

TechyTron STEM Electronics

Classes 9-10 · Ages 14–16

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

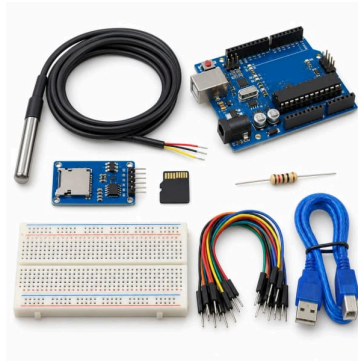
Each kit is supplied separately with the components listed in its guide.

- Temperature Data Recorder Kit
- OLED Weather Dashboard Kit
- RFID Access Indicator Kit
- Precision Voltage Monitor Kit
- Keypad PIN Lock Kit
- Motion Event Alarm Kit
- Optical Speed Lab Kit
- Smart Streetlight Control Kit
- Tilt and Acceleration Logger Kit
- Water Level Display Kit

Temperature Data Recorder Kit

Classes 9-10 · Ages 14–16

Save temperature readings as a CSV file on a memory card.



Activity engagement: 90–120 minutes. Explore 15–20 min · Build 30–40 min · Test and discuss 45–60 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

- Different digital buses can share one controller.
- A CSV file stores a sequence of measurements.
- Sensor and storage failures need explicit handling.

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
- DS18B20 Water Proof Temperature Sensor Module — 1 probe
- Data Logging Shield for Arduino with SD card Interface and RTC — 1 logging shield · Uses SD interface on SPI; RTC function is not required and no RTC backup battery is supplied.
- 8 GB microSD Card with SD Adapter — 1 card set
- 4.7K Ω 1/4 Watts (10 PCS) — 1 resistor · Quantity is individual resistors; catalogue pack is divided by 10.
- Micro SD Memory Card Reader for Raspberry Pi -Mini USB 2.0 — 1 USB card reader

Connection guide

- Unpowered Uno: fit the SD logging shield, leaving its RTC battery socket empty. SD chip-select is D10; SPI uses D11/D12/D13.
- DS18B20 labelled VCC → 5 V; GND → GND; DATA → D2 with 4.7 kOhm pull-up to 5 V.
- Use the supplied microSD-to-SD adapter in the shield; format the supplied 8 GB card as FAT32.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Format the card and insert it with power disconnected. Identify probe wires from their labels.
3. Install the listed libraries; open and upload the supplied sketch.
4. Record room temperature for three minutes, then hold only the metal probe tip in your hand.
5. Disconnect USB before removing the card. Read TEMP.CSV through the included USB card reader and plot temperature against elapsed milliseconds.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

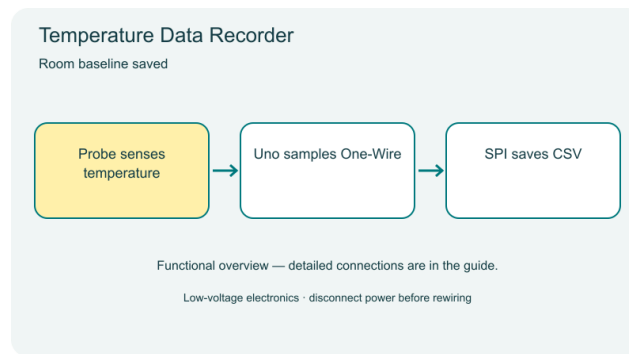
Programming resources

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: OneWire, DallasTemperature, SD (Arduino).

[Download starter sketch](#)

How this experiment works



Test challenge

Compare room, hand-warmed and cooling segments; then test the error message with the card removed while power is off.

Observation task

Elapsed milliseconds, degrees C, segment label and any missing-sensor/storage messages.

Expected result

Valid temperature readings append to TEMP.CSV; timing is elapsed time since reset, not an RTC timestamp.

If it does not work

Check FAT32, shield CS D10, probe pin labels and the 4.7 kOhm pull-up. A USB card reader cannot replace the SPI logging shield.

Try next

Calculate the time needed for the warmed probe to return halfway toward room temperature.

