

TechyTron STEM Electronics

Classes 7-8 · Ages 12–14

Ten guided electronics experiments. Typical school-age ranges vary by learner.

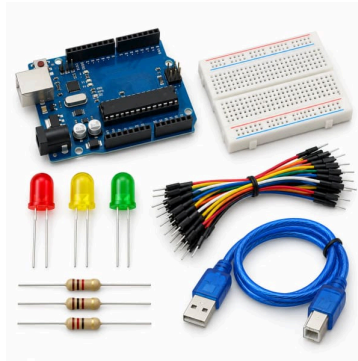
Each kit is supplied separately with the components listed in its guide.

- Traffic Signal Coder Kit
- Reaction Time Challenge Kit
- Ultrasonic Parking Guide Kit
- Servo Angle Dial Kit
- Melody Maker Kit
- Rainbow PWM Mixer Kit
- Light Meter Logger Kit
- Mini Weather Station Kit
- Digital Dice Kit
- IR Visitor Counter Kit

Traffic Signal Coder Kit

Classes 7-8 · Ages 12–14

Program a red, amber and green light sequence.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- A program can sequence multiple outputs.
- Each LED needs its own resistor.
- Timing changes the behaviour without changing the wiring.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- 5mm LED - Red Color — 1 LED
- 5mm LED - Yellow Color — 1 LED
- 5mm LED - Green Color — 1 LED
- 330 Ω 1/4 Watts (10 PCS) — 3 resistor(s) · Quantity is individual resistors; catalogue pack is divided by 10.

Connection guide

- D8 → 330 Ohm → red LED anode; D9 → 330 Ohm → yellow anode; D10 → 330 Ohm → green anode.
- All three cathodes → GND; no LED is connected directly to a GPIO pin.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Wire the three independently limited LED branches.
3. Open the supplied sketch and predict the output of each stage before uploading.
4. Upload, observe three full cycles, and time each colour.
5. Change the red interval from 3000 ms to 5000 ms, re-upload and compare.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

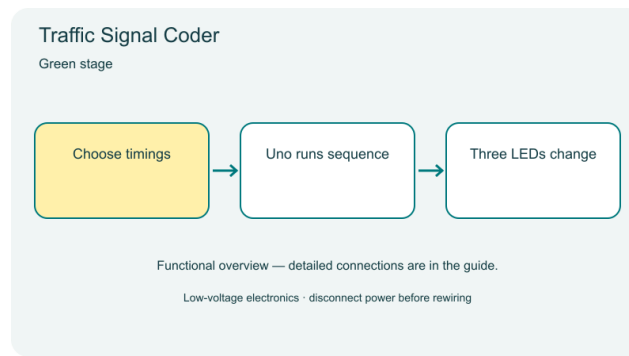
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Uses libraries supplied with the selected board core.

[Download starter sketch](#)

How this experiment works



Test challenge

Does exactly one light stay on at a time, and does the changed interval match the program?

Observation task

Stage, predicted colour, observed colour and measured duration.

Expected result

The LEDs repeat a green → amber → red sequence with programmed durations.

If it does not work

Match D8/D9/D10 to the colour assignments and inspect LED polarity. Uploading requires the selected Uno port.

Try next

Write a timing table for a pedestrian crossing before changing the sketch.

Use and supervision

Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

Reaction Time Challenge Kit

Classes 7-8 · Ages 12–14

Measure reaction time using a light cue and button.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- INPUT_PULLUP reads LOW when a button connects to ground.
- Elapsed time is the difference between two timestamps.
- Repeated trials reveal variation and false starts.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- 5mm LED - Red Color — 1 LED
- 330 Ω 1/4 Watts (10 PCS) — 1 resistor · Quantity is individual resistors; catalogue pack is divided by 10.
- Tactile Push Button Switch 12x12x8mm 4-Pin — 1 button

Connection guide

- D9 → 330 Ohm → LED anode; LED cathode → GND.
- Push button between D2 and GND; use opposite button contact pairs.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Wire the LED and button; upload the reaction sketch.
3. Open Serial Monitor and keep the button released while waiting.
4. Press as soon as the LED comes on; read the elapsed milliseconds.
5. Collect ten trials and compare median, minimum and false starts.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

