Daily Review

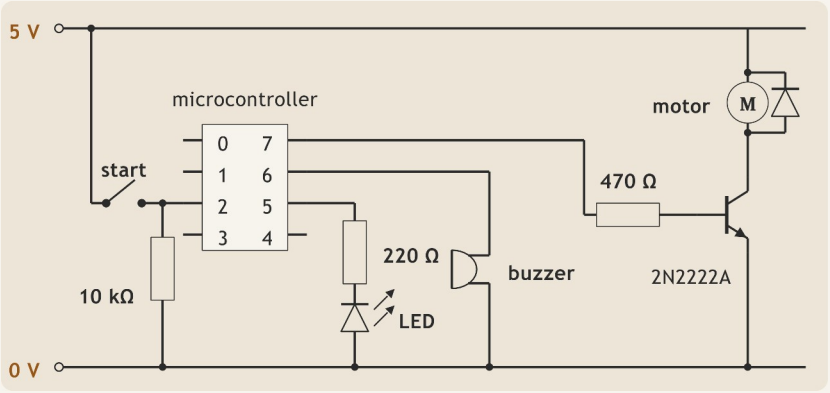


1. **State** the **column** that records **what you saw**.
2. **State** what you must do **after** a **change**.
3. **State** how many **things** you **change** at once.
4. **State** the **resistor** an **LED** needs on the Uno. (**value and unit**)
5. **State** what an **exam** answer **never** uses.

Assignment

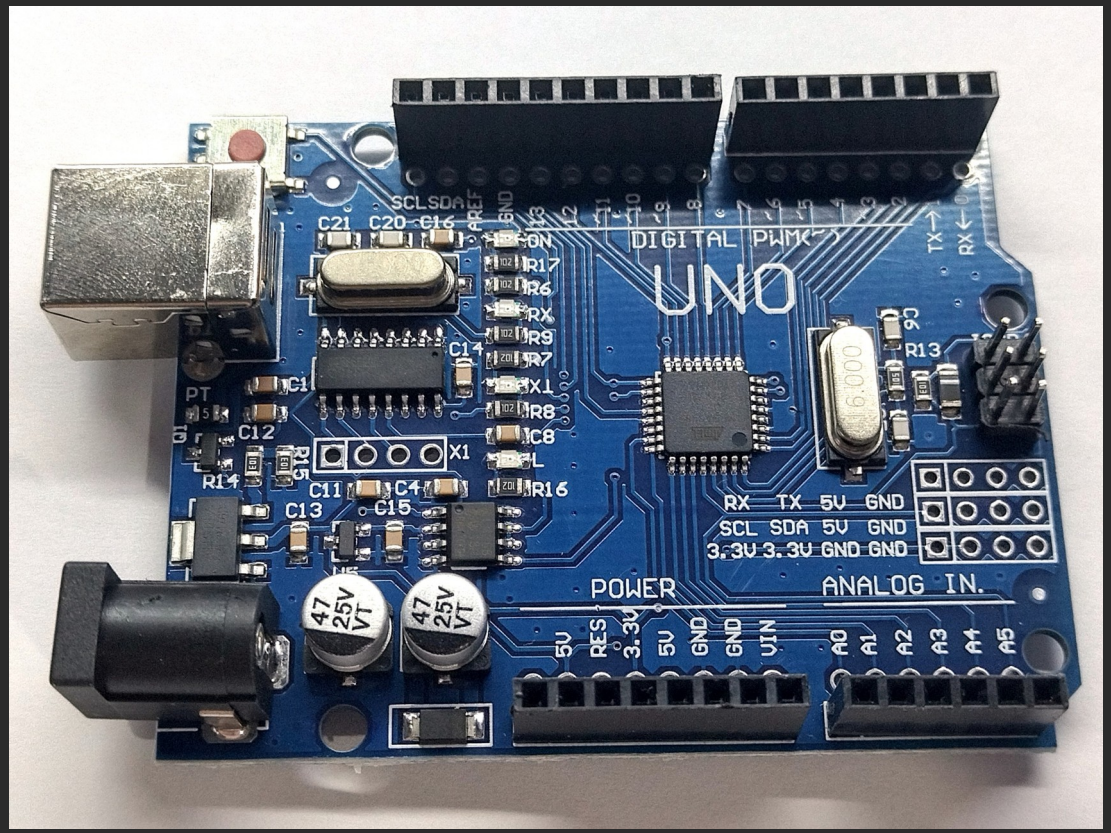
Practice

The engineer's design has mistakes.
Find them.



National 5 Engineering Science

The engineer's design has two mistakes. Why build it exactly as drawn?



PART 1

Assignment Practice

Build it as drawn. Test it. Fix it. Prove it.

By the end of this part…



1. I can **build** a design **exactly as drawn**
2. I can **tell** a **circuit fault** from a **flowchart fault**
3. I can **fill in** a **test table** and **prove** it with a **hard copy**

The Programmable Task (14 marks)

