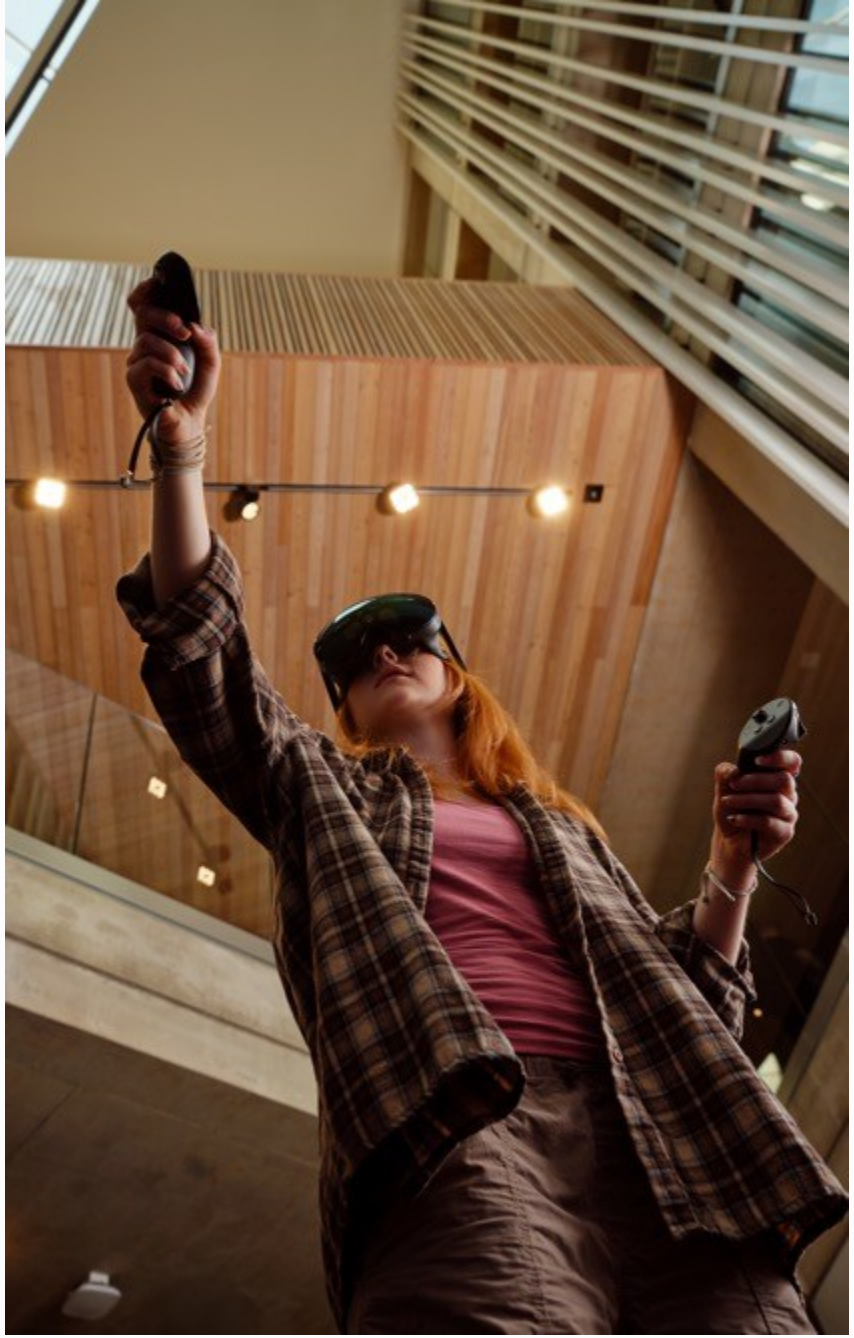
A Taste of Science and Engineering

Dr Leonardo Alves Dias



UNIVERSITY OF WARWICK

Agenda

01 Building prototypes with Arduino on TinkerCAD

02 Introduction to sensors and actuators.

01

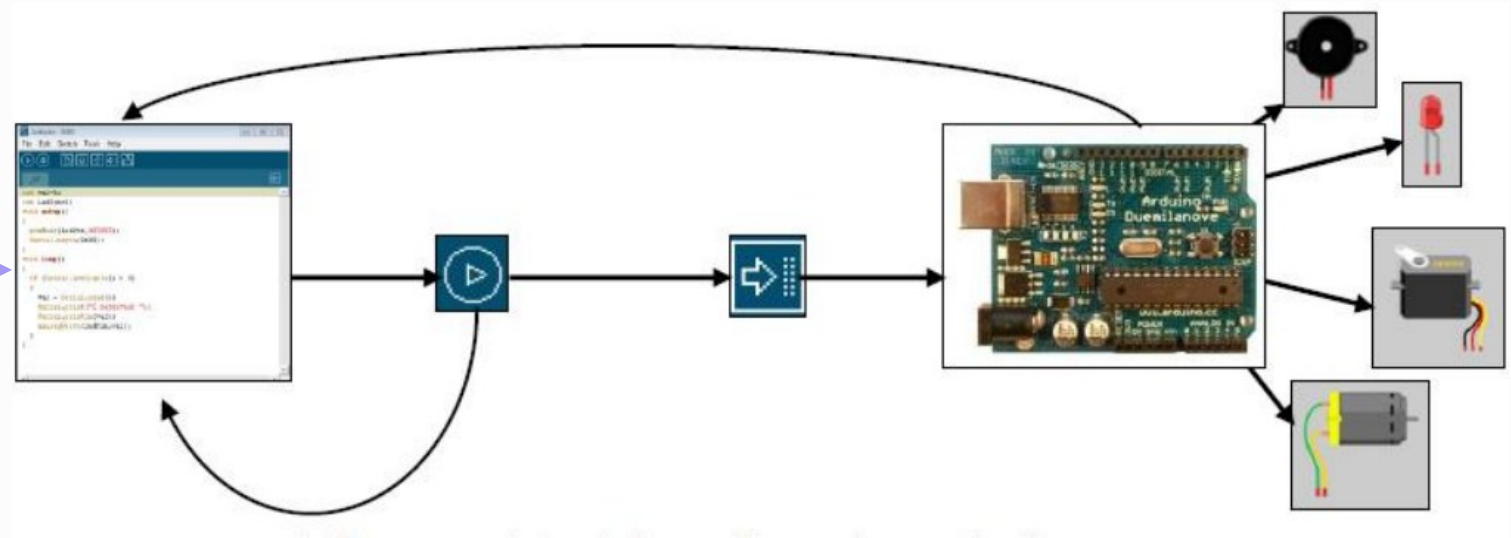
Prototyping with Arduino

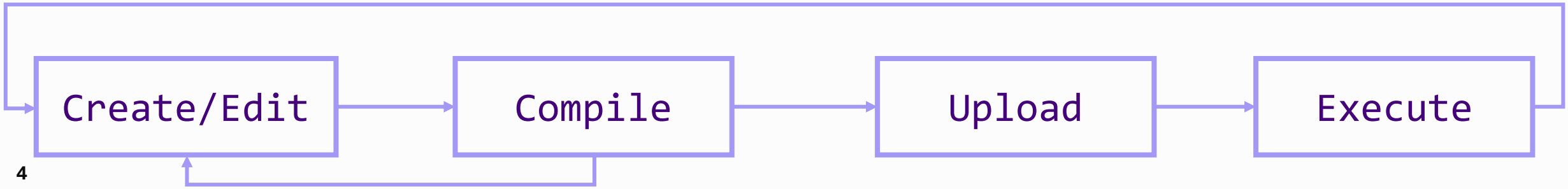
UNIVERSITY OF WARWICK

A Recap: Prototyping

A prototype is developed based on the following step-by-step:

1. Tinker with the circuit
2. Program the microcontroller





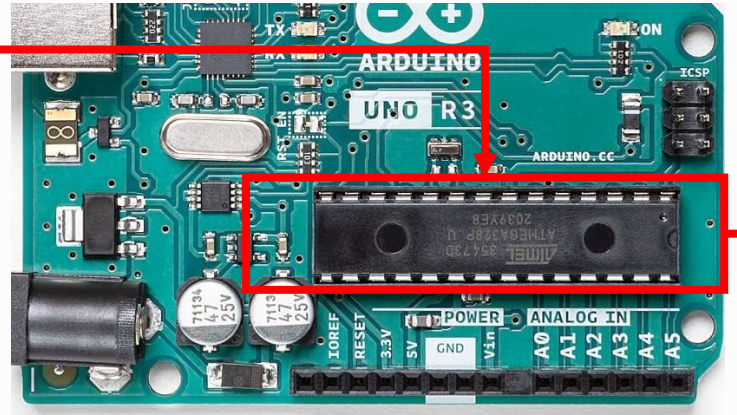
A Recap: Input/Outputs

- The inputs, or inputs, are usually electronic or mechanical sensors that convert signals (in the form of temperature, pressure, humidity, contact, light, movement, pH, etc.) from the physical world and convert them into current or voltage signals. Examples of inputs are photocells, potentiometers, gas, temperature, movement sensors, etc.
- Outputs are actuators or other devices that convert current or voltage signals into physically useful signals such as movement, light, sound, force or rotation. Examples of outputs are motors, LEDs or lighting systems, a buzzer that generates different tones, etc.

Input:
Receive signals from the physical world and convert them into current or voltage

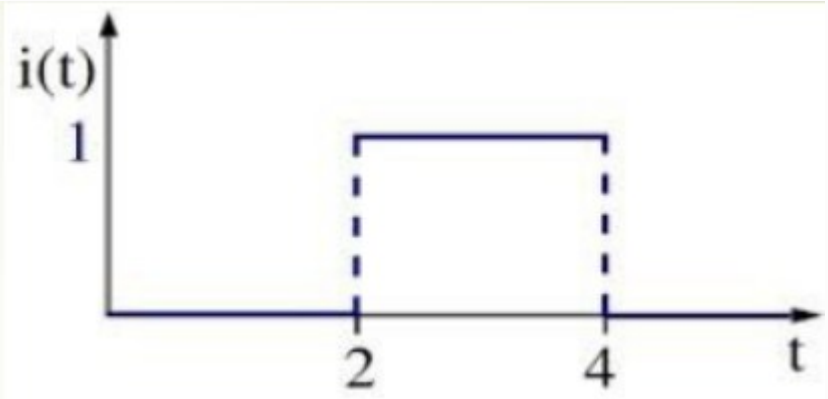
Processing:
Interpret and manipulate signals

Output:
Convert current or voltage into physically useful signals

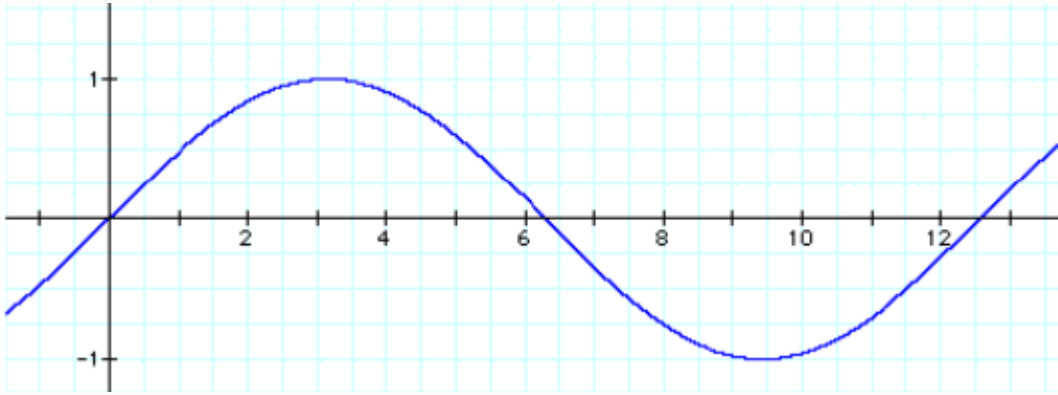


A Recap: Types of Signals

- Digital:
  Works according to Boolean principles: bit 0 represents 0V (volts), and bit 1 represents 5V (volts).