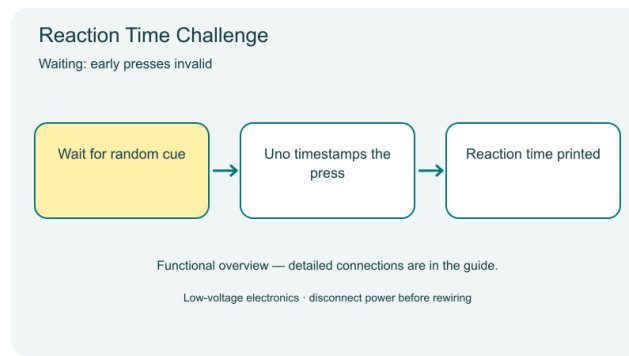
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Uses libraries supplied with the selected board core.

[Download starter sketch](#)

How this experiment works



Test challenge

Compare normal trials with one deliberately early press.

Observation task

Trial number, reaction milliseconds, false start and median of valid trials.

Expected result

A valid response prints elapsed milliseconds; an early press is marked as a false start.

If it does not work

A permanently LOW button usually means the two jumper wires use the same internal contact pair.

Try next

Compare two sets of ten trials without claiming a medical or attention assessment.

Use and supervision

Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

Ultrasonic Parking Guide Kit

Classes 7-8 · Ages 12–14

Turn ultrasonic echoes into a distance-based sound alert.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- Distance can be estimated from echo round-trip time.
- Missing echoes must be handled as invalid readings.
- Thresholds turn a measured value into a response.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- HC-SR04 Ultrasonic Distance Sensor Module — 1 sensor module
- Small Piezoelectric Buzzer - 5V Passive Buzzer — 1 piezo sounder · Passive piezo type, driven by tone(); not an electromagnetic high-current buzzer.

Connection guide

- HC-SR04 VCC → 5 V; GND → GND; TRIG → D7; ECHO → D6.
- Passive piezo + → D9; – → GND.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Wire the sensor and passive piezo, pointing the transducers toward a flat target.
3. Upload and open Serial Monitor; compare a ruler with readings at 10, 20 and 40 cm.
4. Move the target inside 15 cm and listen for the alert.
5. Remove the target and confirm that an invalid/no-echo reading is not treated as zero distance.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

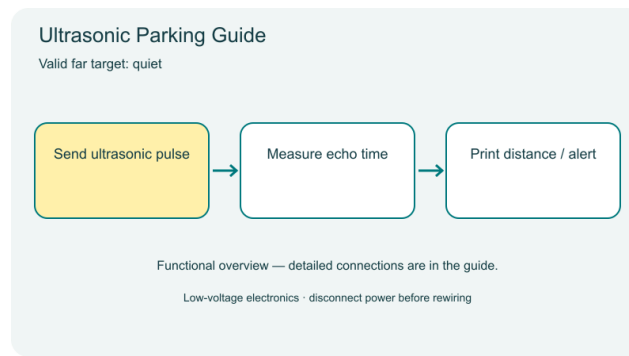
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Uses libraries supplied with the selected board core.

[Download starter sketch](#)

How this experiment works



Test challenge

Compare five readings at each known target distance; vary target angle once.

Observation task

Ruler distance, five sensor estimates, mean error and any invalid echoes.

Expected result

The sketch prints a distance estimate and sounds only when a valid target is closer than 15 cm.

If it does not work

Confirm TRIG/ECHO assignments and use a firm flat target. Soft or angled surfaces may not return a usable echo.

Try next

Replace the fixed alarm with three distance bands.

Use and supervision

Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

Servo Angle Dial Kit

Classes 7-8 · Ages 12–14

Map a rotary dial position to a servo angle.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- A potentiometer supplies an analog input.
- An input range can be mapped to an output range.
- A servo needs a suitable power supply separate from its signal pin.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- Tower Pro SG90 Servo Motor - 9 gms Mini/Micro Servo Motor — 1 servo
- 10 K POTENTIOMETER — 1 potentiometer
- Enclosed Regulated 5 V Servo Supply with Fitted Lead — 1 power supply