Simulate it as drawn

5 marks

Test table

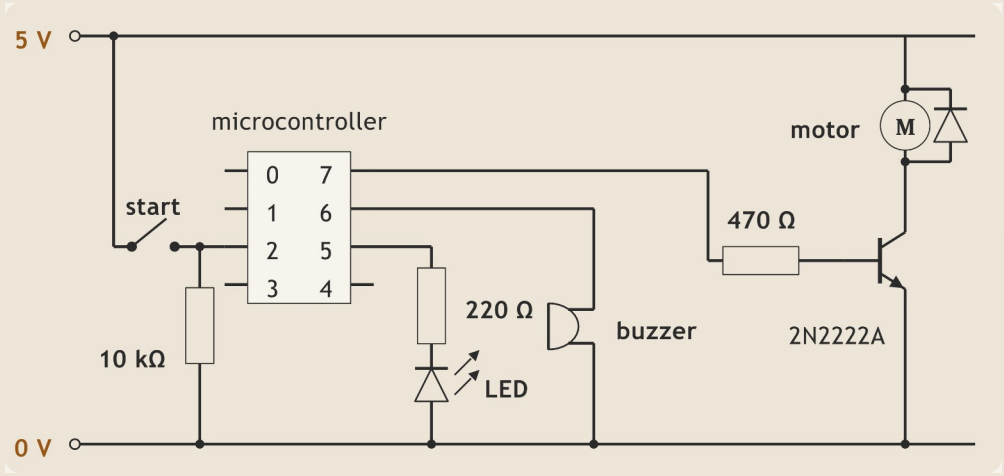
5 or 6 marks

Amended version

2 marks

Then one of

code (1) or an improvement (2)