- Analogue:
  Time-varying signals (constantly take on different values)



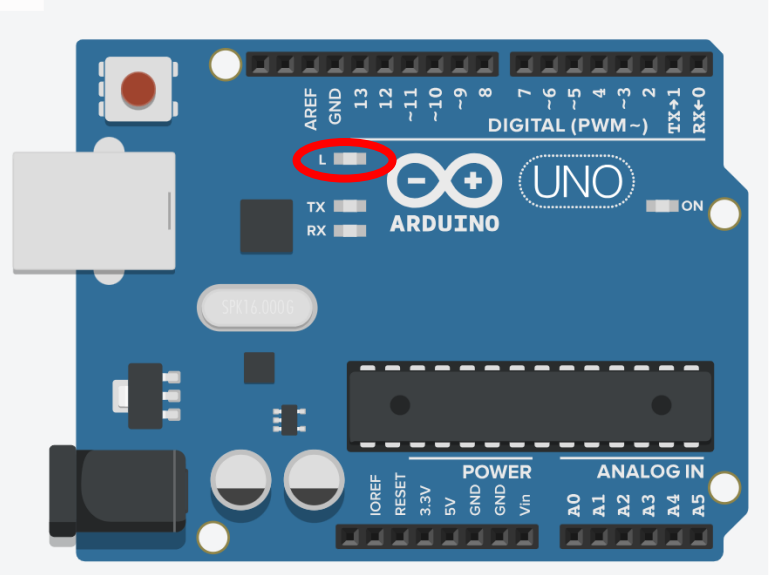
A Recap! Digital Outputs: Hello World

- This example shows the simplest experiment you can do with an Arduino to verify a physical output: blinking an LED. In this example, you will learn how to drive a load (built-in LED) connected to an Arduino pin.

What will I learn?

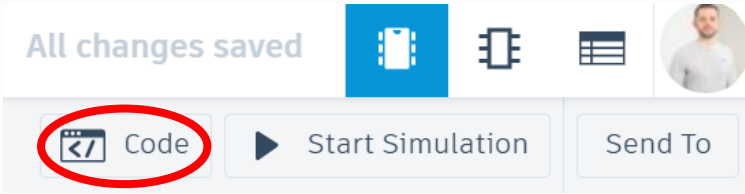
- Activate a digital output.
- Blink the on-board LED.
- The syntax of an Arduino Code.
- Use timers on an output pin.

Components

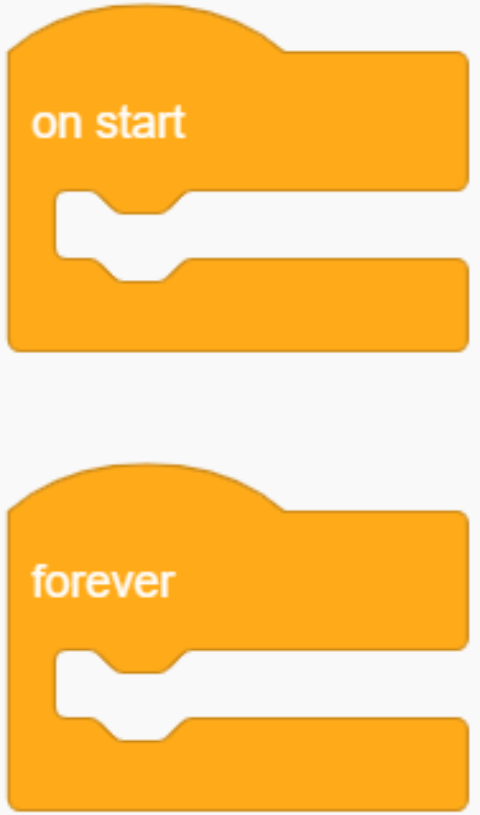


Digital Outputs: Hello World

The Code



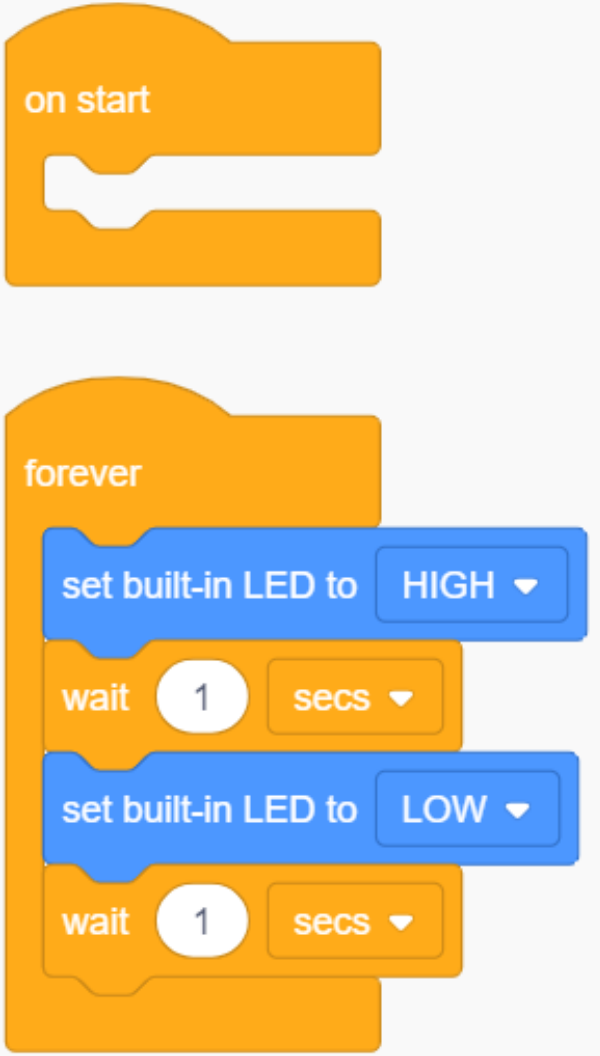
- Control Structures (Yellow):
  - *on start*: code structure that can be used to execute blocks of code only once, and when the board is powered. Use it for code that you only want to execute once when the board is turned on.
  - *forever*: code structure that can be used to execute code blocks repeatedly in an infinite loop while the board is powered. Use it for code that you want to be constantly executed while the board is turned on.

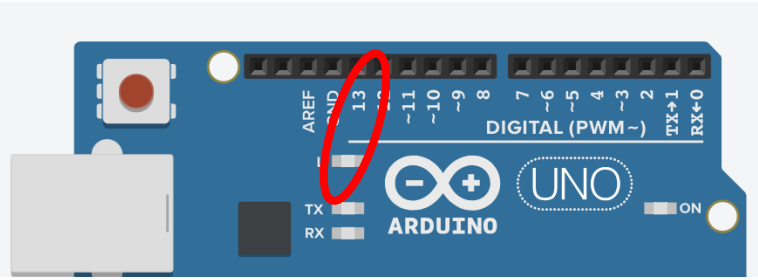


Hello World

The Code

- Control Commands (Yellow):
  - *wait*: this code block activates a timer. The microcontroller waits for the defined time to be over before executing the next code block. For this example, the timer was set to 1 second.
- Output Commands (Blue):
  - *set built-in LED*: define the logical value on that pin. The pin will work as an output, controlling anything connected to it. Built-in LED is the name given to pin 13.

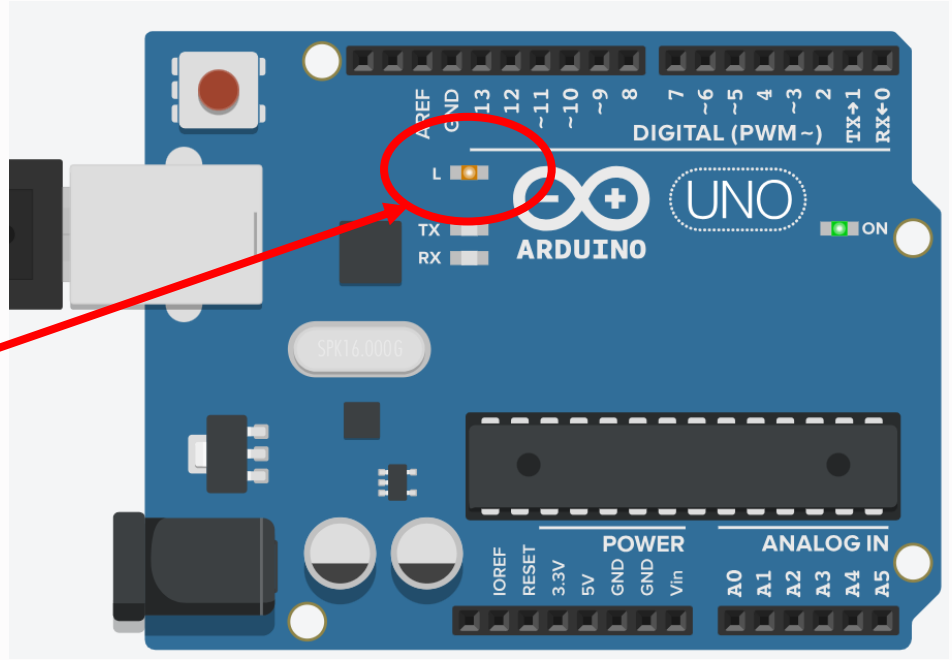


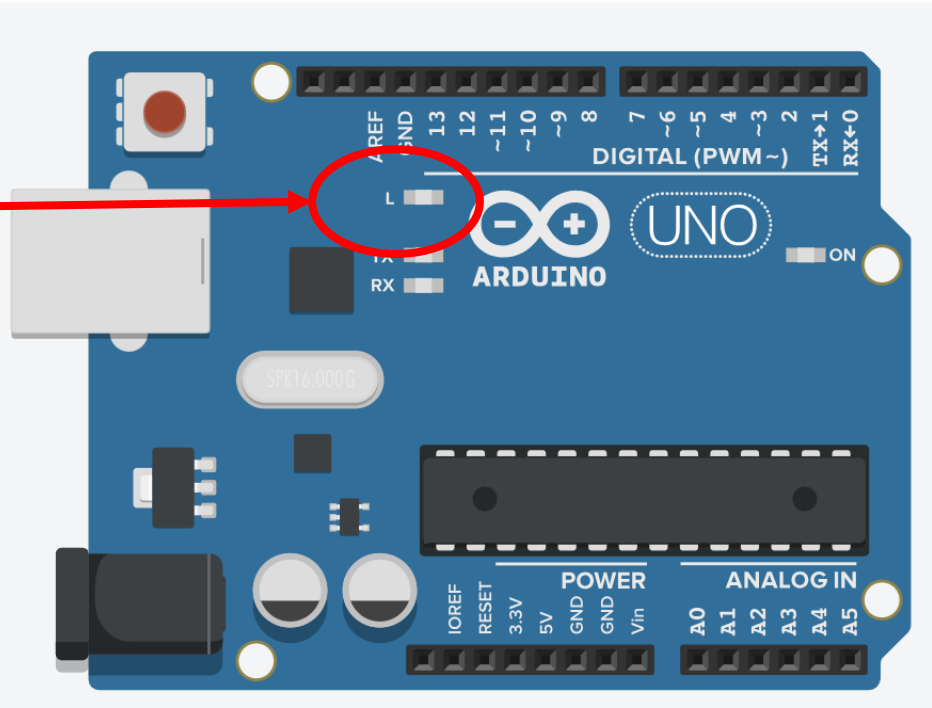


Hello World

The Code

The “set” command is used to write on the pin, i.e., to send a logical value to that pin.





HIGH is equivalent to bit 1 (5V), while **LOW** is equivalent to bit 0 (0V).

