

Why the ESP32: the same circuit works with a pot today and a command from a phone over Wi-Fi tomorrow, without a second board; the 16-channel LEDC peripheral can make the 50 Hz servo pulse on almost any pin. The pot goes on an ADC1 pin (GPIO34), not an ADC2 pin, because ADC2 stops working when Wi-Fi is on. The servo's own supply is separate, as in Basic Electronics 9.

Parts: ESP32 dev board (30- or 38-pin), SG90 or similar 9 g servo, 4 × AA NiMH holder (4.8 V), 470 µF electrolytic, 220 Ω resistor, 10 kΩ potentiometer, breadboard.

```
 Wiring (the board's labels are printed on it: use the GPIO numbers):
   Servo supply: pack + -> servo red wire;   pack - -> servo brown or black wire;   470 
   Pack - also -> ESP32 GND                          (the ONLY link between the supplie
   Servo signal (orange or yellow) -> 220 ohm -> ESP32 GPIO18
   Pot: one end -> ESP32 3V3, other end -> GND, wiper -> ESP32 GPIO34 (an input-only pin
   Power the ESP32 from USB, connect the servo pack last.
```

```
from machine import ADC, Pin, PWM
import time

adc = ADC(Pin(34))
adc.atten(ADC.ATTN_11DB)             # about 0 to 3.1 V range: nonlinear near the ends, 
servo = PWM(Pin(18), freq=50)

def angle_to_duty(angle):
    angle = max(0.0, min(180.0, angle))
    pulse_ms = 1.0 + angle / 180.0   # 1.0 ms = 0 deg, 2.0 ms = 180 deg
    return int(pulse_ms / 20.0 * 65535)

def read_fraction(samples=8):
    total = 0
    for _ in range(samples):
        total += adc.read_u16()
    return total / samples / 65535

while True:
    angle = read_fraction() * 180.0
    servo.duty_u16(angle_to_duty(angle))
    time.sleep_ms(20)
```

Simulator, Wokwi: start a new *ESP32* project with MicroPython (the project templates may change names), add a servo and a potentiometer from the parts list, wire the servo's signal to GPIO18, its supply pins to 5V and GND (the simulator does not need the separate pack; the real circuit does) and the potentiometer's wiper to GPIO34, its ends to 3V3 and GND. Paste the program, start the simulation and drag the pot slider: the servo horn follows. At the middle of the pot the program calls `angle_to_duty(90)`.

Circuit 4 — Raspberry Pi: a button, an LED and an HC-SR04 echo through a divider

Why the Raspberry Pi: it is the right board when the job needs an operating system (a web page, a camera, a screen) and a handful of simple signals. It is the wrong board for an analogue sensor (no ADC) or a precise pulse train (the operating system can delay code by a fraction of a millisecond). Its GPIO pins are 3.3 V and not 5 V tolerant, so the HC-SR04's 5 V echo signal goes through the 1 k / 2 k divider of section 7.

Parts: any Raspberry Pi with the 40-pin header and Raspberry Pi OS, push button, red LED, 330 Ω resistor, HC-SR04 ultrasonic sensor, 1 kΩ and 2 kΩ resistors, breadboard.

```
Wiring (physical header pin numbers; GPIO numbers are the names the program uses):
  Button:  one leg -> GPIO17 (pin 11), other leg -> GND (pin 9)             (the progra
  LED:     GPIO27 (pin 13) -> 330 ohm -> LED anode;  LED cathode -> GND (pin 14)
  HC-SR04:  VCC -> 5 V (pin 2);  GND -> GND (pin 6);  TRIG -> GPIO23 (pin 16)
            ECHO -> 1 k -> GPIO24 (pin 18);  the junction of that 1 k and the pin also
  Never connect ECHO straight to a GPIO pin.
```

```
from gpiozero import Button, LED, DistanceSensor
from signal import pause
import time

led = LED(27)
button = Button(17, bounce_time=0.02)     # internal pull-up on by default; 20 ms debou
button.when_pressed = led.toggle

sensor = DistanceSensor(echo=24, trigger=23, max_distance=2.0)

try:
    while True:
        print("distance %.2f m" % sensor.distance)
        time.sleep(0.5)
except KeyboardInterrupt:
    pass
```

Simulator note: I do not know of a free browser simulator that runs Raspberry Pi OS with its GPIO, so the code is tested on the real Pi. The *circuit* (the divider and the LED) can be checked in Falstad (below) and the logic of the button and LED in Wokwi, which also provides a Pico and several Arduino-compatible boards, with the equivalent MicroPython calls (`Pin(17, Pin.IN,`

`Pin.PULL_UP)` and `Pin(27, Pin.OUT)`). Do not rely on a simulator's sensor part for the echo timing.

Falstad: the divider, the LED and the PWM filter

Falstad Circuit Simulator (falstad.com/circuit) has no microcontroller, but it checks every circuit-level calculation of this chapter. Part names are in the *Draw* menu (menu names may differ a little between versions; the right-click menu on a part offers its settings). *Level-shift divider:* a voltage source set to a square-wave (a 2-terminal source from *Inputs and Sources*; set the waveform to square, amplitude 2.5 V and DC offset 2.5 V, so it swings from 0 to 5 V), a 1 kΩ resistor, a 2 kΩ resistor to ground and the scope on the junction: the high level reads 3.33 V. Change 2 k to 1 k and the high level falls to 2.50 V. *LED:* a 3.3 V DC source, a 330 Ω resistor and a LED (*Diodes* group, or *Outputs and Labels*, depending on version): the current reads about 3.9 mA with a 2.0 V LED model (Falstad's default LED model may drop a little more than 2.0 V, so expect 3.5 to 3.9 mA). *PWM filter:* a square-wave source (0 to 5 V, 490 Hz, 50 % duty, which the source's duty setting provides), a 10 kΩ resistor, a 1 µF capacitor to ground, and the scope on the capacitor: the average is 2.5 V with a ripple of about 0.26 V peak to peak; change the frequency to 31.37 kHz and the ripple falls to about 4 mV.