Evidence: No Hard Copy, No Marks

Snip

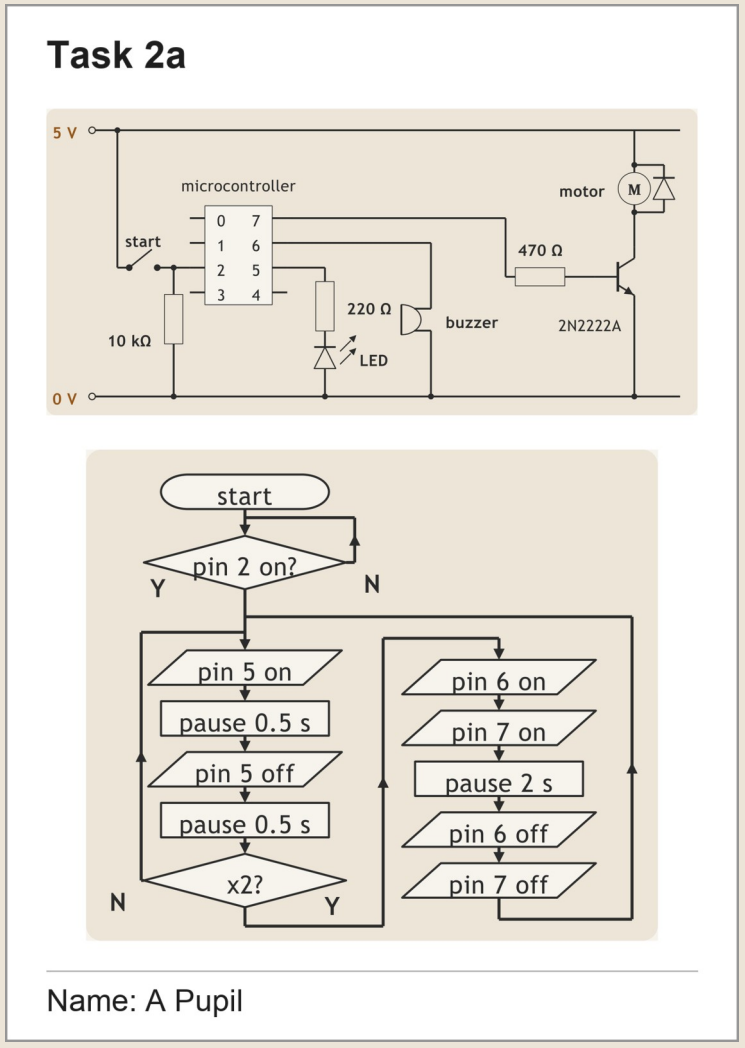
Windows + Shift + S

Label

task number at the top, name at the foot

Print

as built AND after amending



Worked Example: The Arcade Rumble

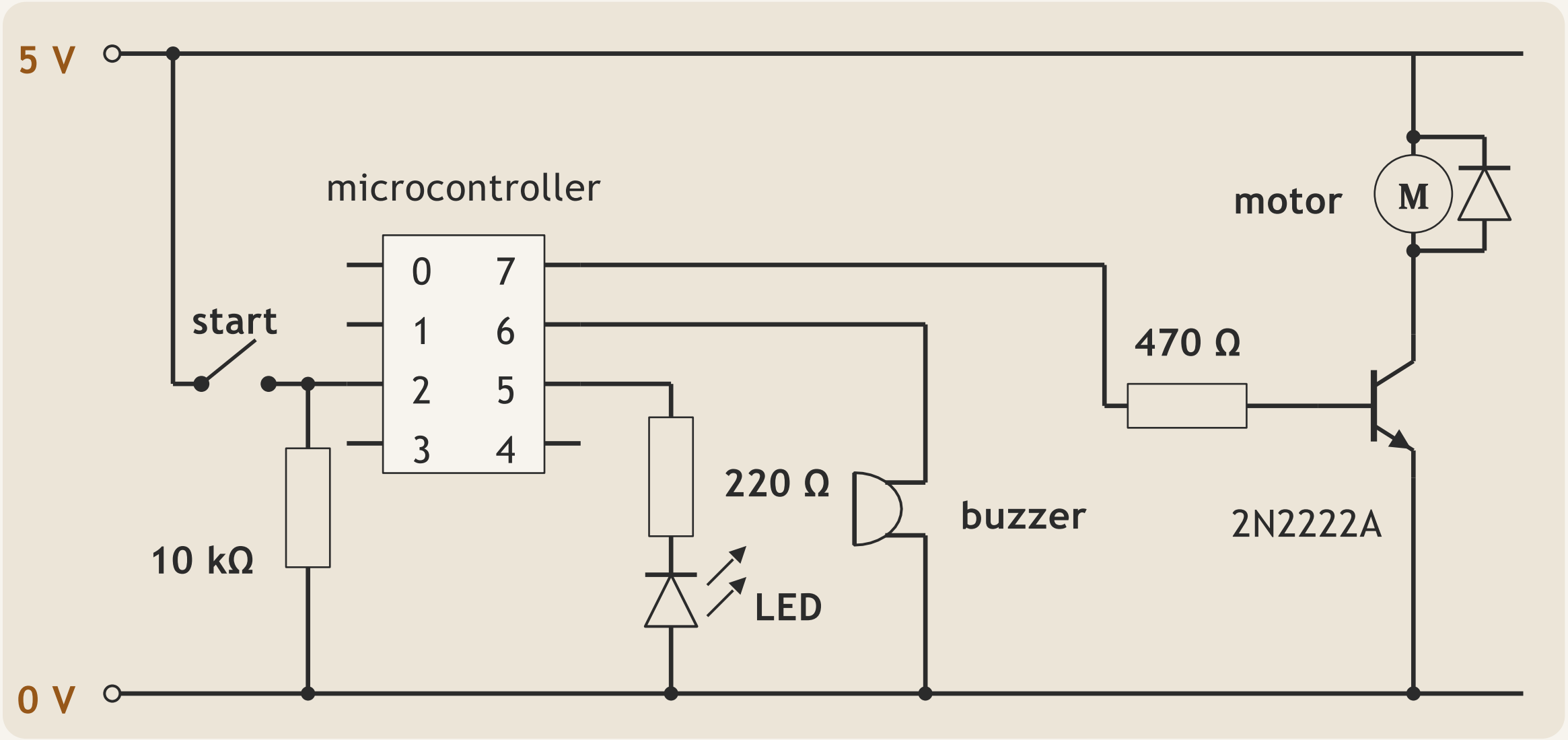
Input connection	Pin	Output connection
	7	rumble motor
	6	buzzer
	5	LED
start switch	2	