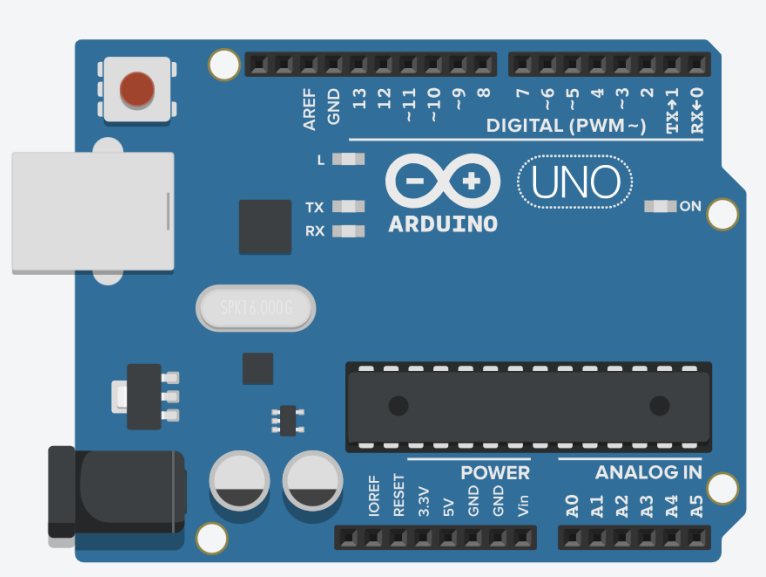
Hello World – External LED

- This example is similar to the previous one: blinking an LED. In this example, you will learn how to drive a load (an external LED) connected to an Arduino pin.

What will I learn?

- Activate a digital output.
- Blink an LED external to the board.
- The syntax of an Arduino Code.
- Use timers on an output pin.

Components

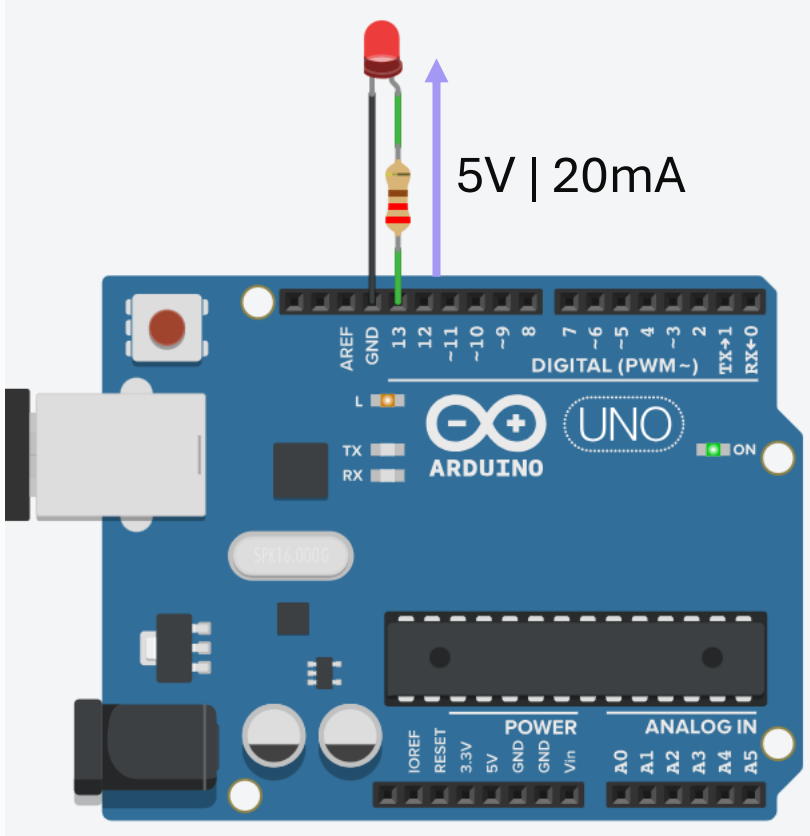


Resistor

LED

Hello World – External LED

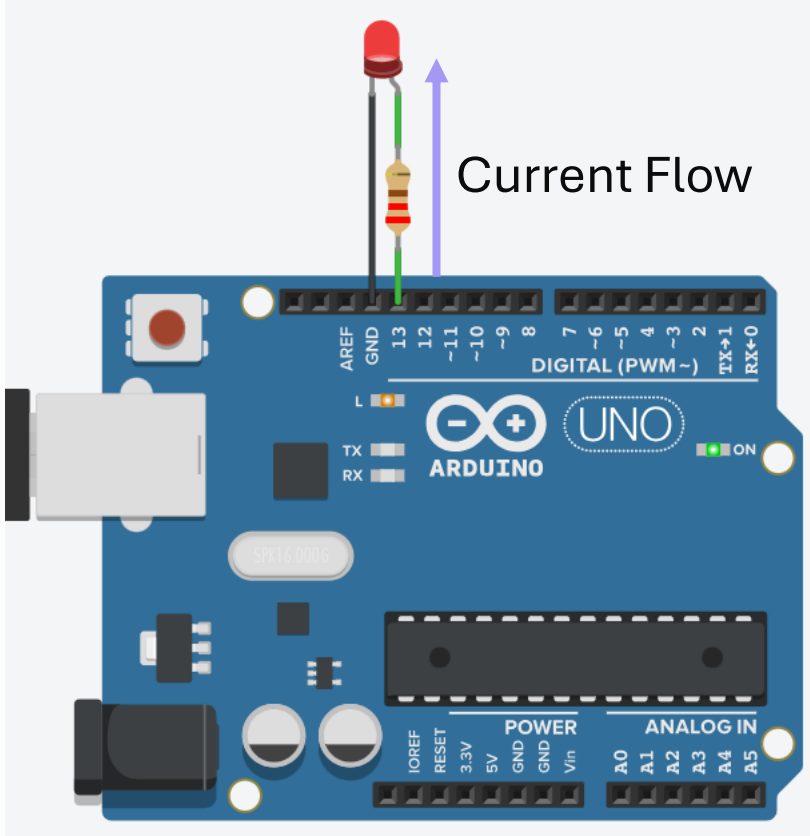
The Circuit



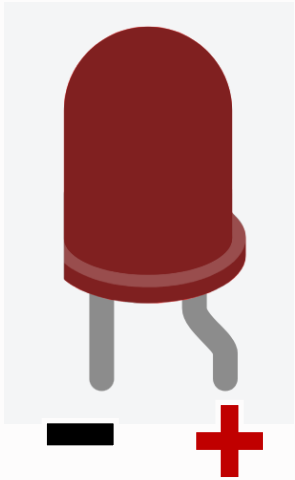
- When an Arduino pin is used as an output, it supplies the load connected to it a total of 5V and 20mA.
- This is enough to damage an LED. Therefore, a resistor is necessary to limit the amount of current that goes to the LED.
- Hence, a resistor was connected between the output pin 13, the LED and the ground (GND), reducing the current (I) from its maximum value ($I < 20$).

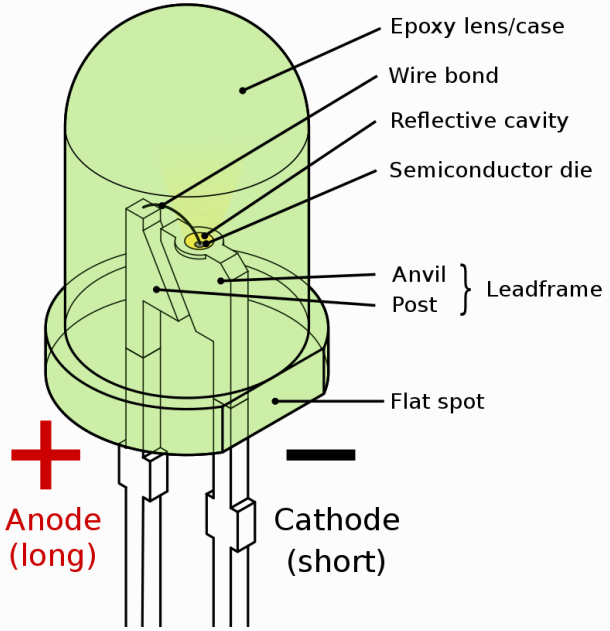
LED (Light-Emitting Diode)

The Circuit



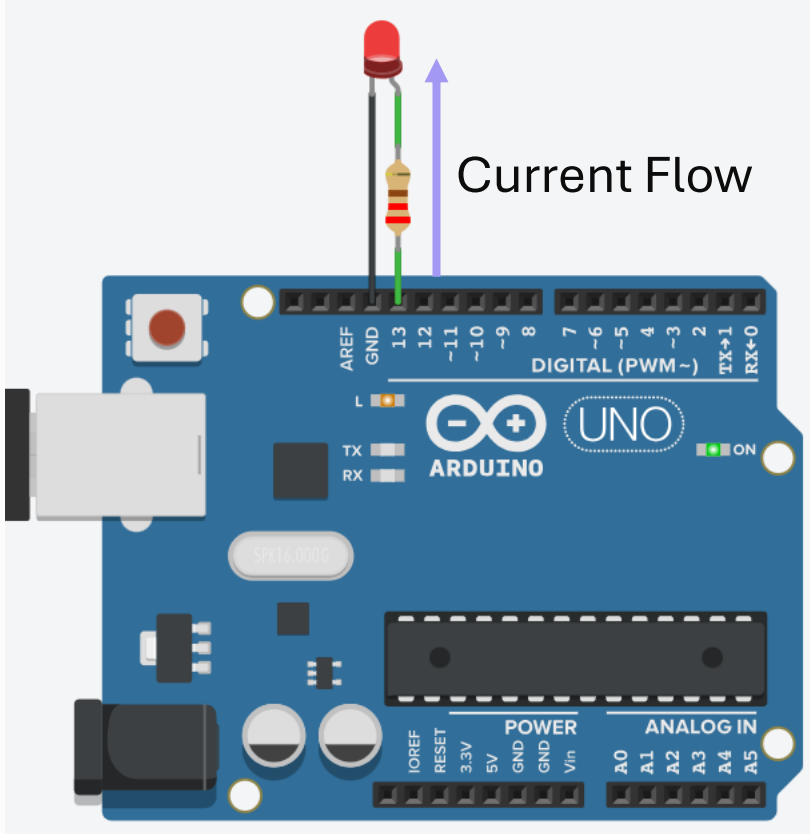
- A LED (light-emitting diode) is a semiconductor device that emits light when current flows through it.
- The LED terminals are called: anode (+) and cathode (-).