Expected readings

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
1: TMP36 output at 0 / 25 / 50 °C	0.500 / 0.750 / 1.000 V	Within about ±2 °C (the TMP36 is accurate to about ±2 °C at 25 °C)	Same
1: ADC code at 0 / 25 / 50 °C (5 V reference)	102 / 153 / 204	±2 counts, and the real reference may be 4.7 to 5.1 V	102 / 153 / 204
1: temperature printed at those codes	−0.2 / 24.7 / 49.6 °C	±2 °C of the true temperature; the digits jump by 0.5 °C (one LSB)	Same
1: D2 idle / pressed	5 V / 0 V (internal pull-up gives 0.1 to 0.25 mA when pressed)	4.9 to 5.0 V / under 0.1 V	5 V / 0 V
1: LED current, D8 high	9.1 mA (3.0 V across 330 Ω)	8 to 9.5 mA	About 9 mA
1: presses counted per press	1 (the 20 ms debounce)	1; two if the contacts are very noisy: raise the delay to 50 ms	1

MEASUREMENT	CALCULATED	REAL (TYPICAL)	SIMULATOR
2: divider output at 5.6 / 4.8 / 4.4 V pack	2.80 / 2.40 / 2.20 V; ADC counts 3475 / 2978 / 2730	Within 20 counts (the Pico's ADC has noise and an offset error)	Whatever the pot gives
2: duty at knob 0.5	59 % (18 + 82 × 0.5); duty_u16 = 38 665	Printed target 59 %; speed ramps up in 0.5 s	LED at about 59 % brightness
2: motor average volts at knob 0.5 on 4.8 V	0.59 × 4.8 = 2.83 V, minus the driver's small drop	2.6 to 2.8 V on the motor terminals (a DC range, meter averages)	(no driver)
2: PWM frequency on GP15	20 000 Hz; period 50 µs	20 kHz on a frequency meter	Logic analyzer: 20 kHz (if the part is offered)
2: battery below 4.4 V	Target 0 %, the duty ramps down to 0 in under 1 s	Same	Set the stand-in pot below the 2.2 V point
3: servo pulse at pot 0 / 0.5 / 1.0	1.0 / 1.5 / 2.0 ms; duty_u16 3277 / 4915 / 6554	0 / 90 / about 180 degrees ± 10 (servos vary)	Horn at about 0 / 90 / 180 degrees
3: a 1.72 ms pulse	129.6 degrees; duty_u16 5636	—	—
3: ADC reading near full scale (3.3 V on the pot)	Clamped by the program at 180 degrees	May read 3.1 to 3.2 V at the top because of the ADC's attenuation; the horn may stop a few degrees short	Reaches full scale
4: V(GPIO24) with a 5 V echo through 1 k / 2 k	3.33 V	3.2 to 3.4 V	3.33 V in Falstad
4: V(GPIO24) with 1 k / 1 k	2.50 V (only 0.2 V above the 2.3 V threshold)	2.4 to 2.55 V: works, with little margin	2.50 V
4: LED current	3.9 mA	3.5 to 4.0 mA	3.5 to 3.9 mA
PWM filter ripple, 10 k / 1 µF, D = 0.5, 5 V	0.255 V p-p at 490 Hz; 3.98 mV p-p at 31.4 kHz	Within 20 %	0.25 V, 4 mV

Real readings spread because the resistors, the sensor and the supplies are all loose-tolerance parts and the ADC reference is only as steady as the supply. The structure does not vary: the code follows the pin voltage to within one or two LSB; the duty determines the average to within the driver's drop; a pull-up holds an input high until the switch pulls it low; and a resistive divider gives the ratio of its resistors whatever the signal.

11. Challenges

CHALLENGE 1: PINS, PULL-UPS AND THE ADC

(a) A red LED (2.1 V) is driven from an Uno pin at about 12 mA. Find the calculated resistor, choose the nearest value from 220, 270, 330, 390 Ω and find the actual current. How many such LEDs can the Uno drive at once (6 LEDs, 12 LEDs, or 20 LEDs) if the chip total is 200 mA and each pin is limited to 20 mA? (b) A button on a 3.3 V board has an external 4.7 kΩ pull-up. Find the current when pressed and the rise time with 150 pF. Compare with the internal 50 kΩ pull-up. (c) Find the LSB of an Uno (10 bit, 5 V) and of a Pico (12 bit, 3.3 V). What voltage does a reading of 300 on the Uno represent, and what temperature is that from a TMP36? What reading do you expect on the Pico for 1.65 V? (d) Design a divider to measure a 3-cell LiPo pack (12.6 V) with a Pico, using 33 kΩ and 10 kΩ: find the pin voltage at 12.6 V and 11.1 V, the codes, the source impedance and the size of one LSB at the battery.