Simulate the engineer's design **exactly as shown**. (5)
Produce a **hard copy** of your simulation.

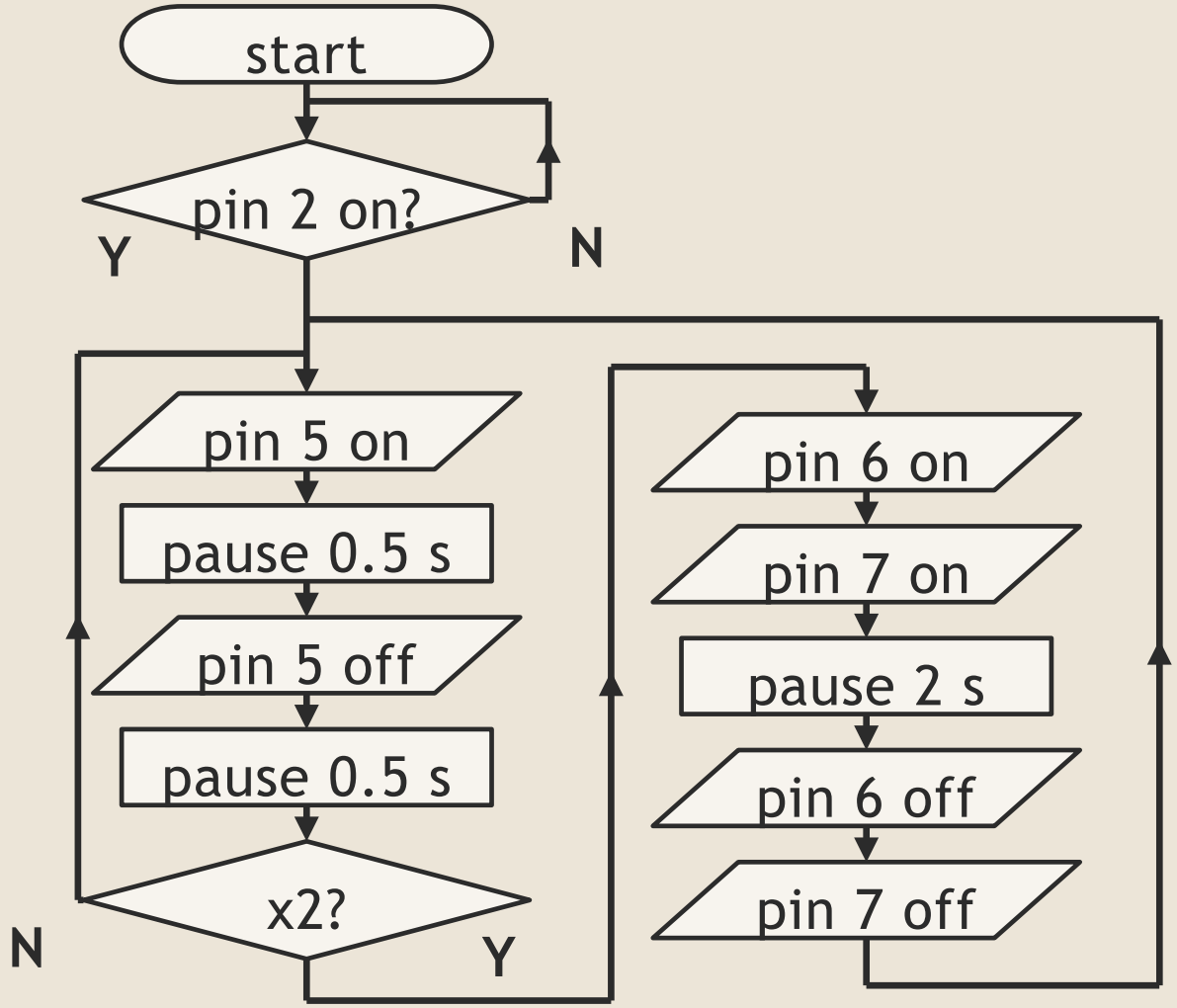
The game's rumble works like this:

- When start is pressed, the LED flashes 2 times: on 0.5 s, off 0.5 s.
- Then the buzzer and the rumble motor run for 2 s.
- They switch off. The game waits for start again.