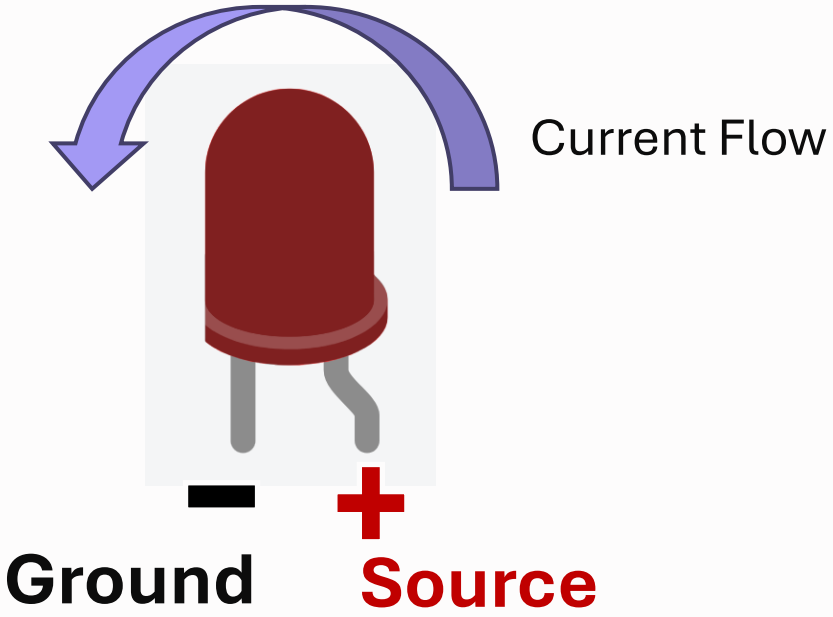


LED (Light-Emitting Diode)

The Circuit

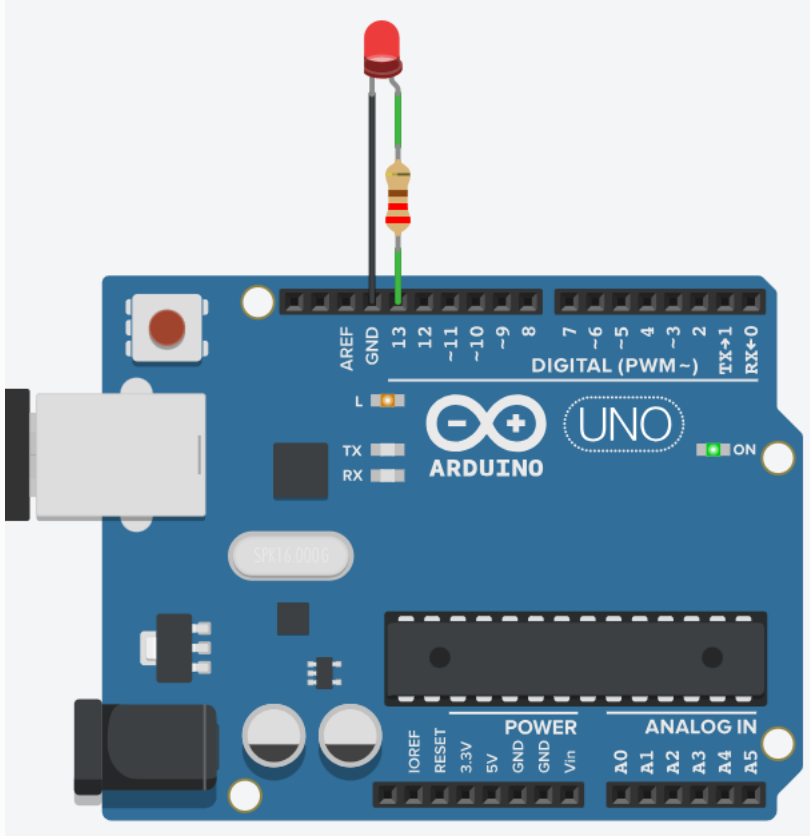


- The LED has a polarity, that is, a connection order. Changing this order will make it not work properly.
- Hence, for it to work, the current must flow from the anode (+) to the cathode (-).



Resistor

The Circuit



- A resistor is a passive two-terminal electrical component that implements electrical resistance.
- In electronic circuits, resistors are used to reduce current flow and divide voltages, among other uses.
- Ohm's law states that the resistance can be defined based on the voltage (V) and current (I), as follows:

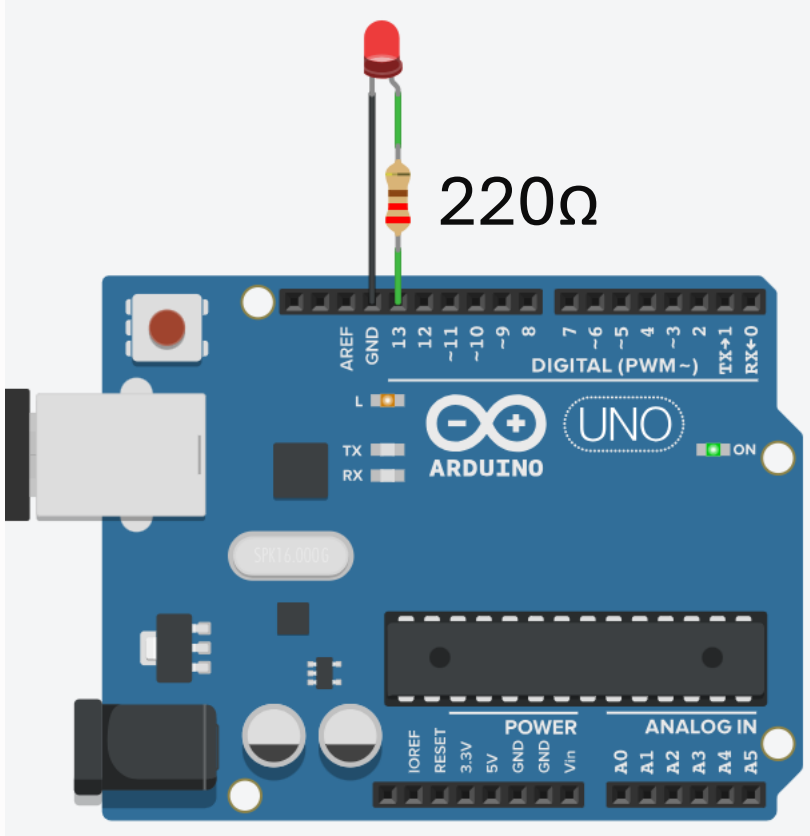
$$R = \frac{V}{I}\Omega$$

Resistor - Wikipedia

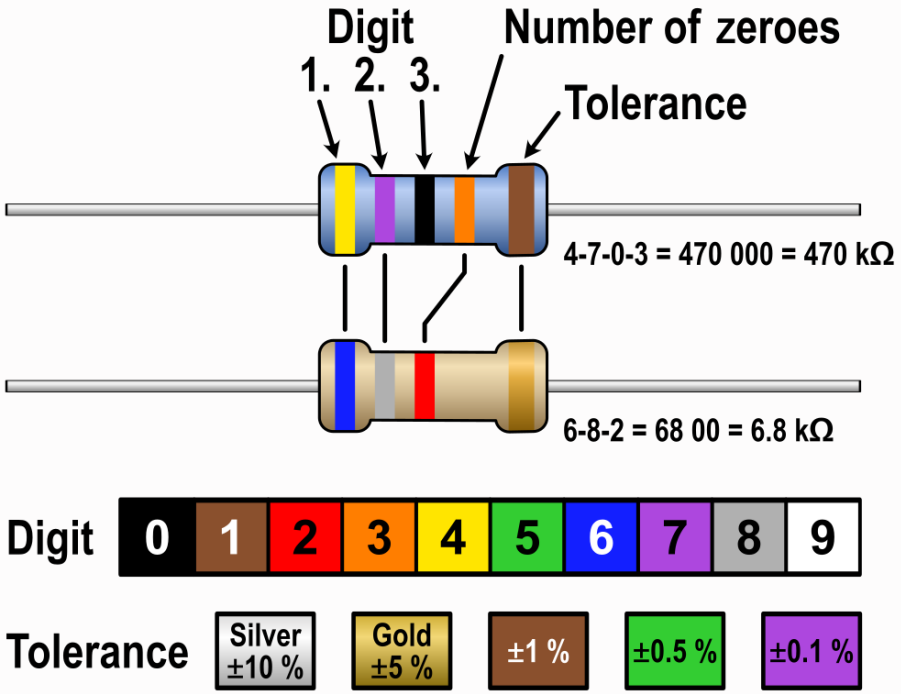
Electronic colour code - Wikipedia

Resistor

The Circuit



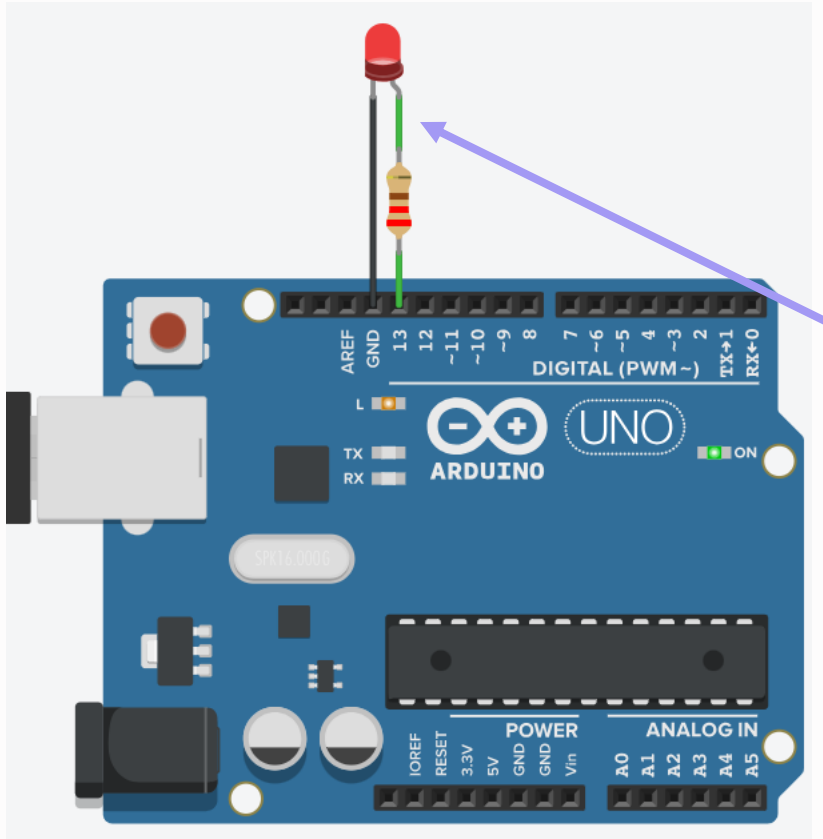
- The resistance value of a resistor can be defined based on its colour code:
- To change its value on Tinkercad, click on the component and modify its value in the pop-up box.