Use and supervision

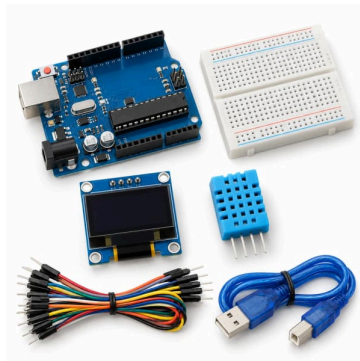
Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

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

OLED Weather Dashboard Kit

Classes 9-10 · Ages 14–16

Show temperature and humidity on a compact OLED screen.



Activity engagement: 90–120 minutes. Explore 15–20 min · Build 30–40 min · Test and discuss 45–60 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

- I2C shares a clock and data pair between a controller and display.
- A display should distinguish missing data from real readings.
- Logic-level compatibility matters when combining modules.

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
- 0.96 inch Monochrome OLED Display (SSD1306) — 1 display module · I2C SSD1306, 128x64 pixels, fitted header; verify address 0x3C.
- 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.
- Bidirectional 3.3 V / 5 V I2C Level Shifter — 1 level-shifter module

Connection guide

- OLED VCC → Uno 3.3 V; GND → GND. I2C level shifter LV → 3.3 V, HV → 5 V, common GND; Uno A4/A5 → HV1/HV2, OLED SDA/SCL → LV1/LV2.
- DHT11 VCC → 5 V; GND → GND; DATA → D2 with supplied 10 kOhm pull-up to 5 V.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Wire the OLED through its supplied level shifter and connect the DHT11.
3. Install the libraries, upload, and confirm the OLED address is 0x3C using the supplied diagnostic sketch if needed.
4. Observe the screen for three minutes at room conditions, leaving two seconds between updates.
5. Disconnect power before deliberately removing DATA, then restart and compare the sensor-error display.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

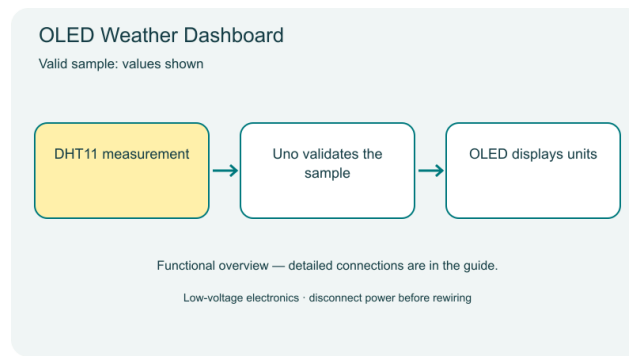
Programming resources

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, Adafruit GFX Library, Adafruit SSD1306.

[Download starter sketch](#)

How this experiment works



Test challenge

Confirm that the screen presents valid units and a clear error instead of stale temperature when the sensor is absent.

Observation task

Time, temperature, humidity, settling conditions and error state.

Expected result

The OLED shows valid temperature/humidity pairs or a sensor-error message.

If it does not work

Verify OLED address, 128x64 resolution, SDA/SCL placement and the separate 3.3/5 V level-shifter supplies.

Try next

Add minimum and maximum values for the current powered session.

Use and supervision

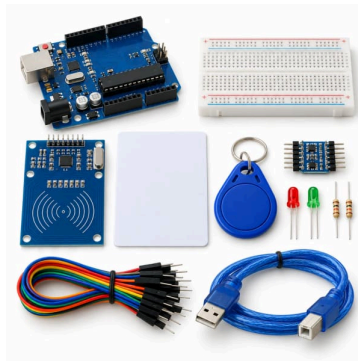
Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

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

RFID Access Indicator Kit

Classes 9-10 · Ages 14–16

Recognise an enrolled RFID tag and indicate the result.



Activity engagement: 90–120 minutes. Explore 15–20 min · Build 30–40 min · Test and discuss 45–60 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 RFID reader returns a tag identifier.
- A comparison rule can control separate accepted/rejected indicators.
- UID matching alone is not secure access control.

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
- RC522 13.56 MHz RFID Reader Writer Module with Card and Tag — 1 reader set · Includes compatible 13.56 MHz card and key tag. Reader supply and logic are 3.3 V.
- 4-Channel 5 V to 3.3 V SPI Level Shifter (74LVC125) — 1 level-shifter module
- 5mm LED - Red Color — 1 LED
- 5mm LED - Green Color — 1 LED
- 330 Ω 1/4 Watts (10 PCS) — 2 resistor(s) · Quantity is individual resistors; catalogue pack is divided by 10.