The Engineer's Design: Circuit



The Engineer's Design: Flowchart



Build it

exactly as drawn

One fault

is in here too

Do not fix it

until a test finds it

Build It Exactly As Drawn

Pins

match the pin table

Parts

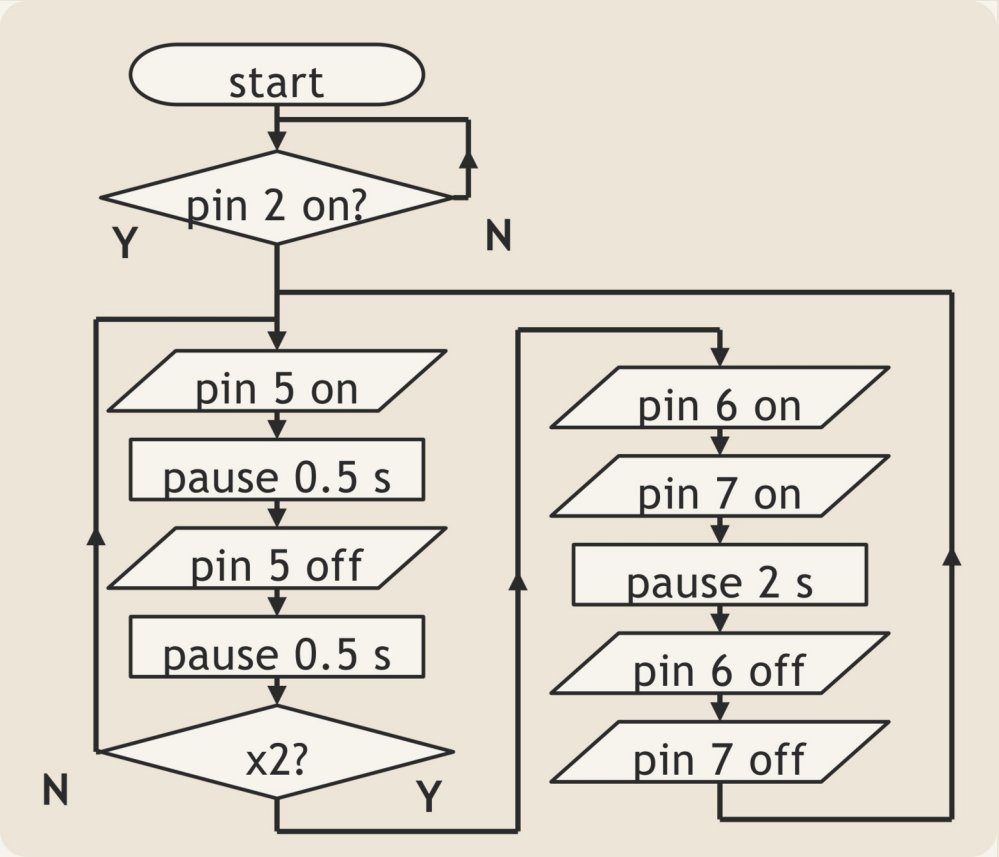
the right way round

Values

220 Ω, 470 Ω, 10 kΩ

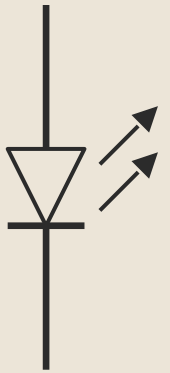
Loops

exactly as drawn, even if wrong



Which Way Round?