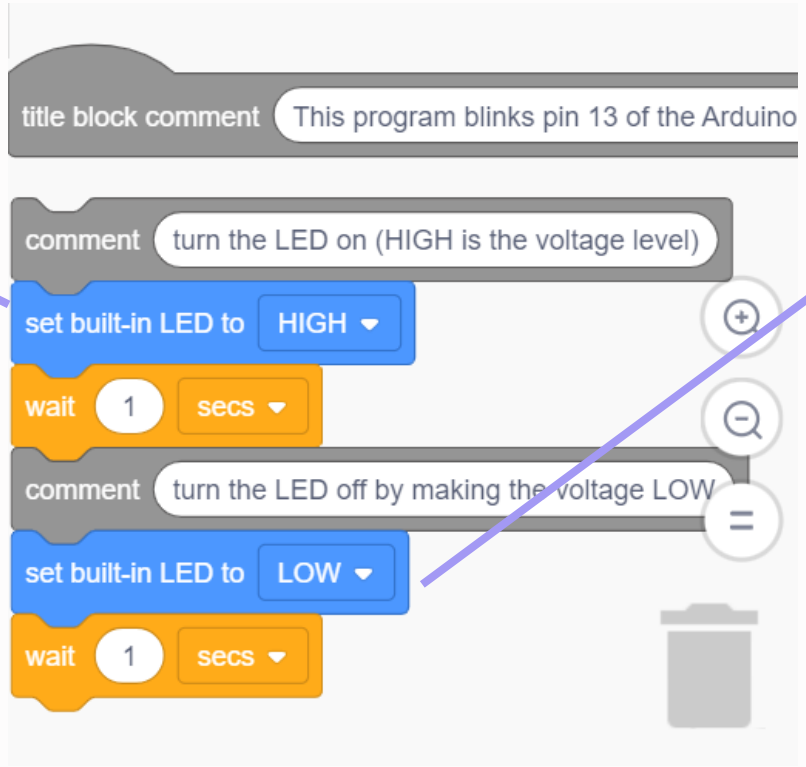


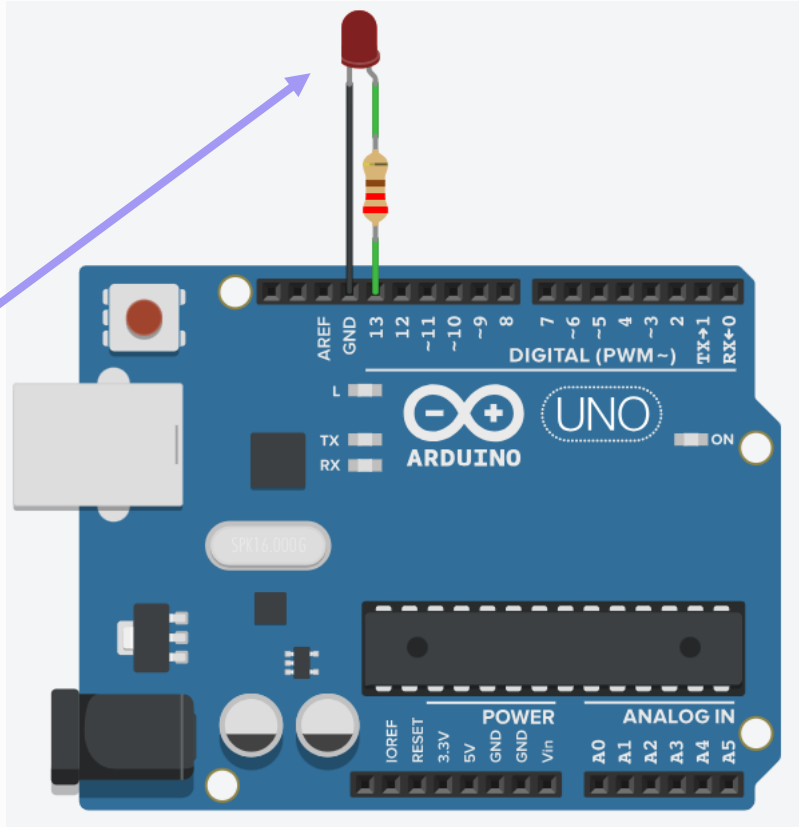
Hello World – External LED

The Code

- Note that the program executes repeatedly despite not using the forever block.



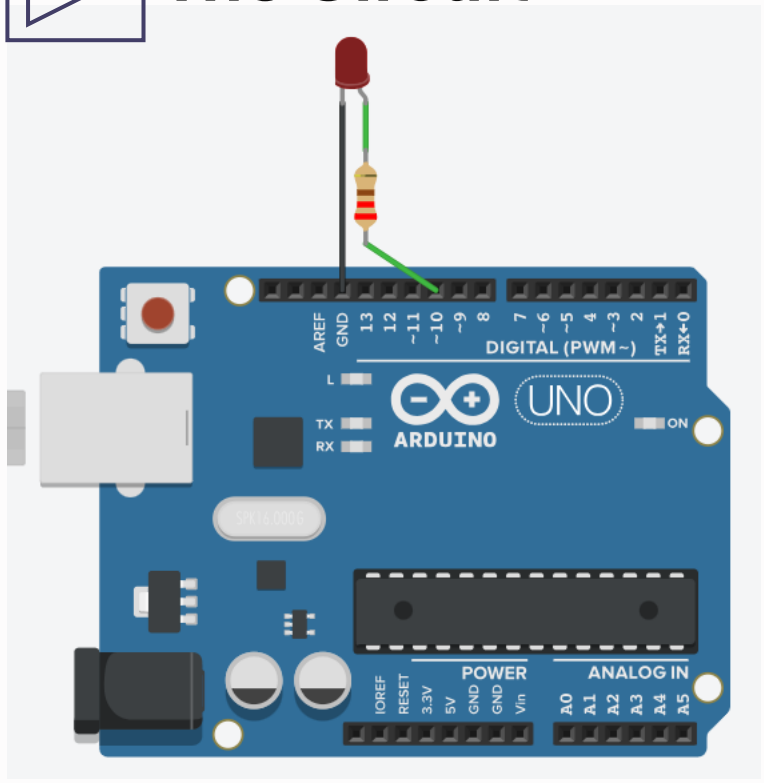




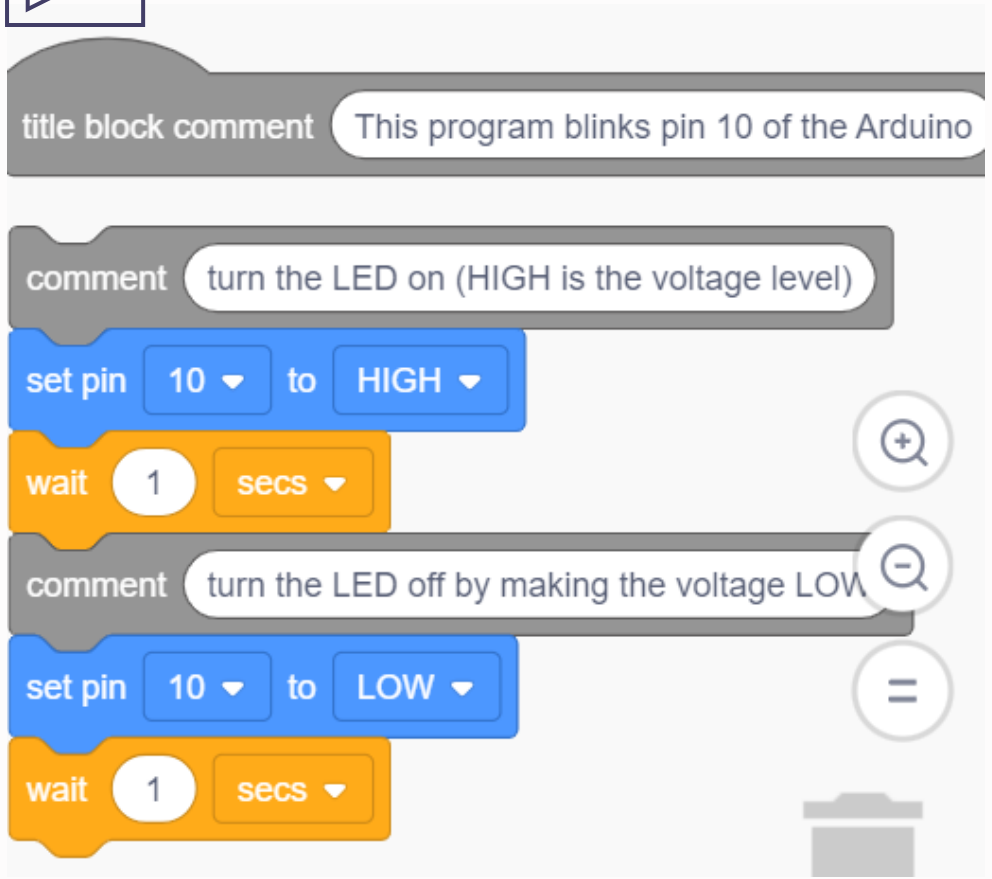
External LED in different Digital pins

The LED can be connected to other digital pins. For example, let's replace pin 13 in the previous example to pin 10.

The Circuit



The Code



Practice Questions

1. Modify the Hello World with an External LED example, as follows:

a) Make the LED stay 3 seconds ON and 3 seconds OFF.

b) Make the LED stay 200 milliseconds ON and 500 milliseconds OFF.

2. Again, from Hello World with an External LED example, make the necessary changes in the code and circuit so that the LED is placed on Pin 5, and stays 2 seconds ON and 1 second OFF.

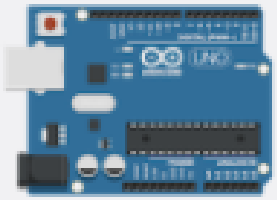
Multiple LEDs

- In this example, you will learn how to drive three loads (LEDs) connected to an Arduino pin.

What will I learn?

- Activate multiple outputs simultaneously.
- Use a breadboard.
- Use Arduino source and ground pins

Components



Arduino Uno R3

Breadboard Small

Resistor

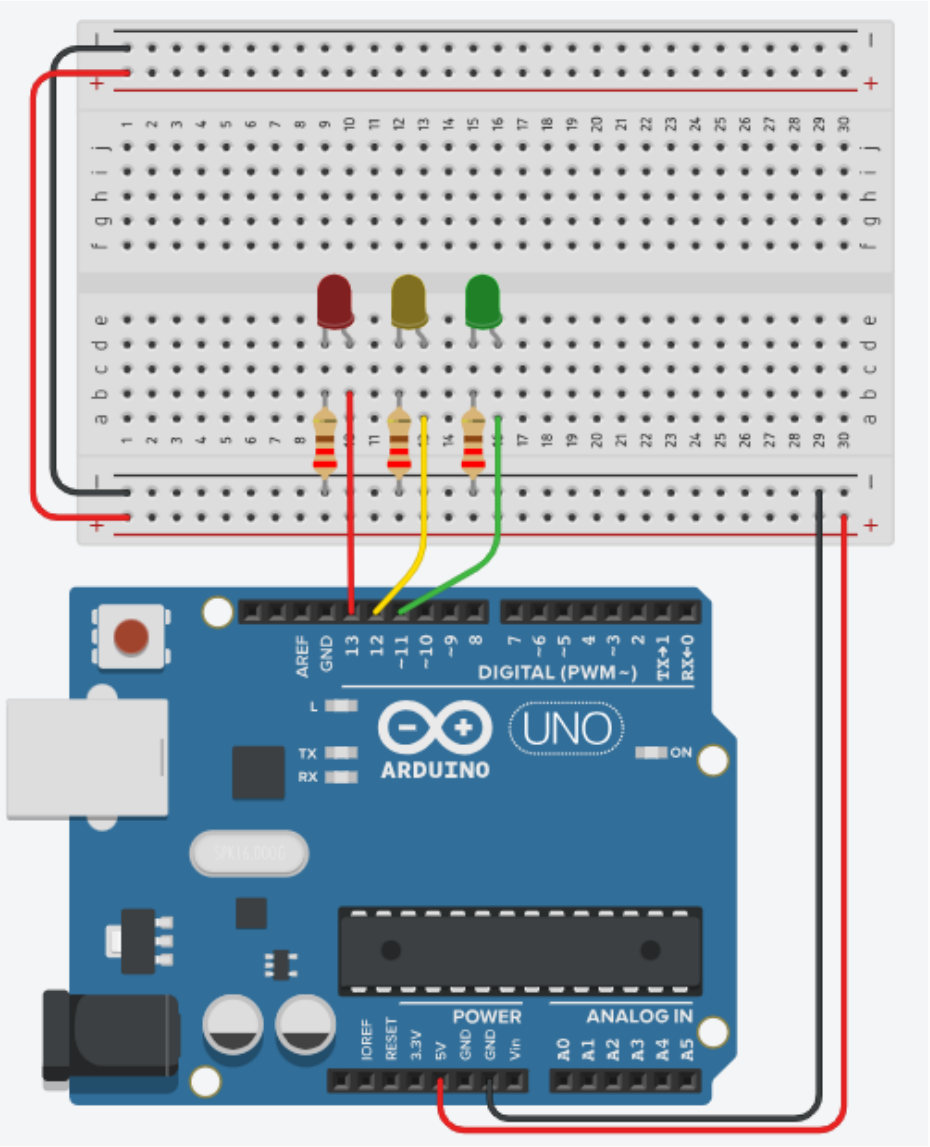
LED

Multiple LEDs

The Circuit

- You will need:
- 1 Arduino Uno board.
- 3 LEDs (Click on it the change its colour).
- 2 Resistors (220Ω).
- 1 Breadboard.