View solution

CHALLENGE 2: PWM AND INTERRUPTS

(a) An Uno writes `analogWrite(pin, 90)`. Find the duty cycle and the average voltage on a 5 V pin and on a 6 V motor supply. (b) For PWM at 25 kHz, find the number of steps and the resolution in bits of a Pico (125 MHz) and an ESP32 (80 MHz), and the frequency of the Uno's Timer 1 in 8-bit phase-correct mode (16 MHz, 510 steps). (c) A PWM output at 5 V and D = 0.5 feeds an RC filter of 4.7 kΩ and 2.2 µF. Find the time constant, the ripple at 490 Hz and at 31.4 kHz, and the time to settle within 1 part in 256. (d) A motor at 9000 rpm has a 7-pulse-per-revolution encoder on its shaft and a 30:1 gearbox. Find the pulse rate, the CPU load of a 12 µs ISR, the number of counts in a 50 ms window, the motor speed represented by one count, and the output shaft speed.

View solution

CHALLENGE 3: LEVEL SHIFTING, SAFE MOTORS AND CHOOSING A BOARD

(a) An HC-SR04's 5 V echo must reach a Raspberry Pi pin. With 1.8 kΩ and 3.3 kΩ, find the pin voltage, the divider current, the source impedance and the rise time with 30 pF. Why can this divider not be used to carry a 3.3 V signal up to 5 V? (b) The 6 V motor M1 (stall current 1.09 A, running 0.25 A) has a 0.22 Ω sense resistor in its ground return. Find the voltage and the Pico ADC count at the running current, at the stall current and at the 0.8 A trip point, the current represented by one count, and the power in the resistor at stall. (c) The program of Circuit 2 uses DEAD = 18, STEP = 2 and a 20 ms loop: find the duty for a knob reading of 0.5, the average voltage on a 4.8 V pack, and the time for the duty to ramp from 0 to 100 %. (d) For each job, choose a board (Uno, Pico, ESP32 or Raspberry Pi) and give one reason: (i) a line-following robot reading five 5 V analogue sensors; (ii) a robot driven by a phone over Wi-Fi; (iii) a robot identifying objects with a camera; (iv) a motor controller at 20 kHz with a TB6612 and a battery monitor.

View solution

QUICK REFERENCE

- **Pins:** a pin is a small voltage source and a high-impedance sensor; budget the current per pin and in total; switch anything above a few milliamps with a transistor or driver; LED R = $(V_{pin} - V_{LED})/I$
- **Inputs:** never leave one floating; pull-up R_p 4.7 to 10 kΩ (external) or about 50 kΩ (internal); rise time about 2.2 R_pC; debounce in software or hardware
- **ADC:** LSB = $V_{ref}/2^N$; code = floor($V_{in}/V_{ref} \times 2^N$); error ±½ LSB; Uno 4.88 mV, Pico 0.806 mV; a smaller reference gives finer steps for a small signal; source impedance 10 kΩ or less and a 100 nF at the pin
- **Divider into an ADC:** k = R2/(R1 + R2), k × V_{max} at most V_{ref}; V_{bat} = code × LSB / k; TMP36 = 0.5 V + 10 mV/°C
- **PWM:** V_{avg} = D × V; steps = f_{clk}/f_{PWM}, bits = floor($\log_2$ steps); 20 kHz is silent; RC filter ripple V D(1 − D)/(R C f); dead zone about 18 %; ramp the duty
- **Interrupts:** for pulses you must not miss; ISR short, shared variable `volatile`, copy with interrupts off on an 8-bit chip; load = rate × ISR time; speed = counts × 60 / (pulses per rev × window)
- **Level shifting:** 5 V to 3.3 V by a divider (1 k / 2 k gives 3.33 V); 3.3 V to 5 V usually fine into an Uno (V_{IH} about 3.0 V) or use a MOSFET module; never 5 V on a 3.3 V pin
- **Safe motors:** 100 kΩ pull-downs on driver inputs (default off); separate supply and common ground; ramp from zero; watchdog; low-battery cut-off through a divider; current sense (0.22 Ω gives 0.22 V/A); fuse and a hard switch
- **Boards:** Uno for 5 V sensors and learning; Pico for 3.3 V work and motor PWM; ESP32 for wireless; Raspberry Pi for software-heavy jobs (no ADC, loose timing)

Next, Basic Electronics 11 compares the Arduino, Raspberry Pi Pico, ESP32 and Raspberry Pi and others side by side (logic level, speed, I/O, power, price, wireless, ease of use), gives a table recommending the best board for each robot job, shows how to read a datasheet, covers test equipment and soldering, and wraps up the course with a recap and the path on to Robotics Fundamentals and Programming for Robotics.

Basic Electronics

Course 1 · Chapter 11 · Choosing a Board and Going Further

Basic Electronics 10 used four boards, each for one circuit, and named the reason each time. This last chapter turns that habit into a method. You will compare the boards on the same axes (logic level, speed, memory, I/O, power, wireless, price, ease of use), put numbers on the one resource every mobile robot runs out of (battery energy), match a board to each kind of robot job, learn to read a datasheet and to *derate* a part for heat, set up a small bench of test equipment, and learn to solder a joint that will last. The chapter ends with a recap of the whole course and the path on to the robotics courses.

All prices, clock speeds, memory sizes and currents below are **typical or approximate figures from memory**. They change with the board revision, the maker, the shop and the year, so treat them as a guide to the *order of magnitude* and check the maker's page or the datasheet before you buy or design around a number. The logic levels, the ratios and the formulas do not change.