anode



cathode

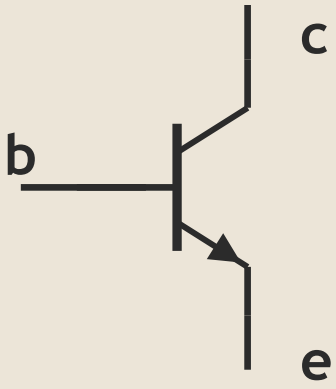
Bar = short leg, flat edge

cathode



anode

Bar = the grey band



Emitter = leg by the tab

Circuit Faults: What You See

Base resistor too big

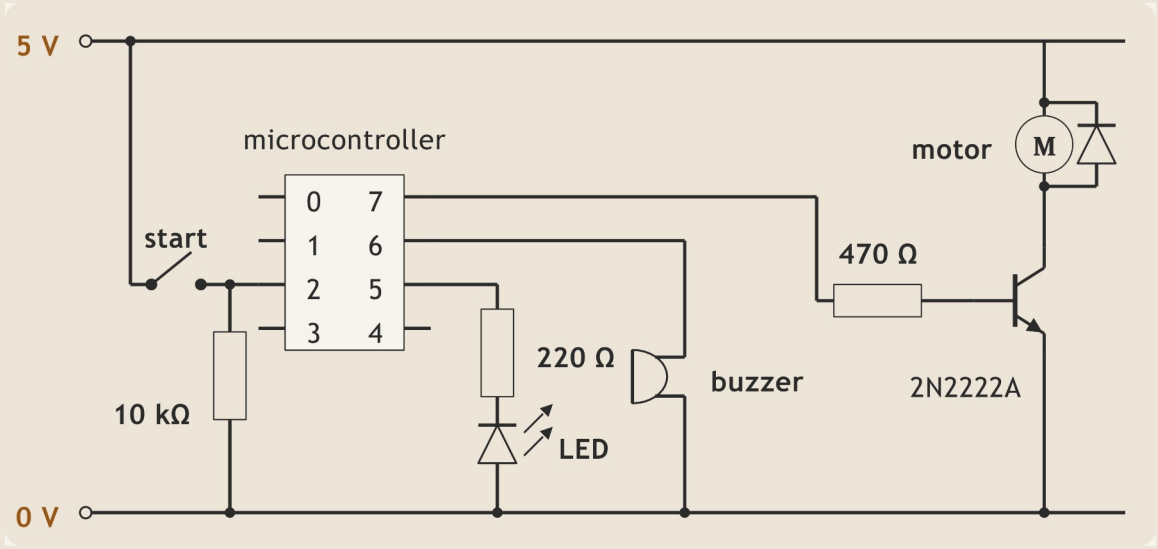
the motor never turns

LED backwards

the LED never lights

Diode backwards

the motor never turns



Flowchart Faults: What You See

Wrong loop count

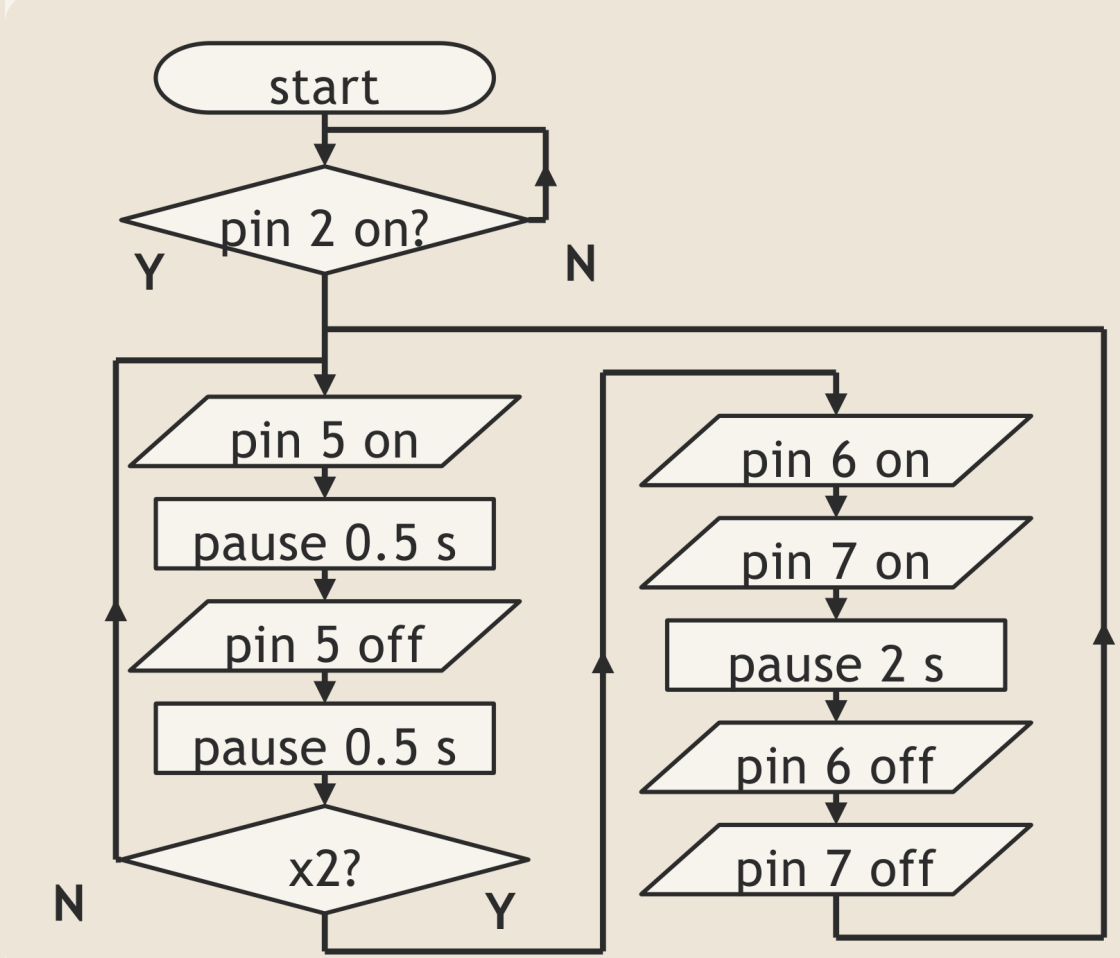
flashes the wrong number of times

Loop to the wrong box

repeats without waiting

A missing off

an output never stops



Why the Base Resistor Matters

470 Ω

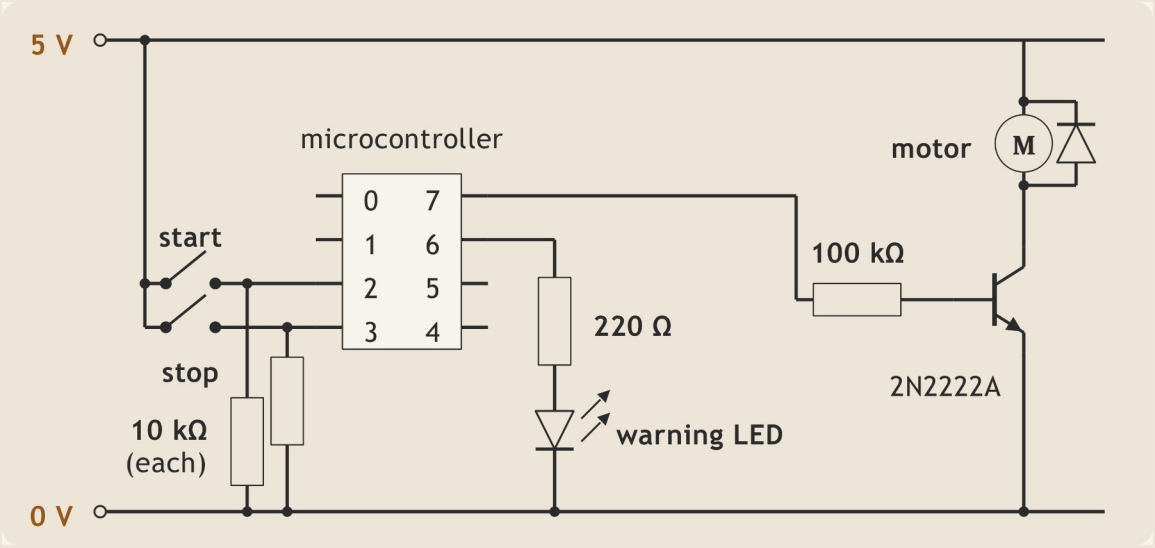
about 9 mA into the base: motor turns

100 kΩ

about 0.04 mA: motor never turns

So

reduce the base resistor



Circuit or Flowchart?

1. The LED **flashes**, but the **motor never turns**. **State** where the fault is.
2. The LED flashes **2 times**, not **3**. **State** where the fault is.
3. The **LED never lights**; the buzzer works. **State** where the fault is.

Circuit or Flowchart?

4 It **repeats** without **start** being pressed. **State** where the fault is.

5 The **motor** never **stops**. **State** one **likely fault**.

The Test Table



Column	What goes in it
Planned test	what you will do
Expected result	what the specification says
Initial result	what did happen, part by part
Planned amendment	the part, and the change
Re-test result	what happens now

Worked Example: Test 1

Test table	Test 1
Planned test	Press start.
Expected result	The LED flashes 2 times.
Initial result	The LED does not light.
Planned amendment	Rotate the LED 180°.
Re-test result	The LED flashes 2 times.

Worked Example: Test 2

Test table	Test 2
Planned test	Wait after the flashes.
Expected result	Buzzer and motor on for 2 s.
Initial result	Buzzer and motor on for 2 s.
Planned amendment	No amendment required.
Re-test result	No re-test required.