Connection guide

- Pot outer pins → Uno 5 V/GND; wiper → A0. Servo signal → D9.
- Servo red/+ → separate supplied 5 V supply; brown/black → its GND. Join that GND to Uno GND.
- Do not join the external supply +5 V rail to the USB-powered Uno 5 V rail.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Leave the servo horn unloaded and connect the two power domains exactly as shown.
3. Upload the sketch before enabling the external servo supply.
4. Turn the dial slowly from end to end; compare angles near 0%, 50% and 100% input.
5. Stop if the servo strains or hits a mechanical limit; the sketch restricts commands to 10–170 degrees.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

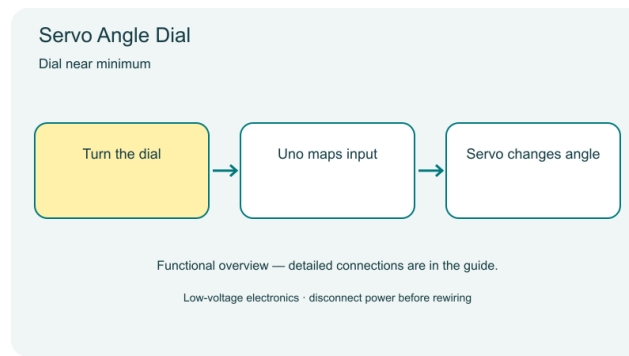
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Install from Arduino Library Manager: Servo (Arduino).

[Download starter sketch](#)

How this experiment works



Test challenge

Compare five knob positions with the commanded angles printed in Serial Monitor.

Observation task

Analog reading, mapped angle and observed horn position.

Expected result

The unloaded servo follows the dial within its practical mechanical limits.

If it does not work

Jitter or resets often indicate inadequate servo power or a missing shared ground. Never power the servo from a GPIO.

Try next

Restrict travel to 45–135 degrees and explain the mapping change.

Use and supervision

Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

Melody Maker Kit

Classes 7-8 · Ages 12–14

Play three notes using push buttons and a passive piezo.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- Pitch is related to sound frequency.
- A passive piezo requires an alternating signal.
- Button priority can be expressed as program logic.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- Small Piezoelectric Buzzer - 5V Passive Buzzer — 1 piezo sounder · Passive piezo type, driven by tone(); not an electromagnetic high-current buzzer.
- Tactile Push Button Switch 12x12x8mm 4-Pin — 3 button(s)

Connection guide

- Three buttons: D2, D3 and D4 respectively → GND when pressed.
- Passive piezo + → D9; – → GND.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Wire the three pull-up inputs and passive piezo.
3. Upload and press each button individually to hear its assigned note.
4. Press two buttons and identify the program priority.
5. Change one frequency by a factor of two and compare pitch.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

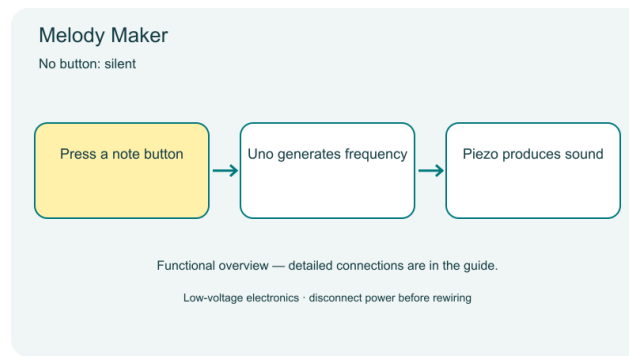
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Uses libraries supplied with the selected board core.

[Download starter sketch](#)

How this experiment works



Test challenge

Compare 262 Hz and 524 Hz using the same button assignment.

Observation task

Button, programmed frequency and perceived pitch order.

Expected result

Each button plays a different tone; releasing all buttons stops the sound.

If it does not work

Use the supplied passive piezo, not an active fixed-tone buzzer; check button contact pairs.

Try next

Create a five-note melody by sequencing tone calls and pauses.

Use and supervision

Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

Rainbow PWM Mixer Kit

Classes 7-8 · Ages 12–14

Mix colours by programming three PWM channels.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- PWM changes average LED brightness.
- A common-cathode RGB LED has three colour channels.
- Each colour channel needs a separate resistor.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- RGB LED - 5mm - Common Cathode — 1 LED · Common-cathode RGB LED; one current-limiting resistor per colour.
- 330 Ω 1/4 Watts (10 PCS) — 3 resistor(s) · Quantity is individual resistors; catalogue pack is divided by 10.