1. The boards side by side

The table of Basic Electronics 10 compared pin current, ADC, PWM, interrupts and pull-ups. This table adds the other axes that decide a purchase, and three more boards that you will meet. Read it as a map, not a league table: no board wins every row.

BOARD	CHIP AND SPEED (TYPICAL)	LOGIC	MEMORY (TYPICAL)	WIRELESS	PRICE (APPROX.)	EASE OF USE
Arduino Uno (R3)	ATmega328P, 8-bit, 16 MHz	5 V	2 KB RAM, 32 KB flash	None	Official about 20 to 25 pounds; clones about 6 to 10	Easiest: huge tutorial base, forgiving, simulated in Tinkercad Circuits
Arduino Nano (same chip)	ATmega328P, 16 MHz	5 V	2 KB RAM, 32 KB flash	None	About 4 to 20 pounds (clone to official)	As the Uno, in a breadboard-sized package

BOARD	CHIP AND SPEED (TYPICAL)	LOGIC	MEMORY (TYPICAL)	WIRELESS	PRICE (APPROX.)	EASE OF USE
Raspberry Pi Pico	RP2040, dual-core Cortex-M0+, up to 133 MHz	3.3 V	264 KB RAM, 2 MB flash (board)	None (the Pico W adds Wi-Fi and Bluetooth)	About 4 pounds (Pico W about 6)	Easy: MicroPython or C, drag-and-drop programming over USB
ESP32 dev board	Dual-core Xtensa, up to 240 MHz (other ESP32 variants differ)	3.3 V	About 520 KB RAM, 4 MB flash (board)	Wi-Fi and Bluetooth built in	About 5 to 10 pounds	Moderate: very capable, but pin quirks (input-only pins, boot pins, ADC2 and Wi-Fi) catch beginners
Teensy 4.x	Cortex-M7, about 600 MHz	3.3 V (pins are not 5 V tolerant)	1 MB RAM, 2 to 8 MB flash	None on the board	About 25 to 35 pounds	Moderate: Arduino-style software, very fast, for jobs like a balancing robot or audio
Raspberry Pi 4 or 5	Quad-core Cortex-A72 (Pi 4, about 1.5 GHz); faster on the Pi 5	3.3 V, not 5 V tolerant	1 to 8 GB RAM; memory card for storage	Wi-Fi, Bluetooth, Ethernet	About 35 to 75 pounds, more for larger memory	Moderate: a full Linux computer, a keyboard and screen help, loose timing
Raspberry Pi Zero 2 W	Quad-core Cortex-A53, about 1 GHz	3.3 V	512 MB RAM	Wi-Fi, Bluetooth	About 15 pounds	As the Pi, smaller and cheaper, less memory

Reading the table: three ideas that save money and time

- **Clock speed is not the point.** A 16 MHz Uno reads a pin and turns on a motor driver with microsecond-level timing that a 1.5 GHz Raspberry Pi running an operating system cannot promise. A microcontroller runs *one* program with no operating system, so its timing is predictable; a Pi runs many programs and its timing is not. Speed matters for vision, maths and networking; predictability matters for motors and encoders (Basic Electronics 10).

- **Logic level decides what you can connect directly.** A 5 V board matches 5 V sensors and 5 V logic ICs; a 3.3 V board matches the TB6612 and DRV8833 motor drivers and most modern sensors, and needs a divider or a level shifter for 5 V signals (Basic Electronics 8 and Basic Electronics 10).
- **The cheapest board is rarely the cheapest project.** A 5-pound board that needs three extra modules, a level shifter and an hour of debugging costs more than a 10-pound board that has the radio built in. Count the whole bill of parts and your own time.

2. Power: a budget per board and a battery runtime

A robot dies when its battery does, and the battery that matters is usually the one that powers the *logic* for hours while the motors only run in bursts. The method is the same as the current budget of Basic Electronics 9: list every load, add the currents, check the peak against what the supply can deliver, then convert to energy and time.

```
Energy in a battery:      E (Wh) = V(nominal) x C (Ah)          Usable energy:    E(usa
Power drawn from the battery by a load behind a converter:      P(batt) = P(load) / eff
       a linear regulator (Basic Electronics 9) has efficiency = V(out) / V(in)  (and i
Runtime:      t = E(usable) / P(total)

Typical board-only power (approximate, idle to light work, at the 5 V supply, the board
       Board                      current at 5 V       power       Notes
       Arduino Uno                about 45 mA          0.225 W     the power LED, the r
       Raspberry Pi Pico          about 25 mA          0.125 W     sleep modes can lowe
       ESP32 dev board            about 100 mA         0.50 W      Wi-Fi active, averag
       Raspberry Pi 4             about 600 mA         3.0 W       idle; heavy load, a

Example 1, how long does each board alone run from a 10 000 mAh power bank (3.7 V cell,
       E = 3.7 V x 10 Ah = 37 Wh        E(usable) = 37 x 0.85 = 31.45 Wh
       Uno:          31.45 / 0.225 = 139.8 h   (about 5.8 days)
       Pico:         31.45 / 0.125 = 251.6 h   (about 10.5 days)
       ESP32 (Wi-Fi): 31.45 / 0.50  =  62.9 h   (about 2.6 days)
       Raspberry Pi 4: 31.45 / 3.0  =  10.5 h
       The factor of 300 between a Pico and a Pi 4 is why a camera robot carries a big
```