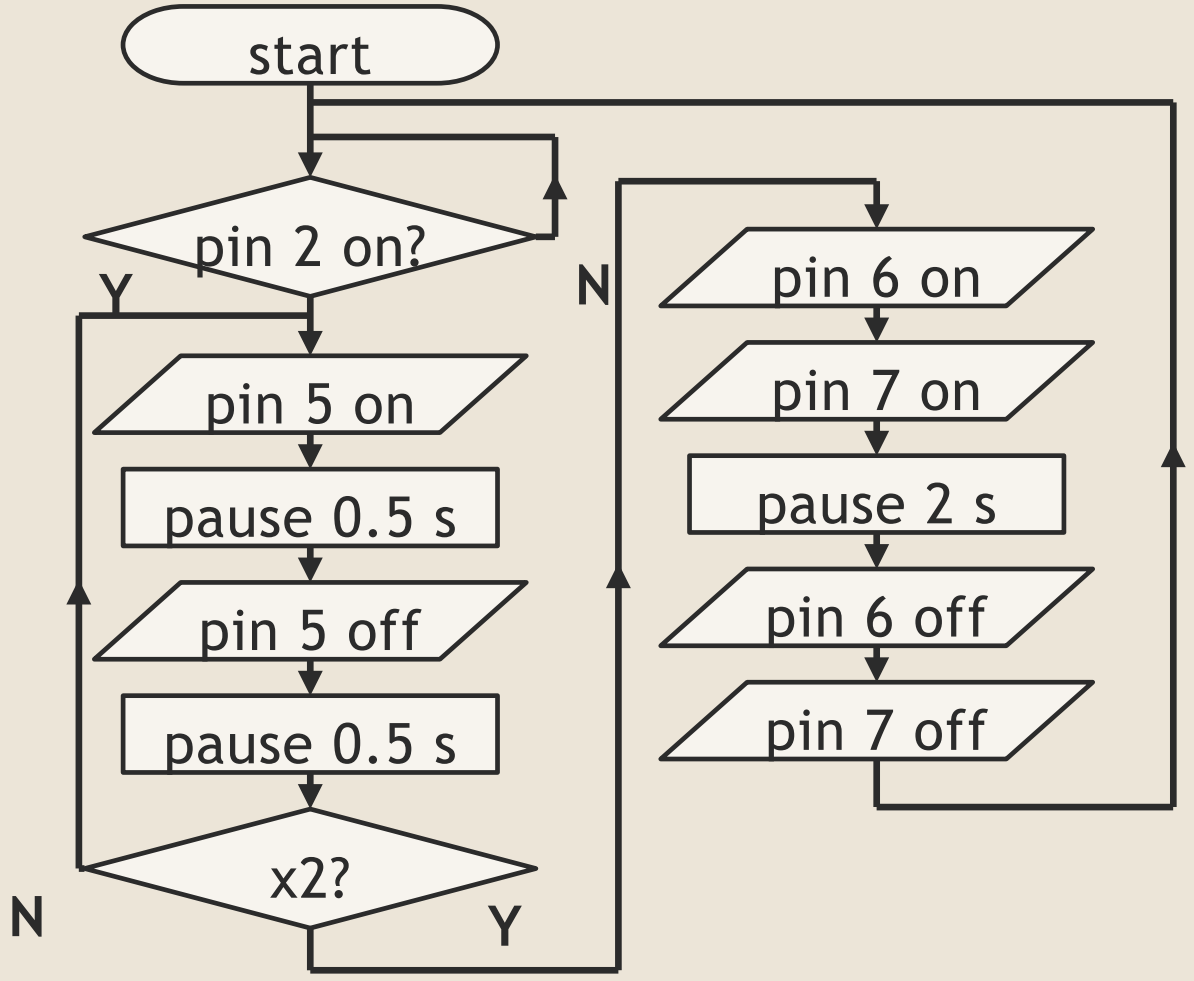
Worked Example: Test 3

Test table	Test 3
Planned test	Wait after the rumble.
Expected result	It waits for start.
Initial result	The LED flashes again.
Planned amendment	Loop back from pin 7 off to start.
Re-test result	It waits for start.

Say What You Saw

THE MISTAKE	THE CORRECT ANSWER
It didn't work.	The motor did not turn.
Change the resistor.	Reduce 100 kΩ to 470 Ω.
Fix the loop.	Loop back from pin 7 off to start.

The Amended Version: Hard Copy 2



Flowchart

pin 7 off loops back to start

Circuit

LED rotated 180°

Then

print it again, labelled

Specification Met?

Specification point	Met?	Justification
i LED flashes 2 times	Yes	Re-test 1: it flashed 2 times
ii Buzzer and motor 2 s	Yes	Test 2: on for 2 s
iii Waits for start	Yes	Re-test 3: it waited

Model Answer: An Improvement (2 marks)

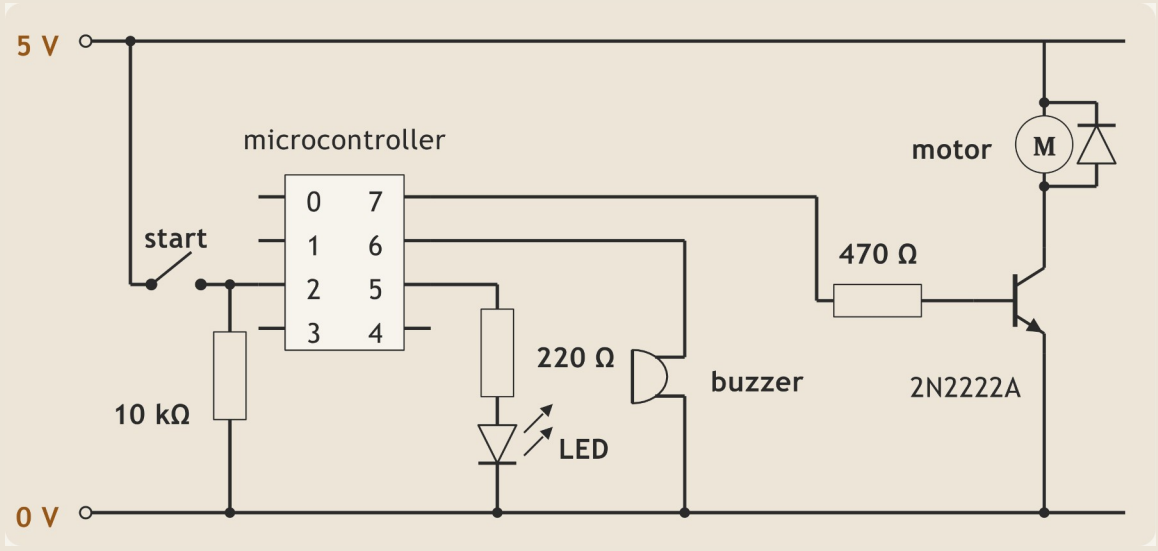
Suggest an **improvement** for the **player**. **Justify** it. (2)

Improvement (1)

A ready LED that lights while it waits

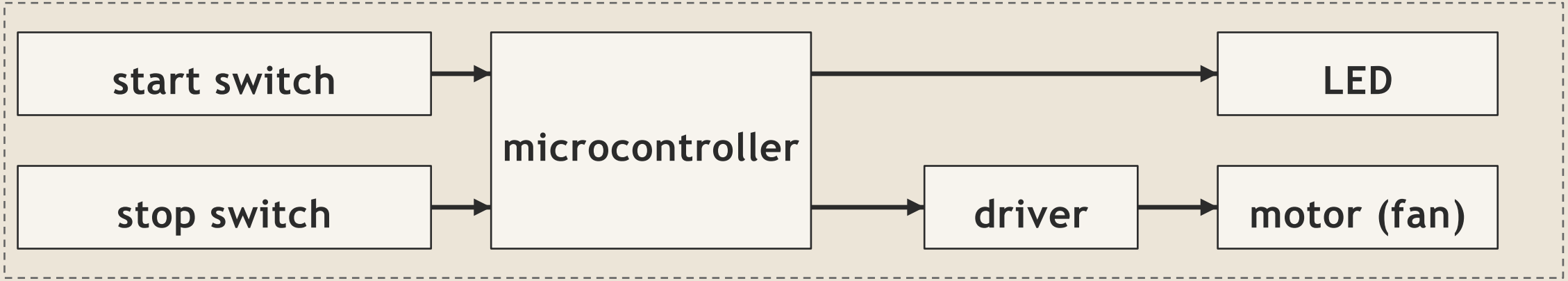
Justification (1)

so the player knows when to press start.