Connection guide

- RC522 VCC → 3.3 V, GND → GND. Never connect its supply or signal inputs directly to 5 V.
- 74LVC125 shifter VCC → 3.3 V, GND → GND; its enabled outputs lower Uno D13/SCK, D11/MOSI, D10/SS and D9/RST to RC522 SCK, MOSI, SDA/SS and RST respectively.
- RC522 MISO → Uno D12 directly; IRQ remains unconnected. Green LED D6 and red LED D7 each use their own 330 Ohm resistor to GND.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Connect the reader through the supplied SPI-capable level shifter.
3. Install MFRC522 and upload; scan the supplied card and write down its UID bytes from Serial Monitor.
4. Replace the demonstration UID array and its length with that card UID, then upload again.
5. Compare the enrolled card and a different supplied tag; treat the LEDs as a classroom decision model.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

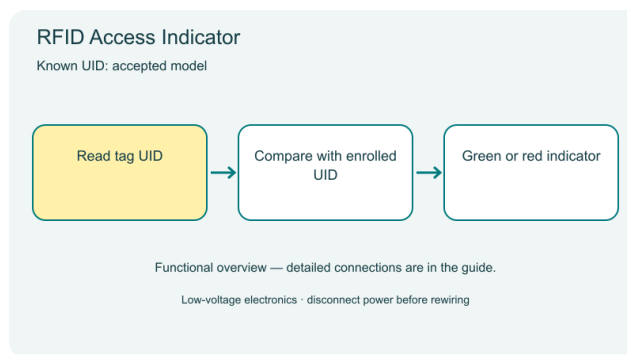
Programming resources

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: MFRC522.

[Download starter sketch](#)

How this experiment works



Test challenge

Test enrolled and different tags five times each, including removing the tag between reads.

Observation task

Tag label, UID length, expected decision and observed LED.

Expected result

The configured UID lights green; a different UID lights red. No real lock or security guarantee is provided.

If it does not work

Verify 3.3 V reader power, SPI pin order, shifter direction and UID length. A 125 kHz card does not work with this 13.56 MHz reader.

Try next

Explain why copying or changing a UID defeats a UID-only rule and list stronger authentication requirements.

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.

Precision Voltage Monitor Kit

Classes 9-10 · Ages 14–16

Calibrate a low-voltage monitor against a multimeter.



Activity engagement: 75–105 minutes. Explore 15–20 min · Build 25–35 min · Test and discuss 35–50 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 divider scales the input before an ADC measures it.
- Reference voltage and resistor tolerance affect accuracy.
- Calibration needs a separate reference 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
- DT9205A Digital Multimeter with Probes and Battery — 1 multimeter · Includes probes and meter battery. Used only on isolated low-voltage circuits.
- 10K Ω 1/4 Watts (10 PCS) — 2 resistor(s) · Quantity is individual resistors; catalogue pack is divided by 10.
- 4 CELL AA BATTERY HOLDER WITH SWITCH / COVER — 1 holder
- AA BATTERY — 4 battery (batteries)

Connection guide

- Isolated battery holder positive → 10 kOhm → A0 midpoint → 10 kOhm → battery negative. Battery negative also → Uno GND.
- Power Uno only by USB. Do not connect battery positive to Uno 5 V, USB or another rail.
- Multimeter COM/V socket only, DC volts mode; input limited to 0–9 V DC.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Measure the two divider resistors with power off and inspect the separated battery circuit.
3. Upload and compare the reported voltage with the meter across the four-AA holder.
4. Measure the Uno 5 V rail with the meter and replace vRef in the sketch with that measured value.
5. Repeat measurements with two and four AA cells using the holder connection guide; switch off before altering cells.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

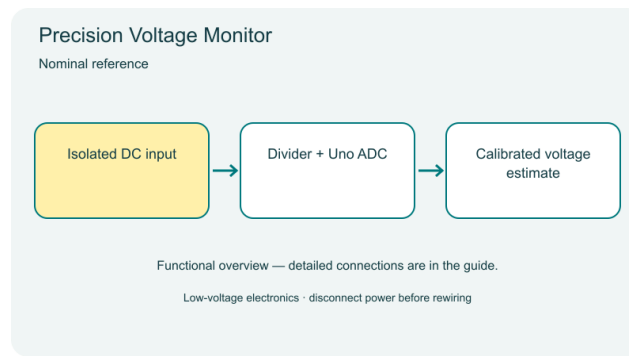
Programming resources

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 uncalibrated and reference-adjusted errors at two battery voltages.