A complete power budget for one robot

```
A phone-driven robot: an ESP32, two ultrasonic sensors, three IR line sensors and one s

  Load on the 5 V rail          average      peak
  ESP32 dev board (Wi-Fi on)    120 mA       250 mA
  2 x HC-SR04 (15 mA each)       30 mA        30 mA
  3 x IR line sensor (20 mA)     60 mA        60 mA
  1 x servo (SG90 class)        100 mA       360 mA     (the peak is the stall-ish m
  Total                         310 mA       700 mA

  The buck converter must supply at least the peak, with margin:  700 mA is 70 % of a 1

  Battery energy:   E = 7.4 x 2.2 = 16.28 Wh       E(usable) = 16.28 x 0.8 = 13.02 Wh
  Logic power:      P(load) = 5 V x 0.310 A = 1.55 W       P(batt) = 1.55 / 0.9 = 1.72
  Motors:           two motors averaging 0.3 A each on the 7.4 V pack:    P = 0.6 A x 7
  Total:            P = 1.722 + 4.44 = 6.162 W

  Runtime:          t = 13.02 / 6.162 = 2.11 h   (about 127 minutes)

  What the budget tells you:  the motors are 72 % of the drain (4.44 / 6.162), so a bet
  The same robot with a linear regulator instead of the buck converter:  the battery su
  Total 6.734 W gives 13.02 / 6.734 = 1.93 h, 18 minutes less, plus a warm regulator.
```

BUDGET THE PEAK AND THE AVERAGE SEPARATELY

The *average* sets the battery life. The *peak* sets the size of the regulator, the wire, the fuse and the capacitor, and decides whether the board brown-outs when a servo starts (Basic Electronics 9). A design that fits on the average but not on the peak is the usual cause of a robot that resets whenever it moves.

3. The best board for each robot job

Basic Electronics 10 gave a short table of four jobs. This one widens it. Each row names a board, the reason and the one thing to watch; “beside” means that a second board handles the part the first is bad at, which is how many robots are built.

ROBOT JOB	BEST BOARD	WHY, AND THE CATCH
A first robot: line follower or sumo with a few 5 V analogue sensors, no wireless	Arduino Uno or Nano	5 V logic and ADC match 5 V sensors; the most tutorials; forgiving of mistakes. Catch: 2 KB of RAM is small, and 8-bit maths is slow for anything fancy.
A precise driving robot: two motors with encoders, a PID loop, 20 kHz PWM, a battery monitor	Raspberry Pi Pico	PWM on any pin at 20 kHz, interrupts on every pin, 3.3 V drivers, a 12-bit ADC, low cost. Catch: no wireless (use a Pico W if you need it), few ADC pins.
A robot controlled from a phone or a web page; a Wi-Fi sensor node	ESP32	Radio and two cores on the chip, many pins. Catch: input-only pins, ADC2 off while Wi-Fi runs, noisy ADC, higher power than a Pico.
A camera robot: following a face, a colour or a line from images; speech; a web dashboard	Raspberry Pi 4 or 5, with a Pico or an Arduino beside it for the motors	Needs an operating system and memory for image libraries; the Pi cannot give hard timing or an ADC, so the small board drives the motors and reads the encoders and sensors. Catch: 3 to 6 W, a 3 A supply, a long boot time, and shutting it down properly.
A small, light robot that needs vision but little else	Raspberry Pi Zero 2 W	The same software at a fraction of the weight and power (about 1 W). Catch: 512 MB of RAM limits the vision work.
A self-balancing robot (a tilt sensor read and acted on 200 times a second)	Pico, ESP32 or Teensy	A fast, predictable loop and floating-point maths. Catch: an 8-bit Uno can do a simple version but runs out of speed and memory first.
An arm with six servos	Pico, or an Arduino Mega or Uno with a servo driver board	Many timed pulses at once: a Pico has 16 PWM channels; a PCA9685 servo-driver module (16 channels over I2C) adds servos to any board. Catch: servo power is its own supply, never the board (Basic Electronics 9).
A sensor logger or beacon that must run for weeks on batteries	Pico (or an ESP32 in deep sleep)	Low power, sleep modes. Catch: the board's own regulator and LEDs, not the chip, often dominate the sleep current, so a bare chip or a low-power board beats a general dev board.
A robot with a spinning range sensor, mapping or ROS software	Raspberry Pi 4 or 5, with a microcontroller for the motors	The robotics software and the network drivers run on Linux. Catch: the loose timing again, so the real-time part goes to the small board.

A quick decision procedure

1. **What must it do that is hard?** If the answer is vision, speech or a screen, a Raspberry Pi goes in; if it is precise timing or analogue input, a microcontroller goes in; if it is both, use both.
2. **What logic level are the parts?** 5 V parts favour an Uno; 3.3 V parts favour a Pico or ESP32; a mixture means dividers or level shifters (Basic Electronics 8).
3. **Does it need a radio?** Yes, and nothing else special: ESP32 or Pico W. No: avoid paying for one.
4. **Do the pin and current budgets fit?** Count analogue inputs, PWM outputs, interrupt pins and the total current (Basic Electronics 10), and the power budget of section 2.
5. **How experienced are you?** A board with plenty of examples gets you running sooner; first projects should not need a pin-quirks chart.

4. Reading a datasheet

Every part you will ever use has a datasheet, and every datasheet has the same few parts. The skill is knowing which table to trust for which question. Download the datasheet from the maker (the part number is printed on the part); a distributor's copy is usually the same document.