Connection guide

- Common cathode → GND. Red anode → D9 through 330 Ohm; green → D10 through 330 Ohm; blue → D11 through 330 Ohm.
- Use the supplied LED pinout labels, not lead length alone.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Wire all three resistor-limited channels.
3. Upload the sketch and identify the first three single-colour stages.
4. Record how red+green, green+blue and red+blue appear.
5. Change one duty value from 255 to 80 and compare the mixture.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

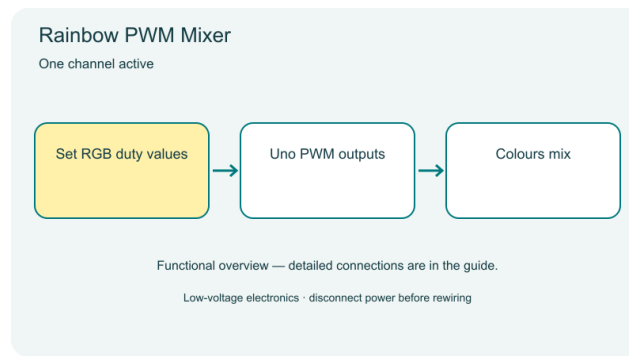
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Uses libraries supplied with the selected board core.

[Download starter sketch](#)

How this experiment works



Test challenge

Predict six colour stages before running them; observe whether equal numeric values look equally bright.

Observation task

Red/green/blue duty values and perceived mixed colour.

Expected result

Three independent PWM outputs create distinct primary and mixed colours.

If it does not work

A common-anode LED uses different wiring and inverted values; use the supplied common-cathode part.

Try next

Add smooth fades between the colour stages.

Use and supervision

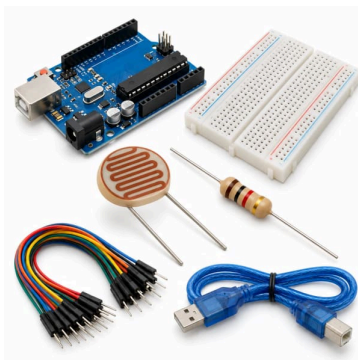
Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

Light Meter Logger Kit

Classes 7-8 · Ages 12–14

Record changing light as a stream of analog readings.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- An LDR changes resistance with light.
- A voltage divider converts resistance change to voltage.
- Raw readings are relative; they are not calibrated lux.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- LDR Sensor (Light Dependent Resistor) - 5mm — 1 photoresistor
- 10K Ω 1/4 Watts (10 PCS) — 1 resistor · Quantity is individual resistors; catalogue pack is divided by 10.

Connection guide

- 5 V → LDR → A0 node → 10 kOhm → GND.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Build the divider and upload the sketch.
3. Open Serial Plotter and observe the baseline for ten seconds.
4. Cover the LDR, uncover it, then illuminate it with a room lamp at two distances.
5. Copy the serial CSV into a spreadsheet and annotate the conditions.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

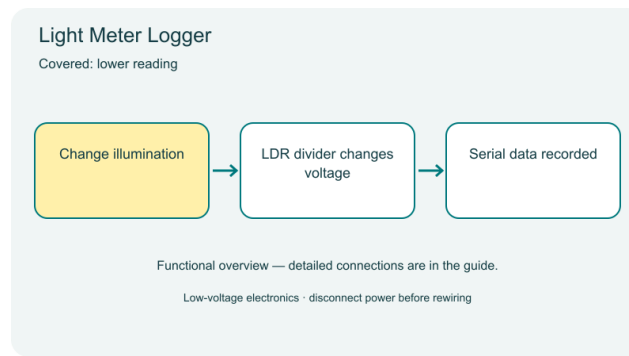
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Uses libraries supplied with the selected board core.

[Download starter sketch](#)

How this experiment works



Test challenge

Collect ten readings per condition without changing the circuit.

Observation task

Elapsed milliseconds, raw ADC count and light condition.

Expected result

In this divider orientation brighter light normally increases ADC readings.

If it does not work

An unconnected A0 pin floats. Reverse sensor/resistor order only when deliberately testing an inverted response.

Try next

Calibrate a relative 0–100 scale using your own dark and bright references.