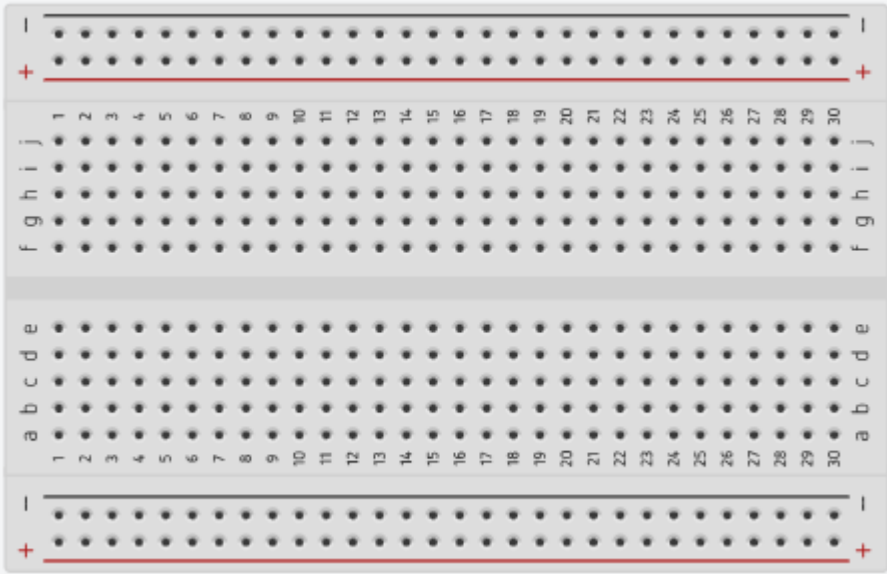
- Connect all components like in the following image:

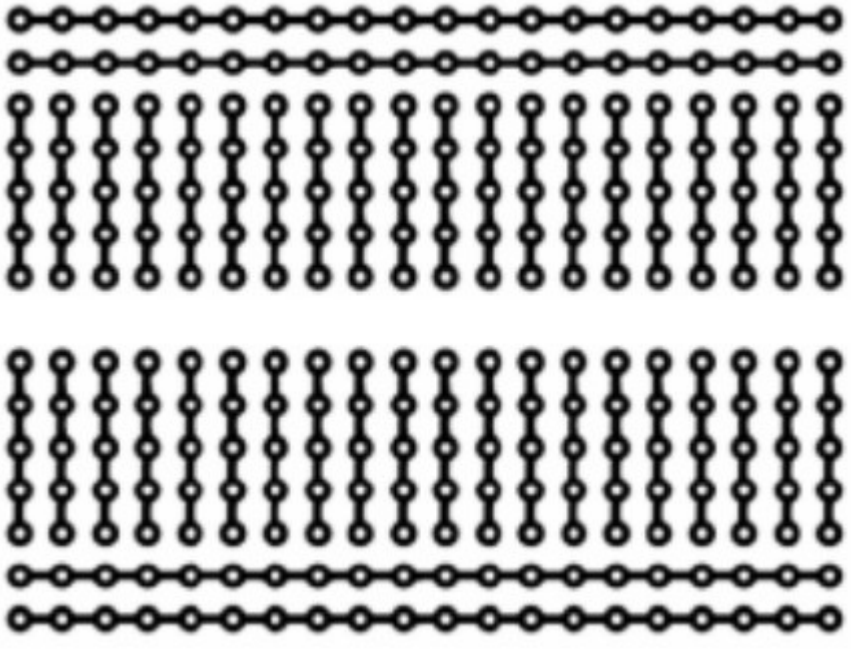


Breadboard

The Circuit

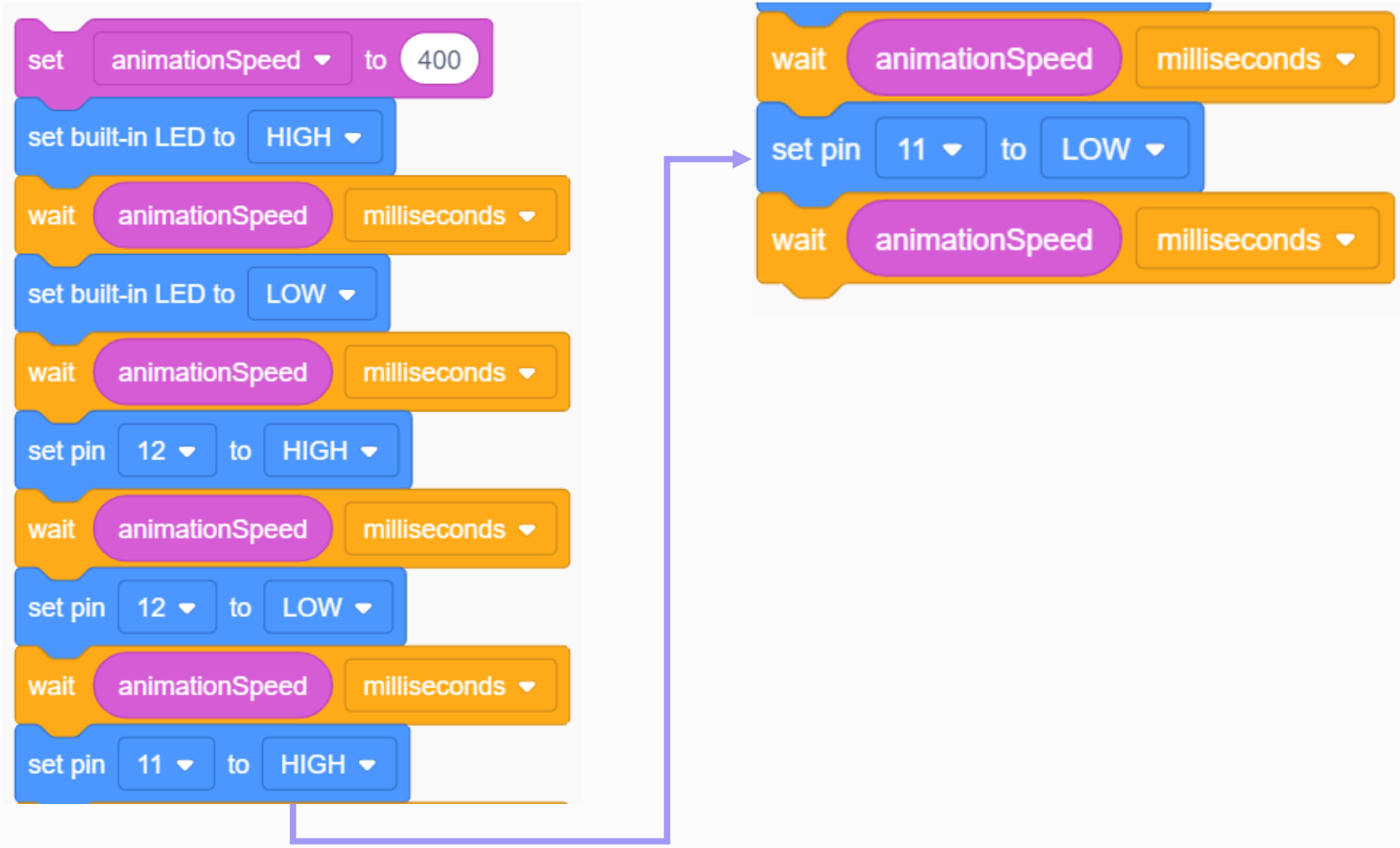
- It is a reusable board used to build electronic circuit prototypes without the need for soldering. Also called protoboard, it is made of perforated plastic blocks and several thin strips of copper metal alloy.

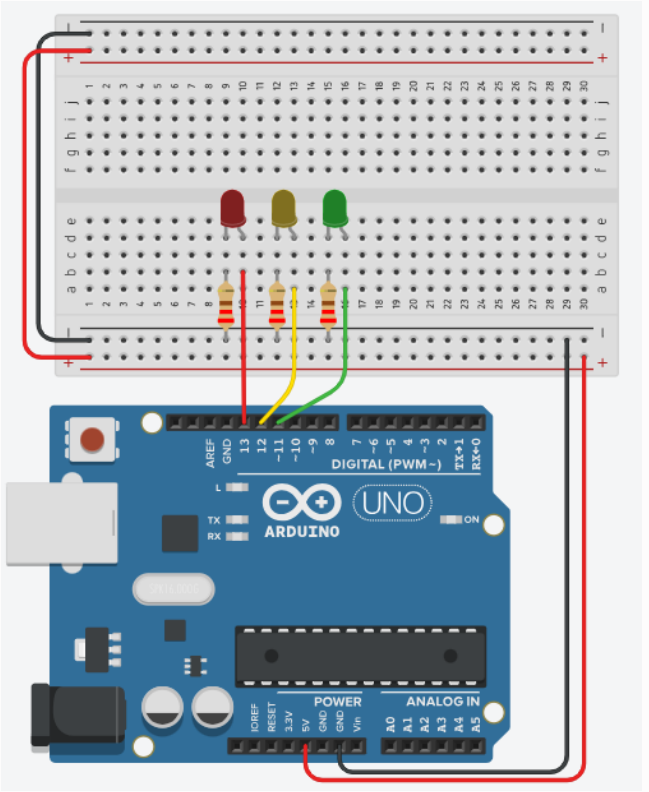




Multiple LEDs

The Code





Practice Questions

1. Create a Two-Way Traffic Light System. For this project, you need to create two different traffic lights (using three LEDs for each), and they should work as follows:

Traffic Light System 1:

1. Red LED ON;
2. Red LED ON;
3. Red LED ON;
4. Red LED ON;
5. Red LED ON for 0.5 seconds;
6. Yellow LED ON for 1 second;
7. Green LED ON for 4 seconds;
8. Yellow LED ON for 1 second;
9. Repeat.

Traffic Light System 2:

1. Red LED ON for 0.5 seconds;
2. Yellow LED ON for 1 second;
3. Green LED ON for 4 seconds;
4. Yellow LED ON for 1 second;
5. Red LED ON;
6. Red LED ON;
7. Red LED ON;
8. Red LED ON;
9. Repeat.

Practice Questions

2. A set of 4 LEDs connected to pins 2 to 5 will be activated sequentially when powering the Arduino. The LEDs will turn off after 2 seconds. And the process restarted 0.5 seconds later. Create variables to define and hold the waiting time value.

3. Make a sequence of six LEDs light up as follows: those present at BOTH ends of the sequence should already be ON when powering the Arduino. Then, sequentially turn the lights ON, one towards the other, giving the impression that they are meeting. Then, return to the ends giving the impression that they are moving away.

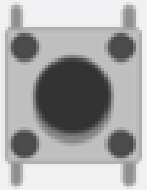
Digital Inputs – Push Button

- The button is a component that connects two circuit points when pressed. In this example, the LED is ON while the button is pressed.

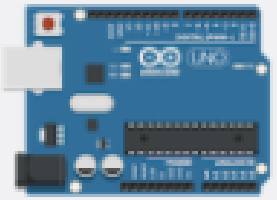
What will I learn?

- Read a digital input.
- Use If conditionals.
- Use pushbuttons.
- Pull Up setting.