Observation task

Meter voltage, raw ADC mean, assumed reference, calibrated estimate and percentage error.

Expected result

Divider-scaled estimates approach the multimeter reading after reference adjustment; this is not a precision-certified meter.

If it does not work

Check the common ground, midpoint and input limit. Never use mains, a vehicle battery or the meter current socket.

Try next

Include measured resistor values in the scaling equation instead of assuming a perfect 1:1 divider.

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.

Keypad PIN Lock Kit

Classes 9-10 · Ages 14–16

Build a keypad-controlled servo latch model.



Activity engagement: 90–120 minutes. Explore 15–20 min · Build 30–40 min · Test and discuss 45–60 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 matrix keypad shares row and column wires.
- Input buffering needs clear, submit and length limits.
- A servo model demonstrates a decision but is not a security lock.

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
- 4X4 MEMBRANE KEYPAD — 1 keypad
- Tower Pro SG90 Servo Motor - 9 gms Mini/Micro Servo Motor — 1 servo
- Enclosed Regulated 5 V Servo Supply with Fitted Lead — 1 power supply

Connection guide

- Keypad rows R1–R4 → Uno D2,D3,D4,D5; columns C1–C4 → D6,D7,D8,D10. Follow the supplied keypad ribbon pin map.
- Servo signal → D9; red → separate supplied 5 V supply; ground → that supply GND and Uno GND.
- Keep the external +5 V separate from the USB-powered Uno +5 V.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Wire the keypad using its row/column labels and leave the servo horn unloaded.
3. Install libraries and upload the sketch; enable the separate servo supply.
4. Enter 2468 followed by # to move the servo; use * to clear an entry and enter a wrong code to compare.
5. Change the code, repeat ten valid/invalid trials and document the state transitions.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

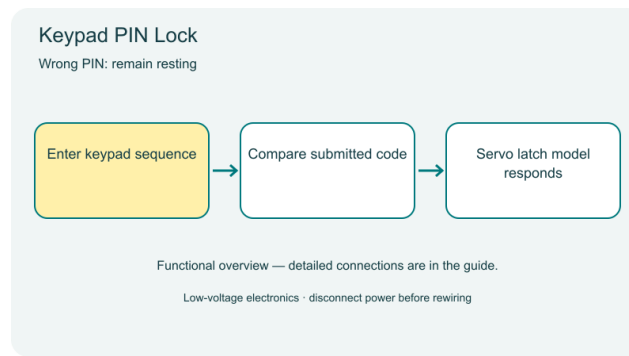
Programming resources

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: Keypad, Servo (Arduino).

[Download starter sketch](#)

How this experiment works



Test challenge

Test correct, incorrect, overlength and cleared entries; verify the model returns to its resting angle.

Observation task

Input sequence, expected decision, printed feedback and servo angle.

Expected result

A correct submitted code moves the unloaded servo briefly; an incorrect code does not.

If it does not work

Check row/column mapping and separate servo power. This demonstration has a visible code and no physical-security rating.

Try next

Add a limited number of attempts and a timed lockout state.

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.

Motion Event Alarm Kit

Classes 9-10 · Ages 14–16

Log motion events and control an electronic 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

- PIR detects changes in infrared patterns rather than measuring distance.
- Sensor warm-up and output hold time affect behaviour.
- Edge detection avoids logging the same held output repeatedly.