Use and supervision

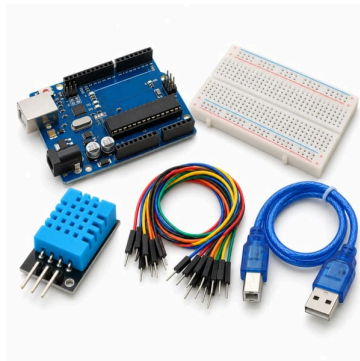
Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

Mini Weather Station Kit

Classes 7-8 · Ages 12–14

Read room temperature and humidity with a digital sensor.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- Digital sensors exchange encoded readings.
- Sampling rate and invalid readings matter.
- Room observations differ from a calibrated weather instrument.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- DHT11 — 1 sensor module · Supplied with DATA/VCC/GND labels and required data pull-up fitted.
- 10K Ω 1/4 Watts (10 PCS) — 1 resistor · Quantity is individual resistors; catalogue pack is divided by 10.

Connection guide

- DHT11 labelled VCC → 5 V; GND → GND; DATA → D2.
- For a bare four-pin DHT11, supplied 10 kOhm pull-up connects DATA to VCC; unused NC pin stays unconnected.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Identify DATA/VCC/GND using the supplied sensor pin card.
3. Install the listed libraries, upload and open Serial Monitor.
4. Record readings for three minutes with at least two seconds between samples.
5. Move the sensor away from a hand or warm device, allow it to settle and compare.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

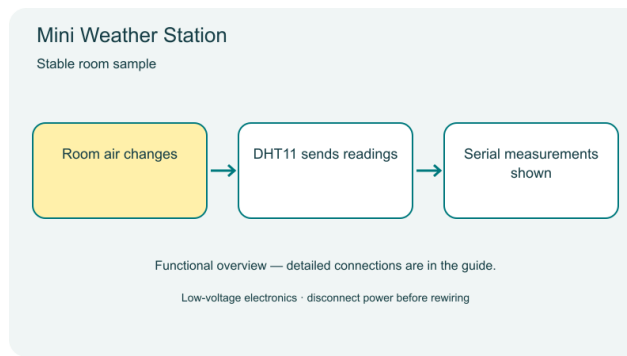
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Install from Arduino Library Manager: DHT sensor library (Adafruit), Adafruit Unified Sensor.

[Download starter sketch](#)

How this experiment works



Test challenge

Compare a stable room position with a gently changed room location, allowing settling time.

Observation task

Time, temperature in degrees C, relative humidity and any invalid sample.

Expected result

Valid readings are printed as temperature/humidity pairs; failed reads are explicitly flagged.

If it does not work

Check the DHT11 type, DATA pull-up, pin order and two-second sampling interval.

Try next

Discuss sensor resolution, placement and how repeatability differs from accuracy.

Use and supervision

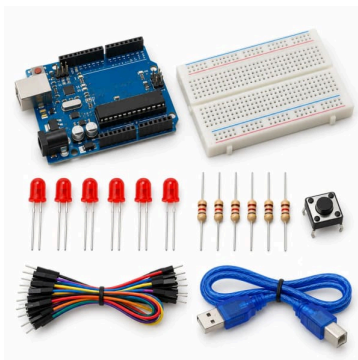
Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

Digital Dice Kit

Classes 7-8 · Ages 12–14

Make an electronic dice display and test its outcomes.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- A pseudo-random generator produces programmed sequences.
- One outcome can select a set of outputs.
- A small sample does not prove equal long-term probabilities.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- 5mm LED - Red Color — 6 LED(s)
- 330 Ω 1/4 Watts (10 PCS) — 6 resistor(s) · Quantity is individual resistors; catalogue pack is divided by 10.
- Tactile Push Button Switch 12x12x8mm 4-Pin — 1 button

Connection guide

- Six LED anodes → D3–D8 respectively, each through its own 330 Ohm resistor; cathodes → GND.
- Roll button → D2 and GND.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Arrange six labelled LED positions on the breadboard and wire each independently.
3. Upload, press and release the roll button, and count the lit LEDs.
4. Collect at least 30 rolls; record a frequency table for 1–6.
5. Reset the board and discuss how the analog seed differs from true randomness.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