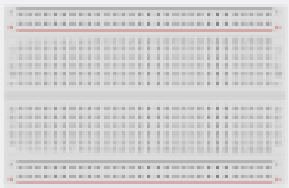
Components



Pushbutton



Arduino Uno R3



Breadboard Small

Resistor

LED

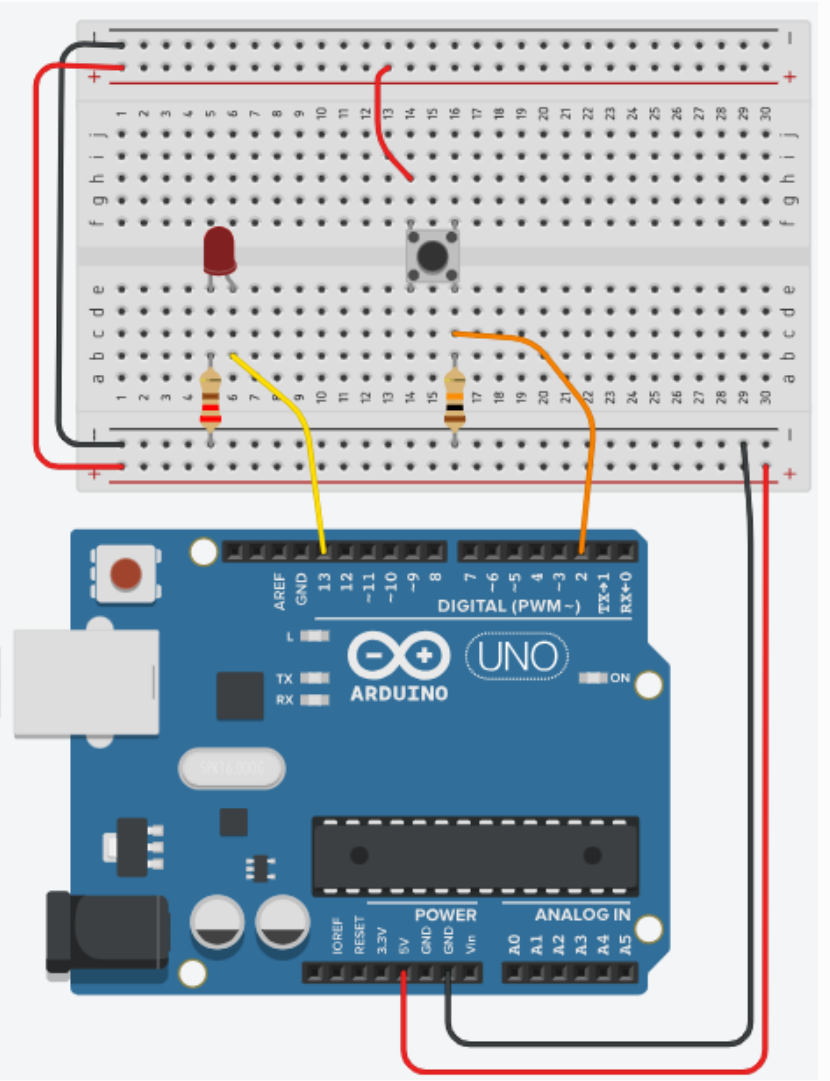
Digital Inputs – Push Button

The Circuit

You will need:

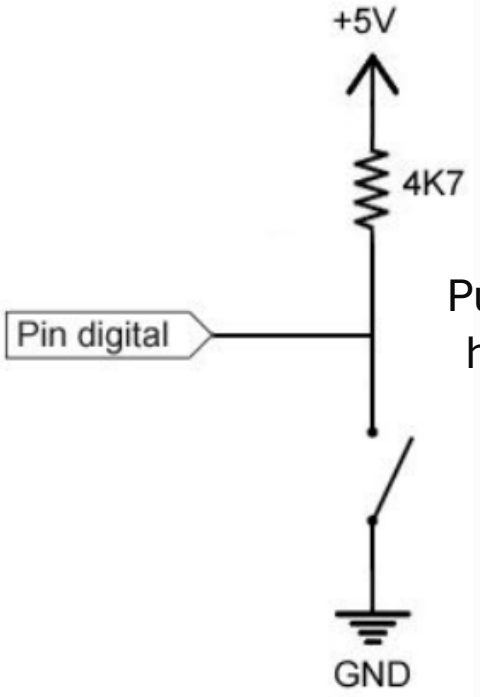
- 1 Arduino Uno board.
- 1 LED.
- 2 Resistors (220Ω and 10kΩ).
- 1 Breadboard.

Connect all components like in the following image:



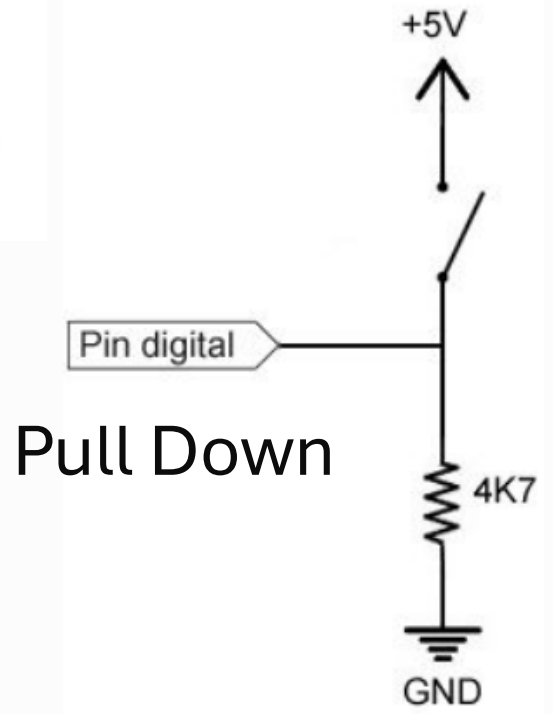
Pull Up and Pull Down

- Normally, when you connect pushbuttons to digital pins on a microcontroller, the pin switches to a floating-point state when the pushbutton is open.
- Floating point: a state is assumed at random. In the case of Arduino, it normally assumes the high level (bit 1).
- Resistors are employed in order to avoid the floating point, fixing a voltage value applied under a pin even when the button is open. When a resistor is connected to the power source (setting the pin high), it is called Pull Up. If it is connected to the ground (setting the pin at a low level), it is called Pull Down.



Pull Up

Pull Up resistors should have high values, e.g., above 10k Ω.

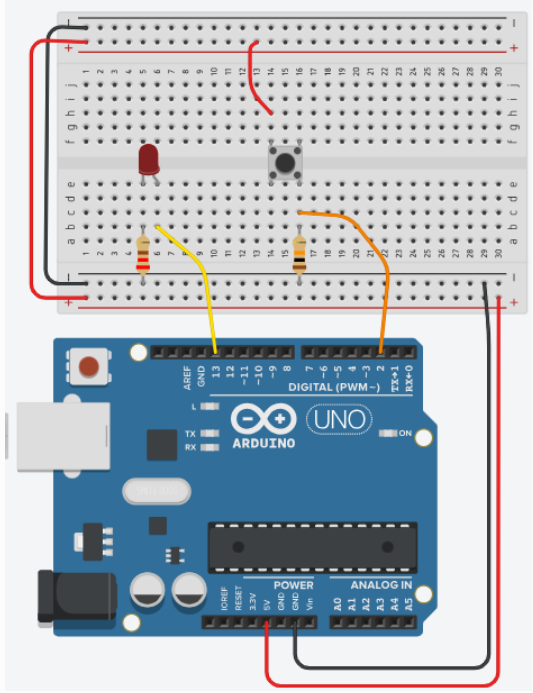


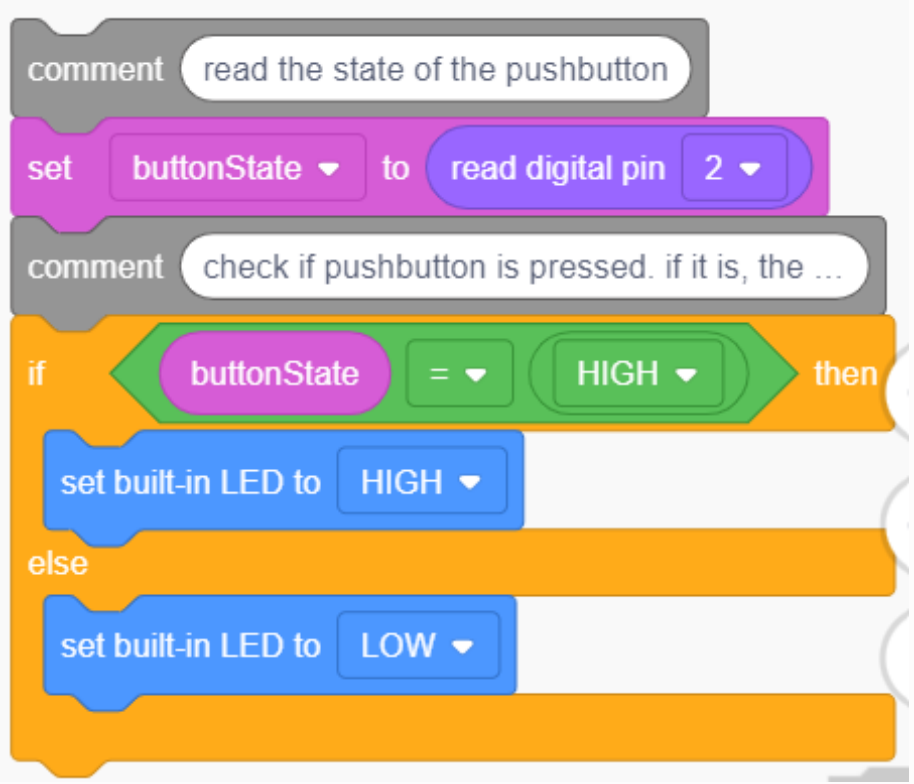
Pull Down

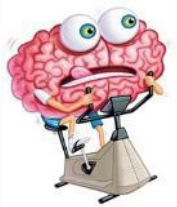
Digital Inputs – Push Button

The Code

- *if conditional*: enables to execute different commands based on a condition.
- *set variable (Pink)*: modify the value of a variable.
- *read digital pin Input Command (Purple)*: read the value the pin receives.







Practice Questions

1. Button and LED: When pressing a button connected to pin 8, an LED connected to pin 10 should turn ON in such a way that it should remain ON for 5 seconds, and after that time, the LED will turn OFF.
2. LED Bar: A set of 4 LEDs connected to pins 2 to 5 will be activated sequentially after button 1 (connected to pin 10) is pressed.
3. LED Bar: A set of 4 LEDs will activate sequentially after button 1 is pressed. The LEDs will be deactivated by a button 2.
4. Create a circuit with one LED and two buttons. The LED should be connected to pin 13, and the buttons to any digital pin you choose. Your system should turn ON the LED when pressing buttons 1 or 2.
5. Create a circuit with two LEDs and two buttons. Each LED should be turned ON by one of the buttons.

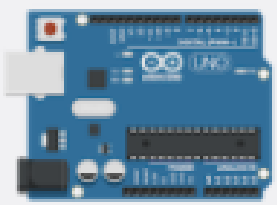
Analogue Inputs

- In this example, you will learn how to read an analogue input. The value read from the input will be used to control the blinking time of the onboard LED using a potentiometer. The potentiometer is a resistor with a varying resistance.

What will I learn?

- Read an analogue input.
- Use a potentiometer.

Components



Arduino Uno R3

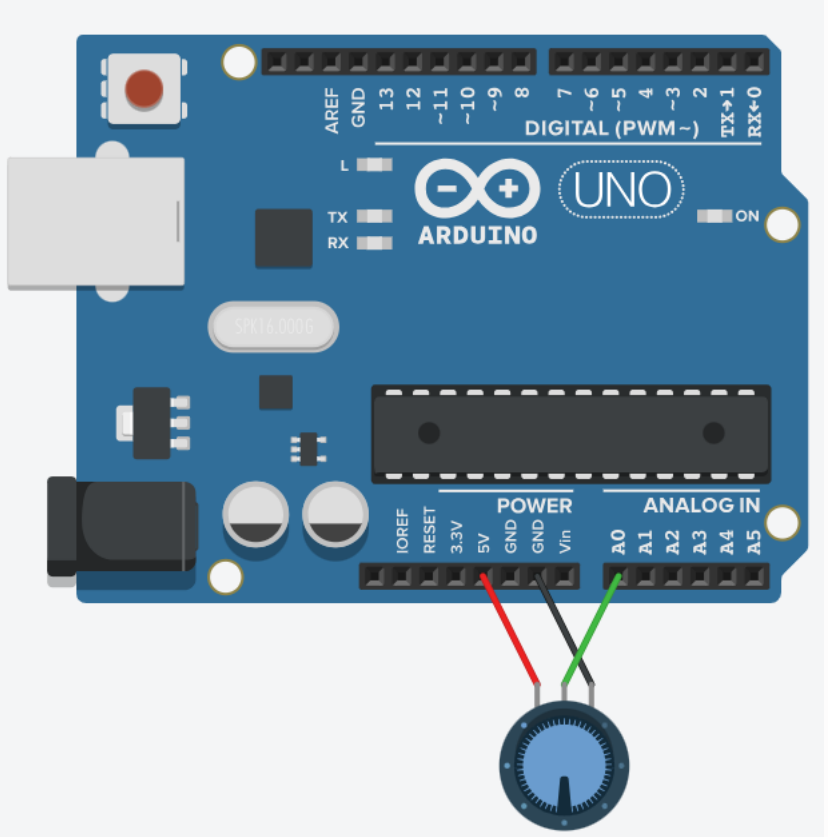
Potentiometer

Analogue Inputs

The Circuit

- A potentiometer is a resistor whose value is variable, suitable for use as a control element in electronic devices. The resistance between the centre and end terminals varies according to the position of the centre control switch.