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-SR501 PIR Motion 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-SR501 VCC → 5 V; GND → GND; OUT → D2.
- Passive piezo + → D9; – → GND.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Set the PIR delay low and identify its retrigger jumper using the supplied card.
3. Upload and leave the sensor stationary for at least 60 seconds to settle.
4. Walk across its field of view, then compare moving directly toward it.
5. Record five event timestamps and compare stillness with movement.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

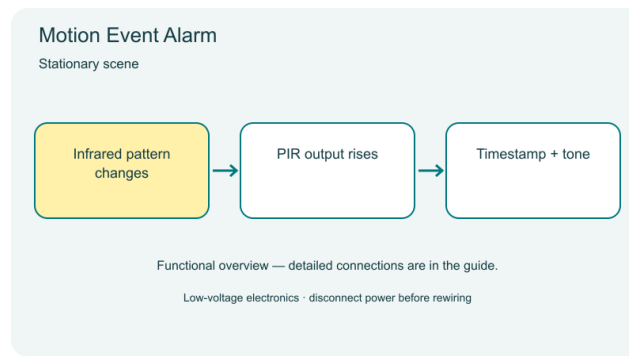
Programming resources

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 the sketch report one event per LOW-to-HIGH transition, rather than every loop?

Observation task

Movement direction, sensor setting, event time and observed output hold length.

Expected result

A rising PIR output prints an event time and starts a tone; a LOW output stops it.

If it does not work

Allow warm-up, avoid moving the board and inspect the delay/retrigger settings. PIR is not a reliable occupancy or life-safety detector.

Try next

Replace the continuous tone with a brief non-blocking event beep.

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.

Optical Speed Lab Kit

Classes 9-10 · Ages 14–16

Measure hand-turned wheel speed using optical pulses.



Activity engagement: 75–105 minutes. Explore 15–20 min · Build 25–35 min · Test and discuss 35–50 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 optical interrupter converts slots into pulses.
- RPM depends on pulses per revolution and measurement time.
- Short measurement windows trade smoothness for responsiveness.

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
- LM393 Motor Speed Measuring Sensor Module — 1 sensor module
- Twenty-Slot Encoder Disc and Hand-Turned Spindle Stand — 1 set

Connection guide

- LM393 optical slot sensor VCC → 5 V; GND → GND; DO → Uno D2.
- Mount the supplied twenty-slot disc so it passes through the sensor gap without touching it. Use the hand spindle only.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Assemble the stable spindle mount and count the disc slots before entering slotsPerTurn.
3. Wire the sensor, upload and turn the disc slowly by hand.
4. Compare one-turn and five-turn pulse totals at slow speed.
5. Turn steadily and compare RPM estimates using one-second and two-second windows.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

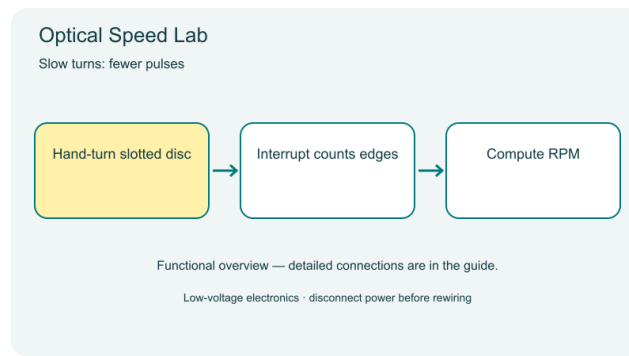
Programming resources

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 pulses per full revolution before trusting the displayed RPM.

Observation task

Slot count, measurement window, pulses and computed RPM.

Expected result

Pulses are counted on one edge; $\text{RPM} = \text{pulses} \times 60 / (\text{slots} \times \text{elapsed seconds})$.

If it does not work

Check sensor gap alignment and the actual disc slot count. Do not spin it with a drill or motor.

Try next

Estimate the minimum resolvable RPM increment for each window length.

Use and supervision

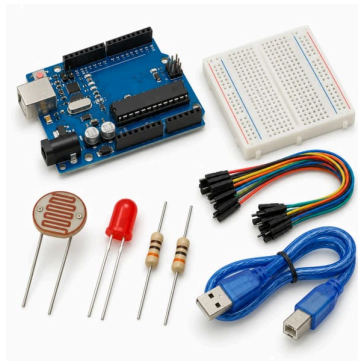
Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

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

Smart Streetlight Control Kit

Classes 9-10 · Ages 14–16

Program an automatic light with hysteresis and PWM dimming.



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

- Two switching thresholds can prevent chatter.
- PWM provides gradual brightness changes.
- A sensor should be shielded from its own controlled light when appropriate.

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.
- 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.