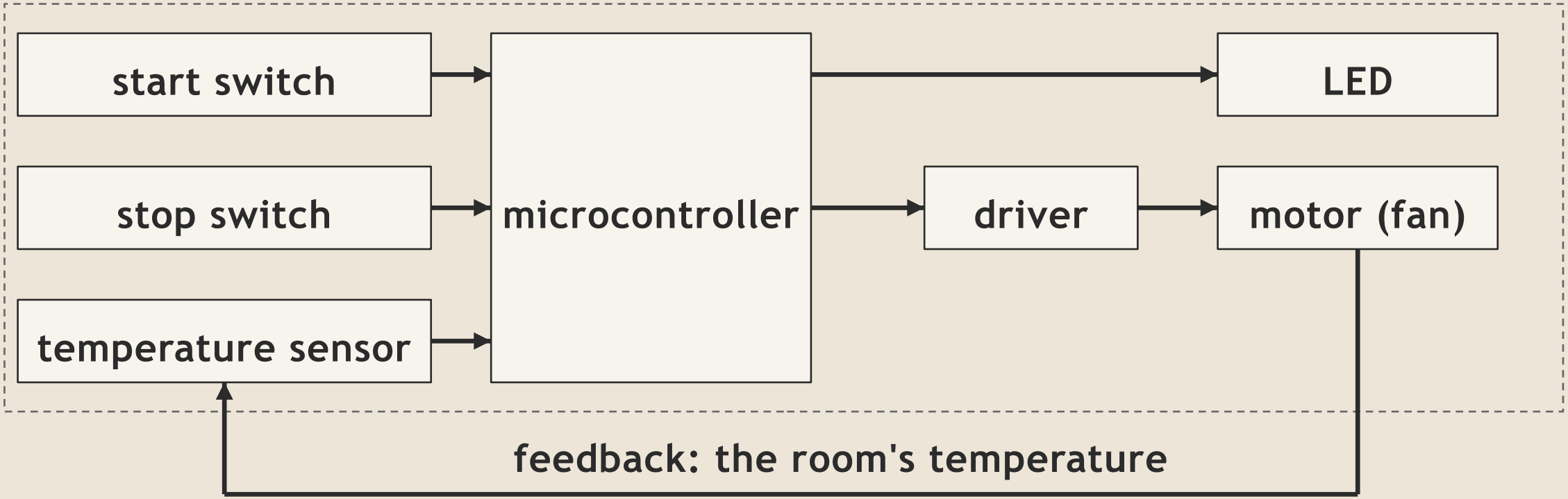
Open Loop: The Fan

Open loop: nothing measures the room.



Closed Loop: The Improved Fan

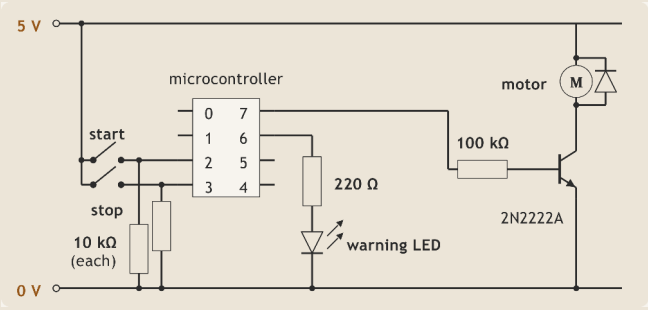
Closed loop: a sensor feeds back the temperature.



Practical 4: The Server-Room Fan

Practical sheet 4

Simulate it **as drawn** in Yenka. **Print** it.
Test, **amend**, **re-test**. **Print** it again.
Build the amended fan on your **Uno**.



SAFETY USB power only. Unplug before you move a wire.

Answers: Practical 4

Test	Initial result, amendment, re-test
Test 1 Start	Flashes 2 times. Change x2 to x3. Re-test: 3 times.
Test 2 Fan	Fan does not turn. Reduce 100 kΩ to 470 Ω. Re-test: it turns.
Test 3 Stop	Fan stops, waits for start. No amendment required.

Plenary



1. **State** why you **build** the design **exactly as drawn**.
2. **State** what the test table gets with **no hard copy**.
3. **State** the **column** that says what **should** happen.

Plenary

4 The motor **never turns**. **State** one **circuit** fault.

5 **State** what makes the fan **closed loop**.

Look back — by the end of this part…



1. I can **build** a design **exactly as drawn**
2. I can **tell** a **circuit fault** from a **flowchart fault**
3. I can **fill in** a **test table** and **prove** it with a **hard copy**

LOOKING AHEAD

Booklet 4a: Logic and Digital Control

From programs to logic gates.

Course Specification

Digital electronic control systems

- using suitable commands (high, low, for…next, if…then, pause, end (or their equivalents)) to design programs to solve simple control problems, involving time delays and continuous and fixed loops

Credits

The big question (An Arduino Uno): Photo: JrawX, CC0, via Wikimedia Commons

Marks and marking points (slides 6, 7, 11, 13, 14, 26): © Qualifications Scotland (SQA). N5 Engineering Science Assignment marking instructions 2024 and 2025, Task 2 (marks and marking points only).

Drawings (all slides): Circuit diagrams, flowcharts and the hard-copy page drawn by R Stewart.

The Test Table (2026 worksheets): © Qualifications Scotland (SQA). N5 Engineering Science Assignment worksheets 2026 (column headings only).

Specification Met? (2026 worksheets): © Qualifications Scotland (SQA). N5 Engineering Science Assignment worksheets 2026 (column headings only).

Course Specification: © Qualifications Scotland (SQA). Course spec v2.0, p.5

R Stewart - October 2026