- **Features and description (page 1).** What the part is for and its headline figures. A summary only; never design from it.
- **Absolute maximum ratings.** The limits that *destroy* the part: supply voltage, pin voltage, current, temperature. These are **not** operating conditions. You design to stay well inside them (Basic Electronics 10 used "supply plus 0.3 V" for a pin).
- **Recommended operating conditions.** The ranges within which the part is guaranteed to meet its other specifications: this is the table to design to.
- **Electrical characteristics.** Each parameter has *minimum*, *typical* and *maximum* columns and, importantly, *test conditions*. The typical value is what a part usually does at 25 degrees C; the minimum and maximum are the guarantees. Design so the circuit works at the **worst case** in the direction that hurts: the largest value if too much is bad, the smallest if too little is bad.
- **Graphs.** Curves such as forward voltage against current, or resistance against temperature, show what the table does not (typically measured at one temperature).
- **Package and pinout.** Which leg is which, and the size and spacing. Check this before wiring, especially for the TMP36 of Basic Electronics 10, whose reversed pins make it hot in seconds.

- **Application circuit.** The maker's recommended wiring, including the capacitors and resistors the part needs.

Worked example: sizing a logic-level MOSFET from its datasheet

```
An illustrative logic-level N-channel MOSFET (typical of the kind in Basic Electronics (
      Gate threshold V(GS)(th):                    1.0 min,  1.7 typ,  2.5 max V     (
      On-resistance R(DS)(on), V(GS) = 4.5 V:       0.035 typ, 0.044 max ohm
      On-resistance R(DS)(on), V(GS) = 2.5 V:       0.048 typ, 0.060 max ohm
      Continuous drain current I(D), 25 C case:     4 A            Absolute maximum V(

  Question: can a 3.3 V Pico pin drive it to switch the 6 V motor M1 (running 0.25 A, s
      The pin gives 3.3 V, above the 2.5 V test point, so use the 2.5 V row and its WO
      Voltage lost in the switch at stall:    V = I R = 1.09 x 0.060 = 65 mV      (1.1
      Power in the switch at stall:           P = I^2 R = 1.09^2 x 0.060 = 0.0713 W
      The R(DS)(on) rises when hot (roughly 1.5 times at 100 C, a typical order of magn
      The drain-source voltage rating of 30 V is 5 times the 6 V supply, and the motor
      Verdict: suitable. Had the only R(DS)(on) given been "at V(GS) = 10 V", the data
```

Derating: the datasheet number is not the number to use

Ratings are given at a stated temperature, usually 25 degrees C, and fall as the part gets hotter. **Derating** is the reduction. Datasheets give a derating curve or a rule such as "derate linearly from 70 degrees C to zero at 155 degrees C"; as a formula, with the power rating P_{rated} given at T_{rated} and falling to zero at T_{max}:

```
Linear derating:      P(allowed at T) = P(rated) x (T(max) - T) / (T(max) - T(rated))
Thermal resistance:    T(junction) = T(ambient) + R(theta,JA) x P          so      P(al

  Example 1, a 0.25 W resistor rated at 70 C, zero at 155 C (a common curve), running i
      P(allowed) = 0.25 x (155 - 100) / (155 - 70) = 0.25 x 55 / 85 = 0.162 W      (65
      A resistor dissipating 0.2 W in that box would be overloaded even though 0.2 W i

  Example 2, a small plastic-cased transistor, R(theta,JA) = 200 C/W (typical of the sm
      P(allowed) = (150 - 40) / 200 = 0.55 W

  Example 3, a 7805-style linear regulator (Basic Electronics 9) dropping 12 V to 5 V a
      Dissipation:  P = (12 - 5) x 0.3 = 2.1 W          Temperature rise = 65 x 2.1 =
      The most it can dissipate:  P(max) = (125 - 40) / 65 = 1.31 W          so the la
```

```
        Cures:  add a heatsink (lowers R(theta)), lower the input voltage, or use a buck
  A good design habit:  run a part at no more than half its derated rating.
```

5. Test equipment: what to buy first

You can build a long way with three instruments. A fourth item, the breadboard and its wires, is cheap and was covered in Basic Electronics 2.

- **The digital multimeter (DMM).** The first purchase (Basic Electronics 2 showed how to use it). A cheap one is adequate for continuity, voltage and resistance, but check its **accuracy specification**, which is written as *±(a % of reading + b digits)*. The percentage scales with what you measure; the digits are a fixed amount at the meter's resolution on that range, and matter most for small readings.
- **A bench power supply with a current limit.** Adjustable voltage, and a current limit you set *before* connecting the circuit. A first power-up with the limit at, say, 100 mA turns a wiring mistake into a lit LED on the supply instead of a burnt part. It is the second purchase; until then, batteries and a fuse (Basic Electronics 9) do the job.
- **An oscilloscope or a logic analyser.** A scope draws voltage against time and is the only instrument that shows a PWM waveform, ripple, ringing or a bounce. Two numbers matter: *bandwidth* and *sample rate*. A logic analyser shows only high and low, many channels at once, and is the tool for serial, I2C and SPI traffic; cheap USB ones with 8 channels cost about 10 pounds, and a small USB scope with 20 MHz bandwidth about 50 to 70 pounds (approximate figures).

```
 Multimeter accuracy:      error = (a / 100) x reading + b x (resolution of the range)
   A 3 1/2-digit meter on its 20 V range displays 0.01 V steps. Spec: +/-(0.5 % of readi
        error = 0.005 x 5.00 + 2 x 0.01 = 0.025 + 0.020 = 0.045 V       the true value i
   A small reading is hit harder by the digits:  0.50 V on the same range:  0.005 x 0.50
   Practical consequences:  a 5 % resistor measures to within a few per cent on a good r

 Oscilloscope bandwidth (the frequency at which the displayed amplitude has fallen to 70
   rise time of the scope itself:    t(r, scope) = 0.35 / BW             20 MHz scope: 0
   a signal edge appears slower:     t(displayed) = sqrt( t(signal)^2 + t(scope)^2 )
   For a square wave, you need a bandwidth of at least 5 times the fundamental to see a
   Use the x10 probe setting for most work: it presents about 10 Mohm and a few pF to th
   Logic analyser:  sample at 4 times the signal frequency at the very least, 10 times i
```