Connection guide

- 5 V → LDR → A0 node → 10 kOhm → GND.
- D9 → 330 Ohm → LED anode; cathode → GND. Keep the LED from shining directly on the LDR.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Build the divider and LED output; upload and observe raw light values.
3. Select darkOn and brightOff values between your covered and uncovered readings, keeping darkOn lower.
4. Move slowly across the boundary and observe whether the output chatters.
5. Compare this two-threshold behaviour with a single threshold and record the difference.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

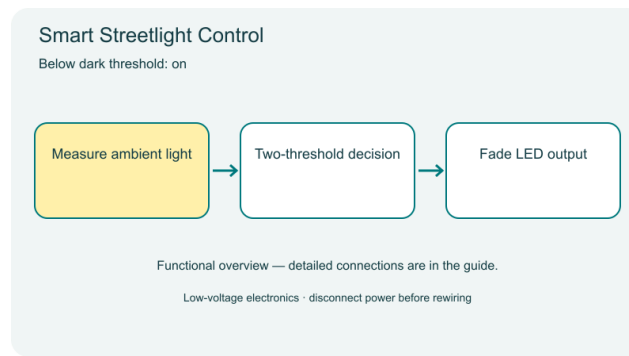
Programming resources

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

Hover light level near the boundary and count unwanted toggles over 30 seconds.

Observation task

Raw light, selected thresholds, LED state and toggle count.

Expected result

The LED turns on below the dark threshold and stays on until light exceeds the higher threshold; PWM changes brightness gradually.

If it does not work

Recalibrate thresholds for the room and check divider orientation; do not use this low-power LED model to switch mains lighting.

Try next

Add time-based dimming while keeping the hysteresis decision separate.

Use and supervision

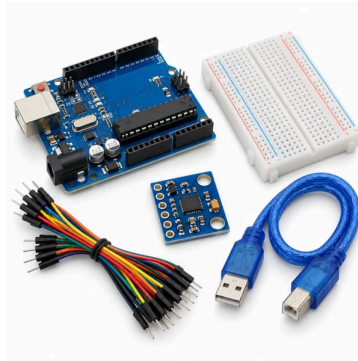
Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

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

Tilt and Acceleration Logger Kit

Classes 9-10 · Ages 14–16

Measure acceleration and compare sensor orientation.



Activity engagement: 90–120 minutes. Explore 15–20 min · Build 30–40 min · Test and discuss 45–60 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 accelerometer reports components of a vector.
- Gravity appears in stationary acceleration measurements.
- Offsets and mounting influence comparisons.

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
- MPU6050 3-Axis Gyroscope/Accelerometer Sensor Module — 1 sensor module · Supplied regulated GY-521 breakout: 5 V-compatible VCC and 3.3 V I2C, with fitted labelled header.
- Bidirectional 3.3 V / 5 V I2C Level Shifter — 1 level-shifter module

Connection guide

- Supplied regulated GY-521 VCC → 5 V; GND → GND. Its I2C signals remain at 3.3 V.
- Level shifter LV → 3.3 V, HV → 5 V, GND common; A4/A5 → HV1/HV2 and MPU SDA/SCL → LV1/LV2.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Connect the I2C shifter and regulated sensor breakout, then install libraries.
3. Upload and hold the module stationary on a level surface for ten seconds.
4. Place each labelled sensor axis upward in turn and record the acceleration vector.
5. Move it gently, compare dynamic readings, and distinguish acceleration from position.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

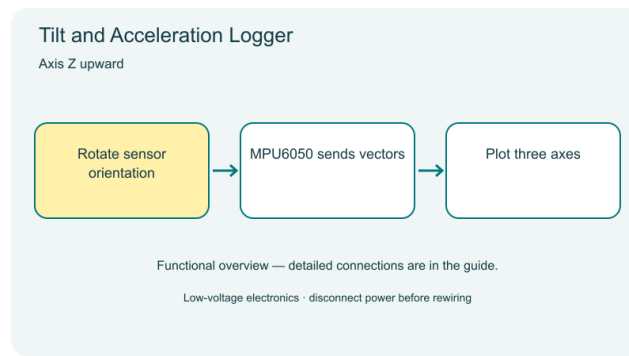
Programming resources

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: Adafruit MPU6050, Adafruit Unified Sensor, Adafruit BusIO.