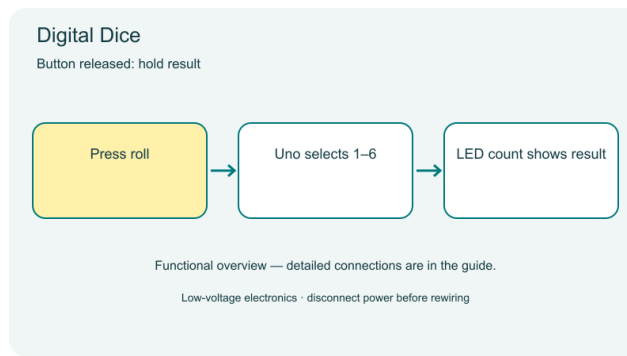
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Uses libraries supplied with the selected board core.

[Download starter sketch](#)

How this experiment works



Test challenge

Check that every roll gives 1–6 illuminated LEDs and compare the outcome frequencies.

Observation task

Roll number, result and tally for each outcome.

Expected result

Each valid press selects a result from one to six; this is a learning model, not a certified gaming device.

If it does not work

Count independent resistor branches; do not share one resistor between all LEDs.

Try next

Display dice pip patterns instead of a linear bar.

Use and supervision

Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.

IR Visitor Counter Kit

Classes 7-8 · Ages 12–14

Count separate object passes with an infrared sensor.



Activity engagement: 60–90 minutes. Explore 10–15 min · Build 20–30 min · Test and discuss 30–45 min. Times are estimates for guided use.

You will need: A computer with a USB port and Arduino IDE is required and is not included.

What you learn

- A transition can count one event without counting every loop.
- Reflective IR sensing depends on target surface and distance.
- A single sensor cannot determine direction or reliably count crowds.

Included components

- Small Breadboard / Mini Solderless Board - 400 points — 1 breadboard
- Jumper Wire Combo - Male to Male, Male to Female and Female to Female Breadboard Connecting Wires (10 Qty Each) — 1 set
- Illustrated Activity Guide and Observation Sheets — 1 set
- Fitted Headers and Labelled Connection Leads — 1 set
- Arduino Uno R3 CH340G version compatible Arduino Development Board — 1 controller board
- USB A to B Programming Cable for Arduino Uno/ Arduino Mega — 1 data cable
- IR Obstacle Avoidance Sensor Module - LM393 Photoelectric Sensor Module — 1 sensor module
- Tactile Push Button Switch 12x12x8mm 4-Pin — 1 button

Connection guide

- IR module VCC → 5 V; GND → GND; DO → D2.
- Reset button → D4 and GND. Supplied module is configured active-low on detection.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Mount the sensor facing a single clear path and adjust sensitivity for a card-sized target.
3. Upload and move one object into range, hold it there and then remove it.
4. Repeat ten separate passes; compare count with actual passes.
5. Press reset and repeat with a dark target.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

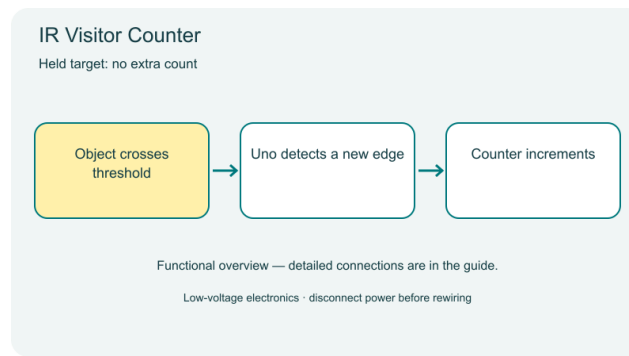
Programming resources

Arduino IDE: select Arduino Uno and the USB port. Use the supplied USB data cable. Serial Monitor/Plotter: 115200 baud.

Uses libraries supplied with the selected board core.

[Download starter sketch](#)

How this experiment works



Test challenge

Does holding one target in range count only once until it leaves?

Observation task

Target type, actual separate passes, reported count and missed detections.

Expected result

A confirmed clear-to-detected transition increments the count once; the reset button clears it.

If it does not work

Check active-low DO configuration, threshold setting and sufficient clear time between passes.

Try next

Design a two-sensor sequence for determining direction, without claiming it is included here.

Use and supervision

Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

Cover images illustrate component types. Flow animations explain principles and are not recorded prototype tests. Use the exact connection labels supplied with the kit.