6. Soldering: making a joint that lasts

A breadboard is for testing; a robot that moves needs soldered connections, because a push-fit lead shakes loose. Soldering melts an alloy (solder) so that it wets both the part's lead and the board's pad and, on cooling, forms a metal joint. The flux inside the solder wire cleans the metal so that the solder can wet it.

- **Equipment:** a temperature-controlled iron or station (about 25 to 60 W, with a fine conical or small chisel tip), solder wire with a flux core (0.6 to 0.8 mm for electronics), a damp sponge or brass wool to clean the tip, desoldering braid, a stand, and ventilation (a small fan or fume extractor to draw the flux smoke away from your face).
- **Temperature:** the usual tin-lead solder (Sn63/Pb37) melts at about 183 degrees C; lead-free solder (for example the SAC305 alloy) melts at about 217 to 220 degrees C. A tip set to roughly 320 to 350 degrees C works for both (typical figures). Too cool, and the joint is dull and lumpy; too hot, and pads lift and parts are damaged.
- **Technique:** tin the tip with a little solder; touch the tip so that it heats *both* the lead and the pad for one to two seconds; feed the solder wire into the junction of lead, pad and tip, **not** onto the tip; let a small amount flow; remove the solder first, then the iron; hold the part still until the joint is solid (a second or two). The whole joint takes about two to four seconds; a longer time risks damaging the part or the board.
- **A good joint** is shiny (a lead-free joint is a little duller), shaped like a small smooth cone or volcano that follows the lead up from the pad, with the lead visible inside it. A **bad joint** is a dull, grainy ball that sits on top (a cold joint: poor contact, intermittent), or a blob that bridges to the next pad (a short). Reheat it with a touch of fresh solder; remove surplus with braid.
- **Heat-sensitive parts:** the first parts to fit are the low-profile ones (resistors, diodes), and the last the tall ones (headers, capacitors). Use sockets for ICs, and respect the polarity of diodes, LEDs and electrolytic capacitors (Basic Electronics 5).

SOLDERING SAFETY

The tip is at 300 to 350 degrees C: always return the iron to its stand, never lay it on the bench, and never grab a falling iron. Wear safety glasses (solder spits and wire clippings fly), solder in a ventilated place, and wash your hands after handling tin-lead solder. Unplug the iron when you finish. If you burn yourself, cool the burn under running cold water for at least ten minutes.

7. Going further: the next two courses

Basic Electronics ends where a robot's electronics begins. Two further courses on this site are **planned** (neither has been written yet) and are written to follow this one:

- **Robotics Fundamentals** (planned): the mechanical and control ideas of a robot, which use the parts of this course as building blocks: sensors and what they measure, actuators and drive systems, kinematics, feedback and control, and how the electronics of a robot is laid out and tested.
- **Programming for Robotics** (planned): the software side, which builds on the boards of Basic Electronics 10 and this chapter: reading sensors and driving motors in code, loops and timing, state machines and simple control programs.

Until they appear, the best next step is the practical one: pick one of the board jobs in section 3, build it on a breadboard, then make it permanent with solder. Each project teaches the same lesson as this whole course: a number you calculated, a reading you measured and a limit you read from a datasheet should agree, and when they do not, the disagreement is where the learning is.

8. Course complete: a recap of Chapters 1 to 10

CHAPTER	TITLE	THE ONE THING TO KEEP
1	Electricity Basics	Voltage pushes, current flows, resistance opposes: V = I R and P = V I are in every calculation after this.
2	Circuits and Measurement	Read a schematic, use Kirchhoff's laws, wire a breadboard, measure with a multimeter (voltage across, current through).
3	Resistors and Dividers	Resistor values and power ratings; the divider $V_{out} = V_{in}$ R2 / (R1 + R2); pull-ups and pull-downs.
4	Capacitors and Inductors	RC time constants, filters, decoupling, and the back-EMF of an inductor that makes a switch-off spike.
5	Diodes and LEDs	One-way current, an LED always needs a resistor, flyback and polarity protection.
6	Transistors as Switches	A small current or voltage at a base or gate switches a larger load; choose a logic-level MOSFET for a 3.3 V pin.

CHAPTER	TITLE	THE ONE THING TO KEEP
7	Analogue Signals and Op-Amps	Analogue versus digital, ADC resolution, the op-amp as buffer, amplifier and comparator, and noise.
8	Digital Logic Basics	Logic levels and noise margins, gates, Schmitt triggers, counters, the 555 timer and level shifting.
9	Motors, Relays and Power	Motor model, PWM, H-bridges, servos, relays, batteries, regulators, and the current budget that keeps a robot from resetting.
10	Microcontrollers in Circuits	A board is a part with limits: pin current, ADC and PWM resolution, pull-ups, level shifting, and safe motor driving.
11	Choosing a Board and Going Further	Choose by logic level, timing, radio, power and price; budget the energy; read datasheets at the worst case and derate; test and solder.