[Download starter sketch](#)

How this experiment works



Test challenge

Does the stationary vector magnitude remain near 9.81 m/s^2 as the orientation changes?

Observation task

Orientation, mean x/y/z acceleration, vector magnitude and observed offsets.

Expected result

Gravity shifts between axes with orientation; gentle movement adds transient accelerations.

If it does not work

Verify address 0x68, level-shifter wiring and a stationary mount. Double-check that the supplied module has its 5 V VCC regulator.

Try next

Subtract a measured stationary baseline and explain why integration drifts.

Use and supervision

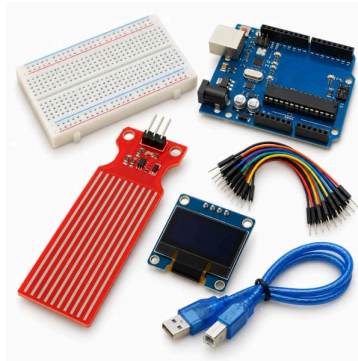
Use only the supplied low-voltage DC parts. Disconnect power before rewiring; check polarity and labelled pinouts. No soldering or exposed mains connections. Adult guidance is recommended.

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

Water Level Display Kit

Classes 9-10 · Ages 14–16

Calibrate a relative liquid-contact level display.



Activity engagement: 75–105 minutes. Explore 15–20 min · Build 25–35 min · Test and discuss 35–50 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 wet contact area changes a resistive sensor signal.
- Calibration maps two reference readings to a relative scale.
- Water composition and corrosion limit repeatability.

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
- Water Level Depth Detection Sensor Module For Arduino — 1 sensor module
- 0.96 inch Monochrome OLED Display (SSD1306) — 1 display module · I2C SSD1306, 128x64 pixels, fitted header; verify address 0x3C.
- Bidirectional 3.3 V / 5 V I2C Level Shifter — 1 level-shifter module

Connection guide

- Water sensor labelled + → 5 V; – → GND; S → A0. Keep its connector and electronics above the water.
- OLED VCC → Uno 3.3 V; GND → GND. I2C level shifter LV → 3.3 V, HV → 5 V, common GND; Uno A4/A5 → HV1/HV2, OLED SDA/SCL → LV1/LV2.

Steps to make it

1. Disconnect all power and identify the supplied part labels and pinouts before wiring.
2. Connect the dry sensor and OLED and upload the sketch.
3. Note dry and safe wet-reference raw readings, keeping water below the marked trace limit.
4. Enter those two values as dryRaw/wetRaw and re-upload.
5. Compare four immersion depths, then disconnect power, rinse the sensing end and dry it.
6. Change only the named parameter for the extension, repeat the controlled trials, save observations and disconnect power afterward.

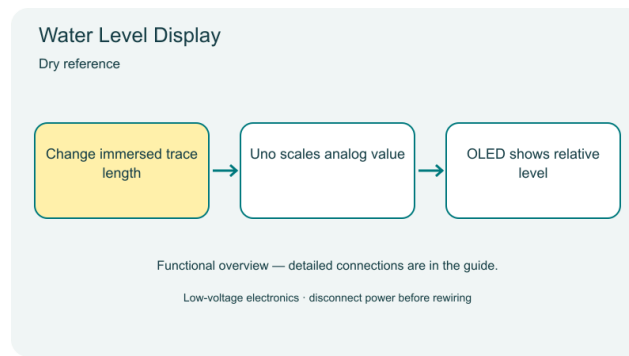
Programming resources

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: Adafruit GFX Library, Adafruit SSD1306.

[Download starter sketch](#)

How this experiment works



Test challenge

Compare the same marked depth on three repeats; do not interpret the relative percentage as tank volume.

Observation task

Immersion mark, raw value, scaled percentage and repeat variation.

Expected result

The screen shows a calibrated relative contact level; the reading depends on water conductivity and geometry.

If it does not work

Recalibrate if the water changes. Protect the board/USB connection from splashes and avoid prolonged probe energisation.

Try next

Draw a calibration curve for a specific vessel without extrapolating beyond the tested depths.

Use and supervision

Only the sensor trace end touches clean water; its connector, display, breadboard and computer remain dry. Use low-voltage USB power, disconnect before handling wet parts and dry after a short experiment. No pumps or mains wiring.

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