9. Challenges

CHALLENGE 1: CHOOSE A BOARD FOR THREE ROBOTS

For each robot, choose a board (Uno, Pico, ESP32 or Raspberry Pi, with a second board if needed), give two reasons from this chapter, and estimate the logic runtime. (A) A sumo robot: four 5 V IR sensors on analogue pins (20 mA each), two motors, no wireless, built by a beginner; the board takes 45 mA, the pack is a 2S LiPo (7.4 V, 1000 mAh) feeding the Uno's own linear regulator. Find the regulator's dissipation and the runtime of the logic alone (80 % usable). (B) A rover driven from a phone: the ESP32 averages 120 mA, a servo 80 mA and two sensors 30 mA on a 5 V rail from a buck converter (90 %) on a 7.4 V, 2200 mAh pack. Find the rail current, the battery power for the logic and the logic runtime (80 % usable); and say why GPIO34 is a sensible pin for a battery-monitor divider. (C) A camera rover that follows a coloured ball: a Raspberry Pi 4 (3.0 W), a camera (0.25 W) and a Pico driving the motors (0.15 W), on a 10 000 mAh, 3.7 V power bank (85 % usable). Find the total power and the runtime; then repeat with a Raspberry Pi Zero 2 W (assume 0.8 W for the board under load) and say what you would lose.

View solution

CHALLENGE 2: READ A DATASHEET TABLE

A sample red LED datasheet (illustrative figures) gives: absolute maximum continuous forward current 30 mA; peak current 100 mA at 1 kHz and 10 % duty; power dissipation 75 mW at 25 degrees C, derated by 1.0 mW per degree C above 25 degrees C; forward voltage at 20 mA of 1.7 V minimum, 1.9

V typical and 2.2 V maximum. (a) The LED is driven from a 3.3 V Pico pin and the current must never exceed 10 mA, whatever the LED's forward voltage: find the minimum resistor and pick the next E12 value (100, 120, 150, 180, 220 ohm). (b) With that resistor, find the current for the minimum, typical and maximum forward voltages, and the power in the LED in the worst case. (c) At 70 degrees C ambient, find the allowed power, and the largest current if V_f is 1.9 V; at what temperature does the allowed power reach zero? (d) A pulsed 100 mA, 10 % duty at 1 kHz: find the average current and the pulse width, and say whether it is within the ratings. (e) Three such LEDs are driven from three Pico pins: is the Pico's total pin budget (about 50 mA) respected?

View solution

CHALLENGE 3: PLAN A BENCH TOOLKIT WITH A BUDGET

You have 120 pounds. Approximate prices: digital multimeter 20; temperature-controlled soldering station 28; solder wire 8; desoldering braid and flux pen 6; side cutters and wire strippers 10; safety glasses 3; two breadboards and a jumper kit 12; component assortment 12; tweezers 5; helping hands 10; fume extractor fan 14; USB logic analyser 10; silicone mat 8; bench power supply (30 V, 5 A) 55; 20 MHz USB oscilloscope 60. (a) Mark the first eight items as essentials; find their total and what is left; choose the extras that give the best value for health and safety and show the new total. (b) Plan the next two purchases (the supply and the scope) and, saving 15 pounds a month, say how many months it takes. (c) A multimeter is specified ±(0.8 % of reading + 3 digits) on a 2 V range (0.001 V steps): find the error for a reading of 1.000 V and of 0.100 V, as percentages of the reading. (d) The same meter has ±(1.0 % + 2 digits) on its 20 kΩ range (0.01 kΩ steps): find the range for a displayed 4.70 kΩ and check whether a 5 % resistor labelled 4.7 kΩ fits within it. (e) For the 20 MHz scope, find the scope's rise time, the displayed rise time of a 10 ns logic edge, and the highest square-wave fundamental that meets the 5-times rule.

View solution

QUICK REFERENCE

- **Choosing:** hard thing first (vision: Pi; timing or analogue: microcontroller; both: both); then logic level, radio, pin and current budgets, and experience; Uno 5 V and easy; Pico 3.3 V and precise; ESP32 wireless; Pi software-heavy
- **Energy:** E = V × C; usable = 0.8 E; P(batt) = P(load)/efficiency; t = E(usable)/P(total); linear regulator efficiency = V_{out}/V_{in}; budget the peak and the average separately
- **Datasheets:** design to Recommended Operating Conditions and the worst-case min/max column; Absolute Maximum is the limit that destroys; check the test conditions, the pinout and the application

circuit

- **Derating:** P(allowed) = P_{rated} (T_{max} − T)/(T_{max} − T_{rated}); $T_j = T_a + R_{\theta JA} P$; run a part at no more than half its derated rating
- **Test gear:** meter error = (a/100) × reading + b × resolution; scope t_r = 0.35/BW and bandwidth at least 5 × the fundamental; logic analyser at least 4 × the signal; set the supply's current limit first
- **Soldering:** tip about 320 to 350 °C; heat lead and pad together 1 to 2 s; feed the solder to the joint, then remove solder, then the iron; shiny cone is good, dull ball is cold; ventilation, glasses, stand
- **Next:** Robotics Fundamentals and Programming for Robotics (both planned)

That completes Basic Electronics. You can now read a schematic, calculate the part values of a circuit, check them with a meter or a simulator, choose a board for a robot and say why, and read a datasheet critically. Robotics Fundamentals and Programming for Robotics, both planned, will use all of it.