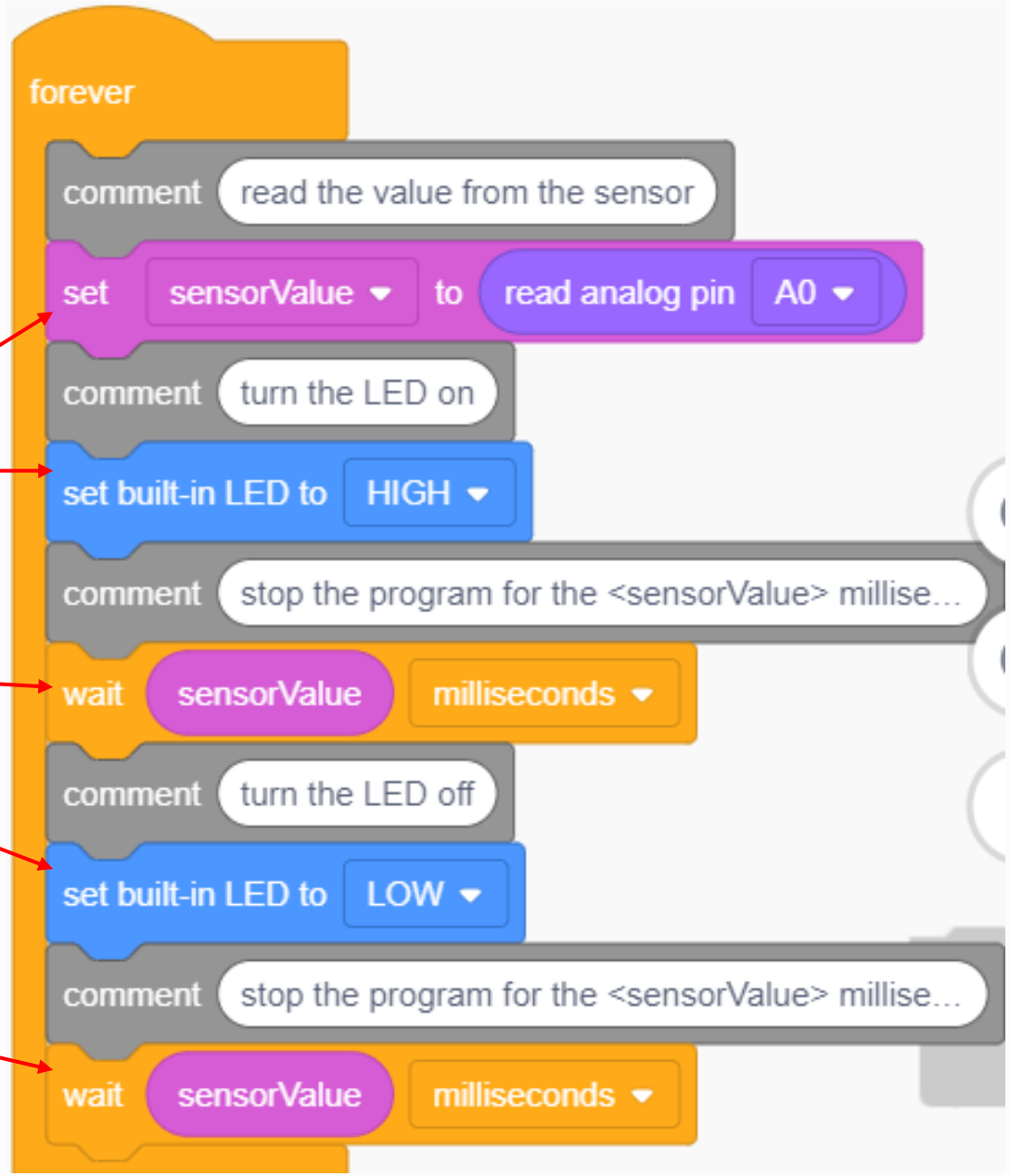
- Note that the end terminals are connected to the source (5V) and ground (GND), while the central terminal is connected to an Arduino analogue pin.



Analogue Inputs

The Code

- Initially, the analogue pin is read, and its value is stored in the variable "*sensorValue*."
- The onboard LED (pin 13) turns ON.
- Pause the code for "*sensorValue*" milliseconds. This pause allows us to continue seeing the LED ON.
- The onboard LED (pin 13) turns OFF.
- Pause the code for "*sensorValue*" milliseconds. This pause allows us to continue seeing the LED OFF.
- Repeat.



Analogue to Digital Converter (ADC)

- The Arduino UNO has 1 ADC to convert the signals inserted in pins A0 to A5. This ADC has a 10-bit resolution and accepts voltages ranging from 0V to 5V (it does not accept negative values).

- The voltage values read will be converted into a range from 0 to 1023, as $2^{10} = 1024$. Thus:

$$5V - 1023$$

- You can use the value read by the Arduino to determine the voltage connected to the pin. For instance, if the value read is 409, the voltage connect to pin is approximately:

$$\frac{5 \times \mathbf{409}}{1023} \cong 2V$$

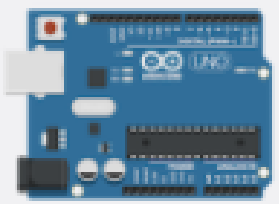
Serial Monitor

- In this example, you will learn how to display data using the serial monitor. As the name suggests, the serial monitors the data transferred between the Arduino and the computer.

What will I learn?

- Use the serial monitor.

Components

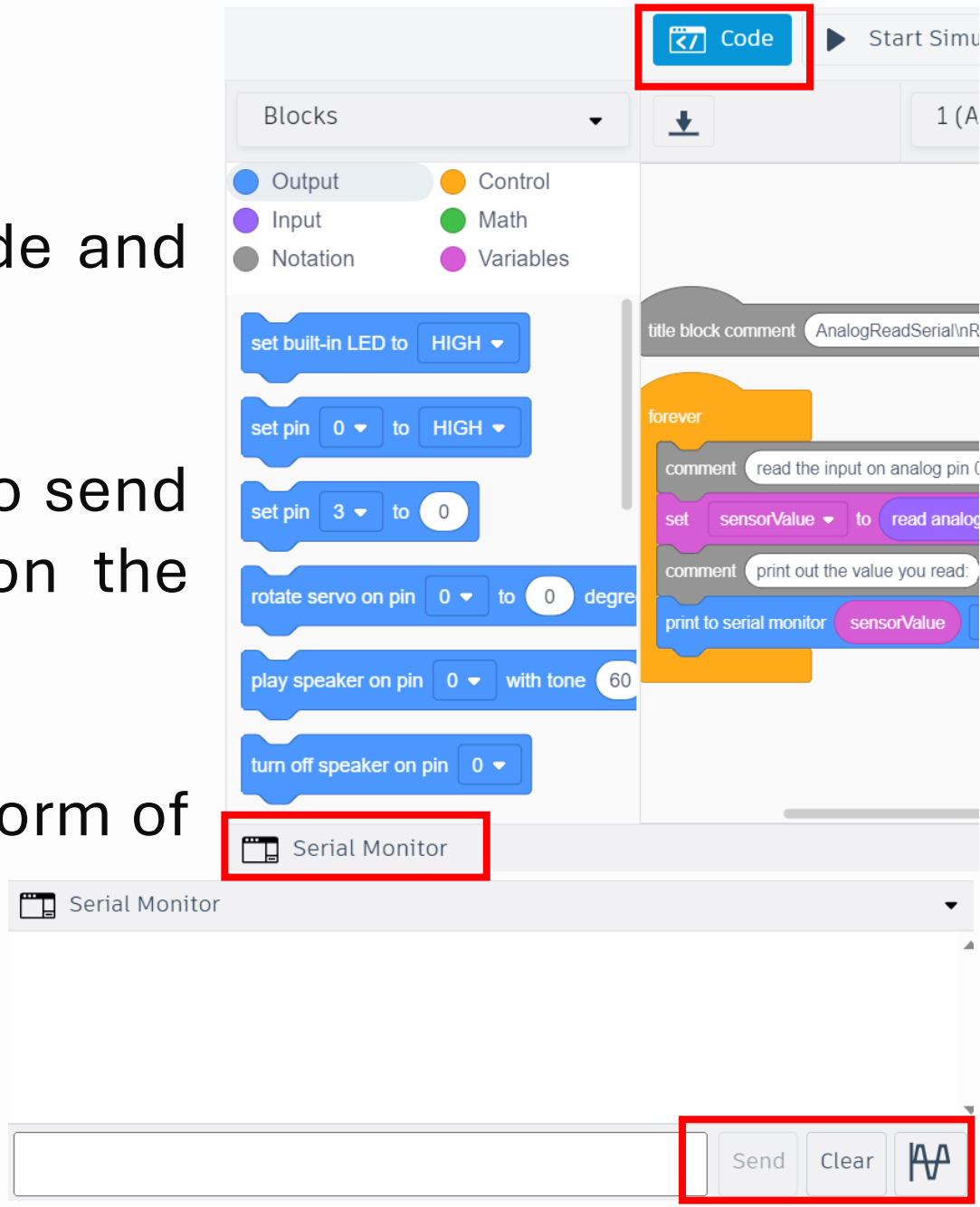


Arduino Uno R3

Potentiometer

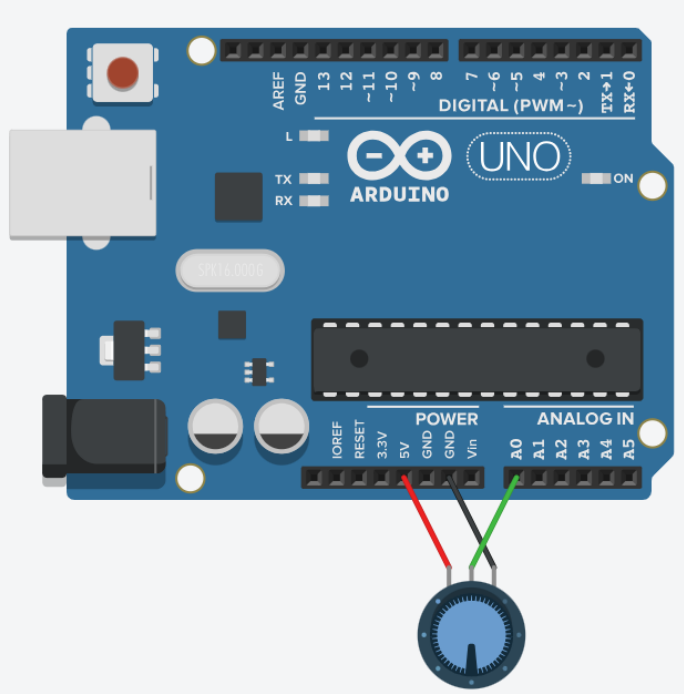
Serial Monitor

- To access the serial monitor, click on Code and Serial Monitor.
- You can use the Send and Clear buttons to send information and delete all data shown on the monitor.
- The graph button shows the values in the form of a graph.

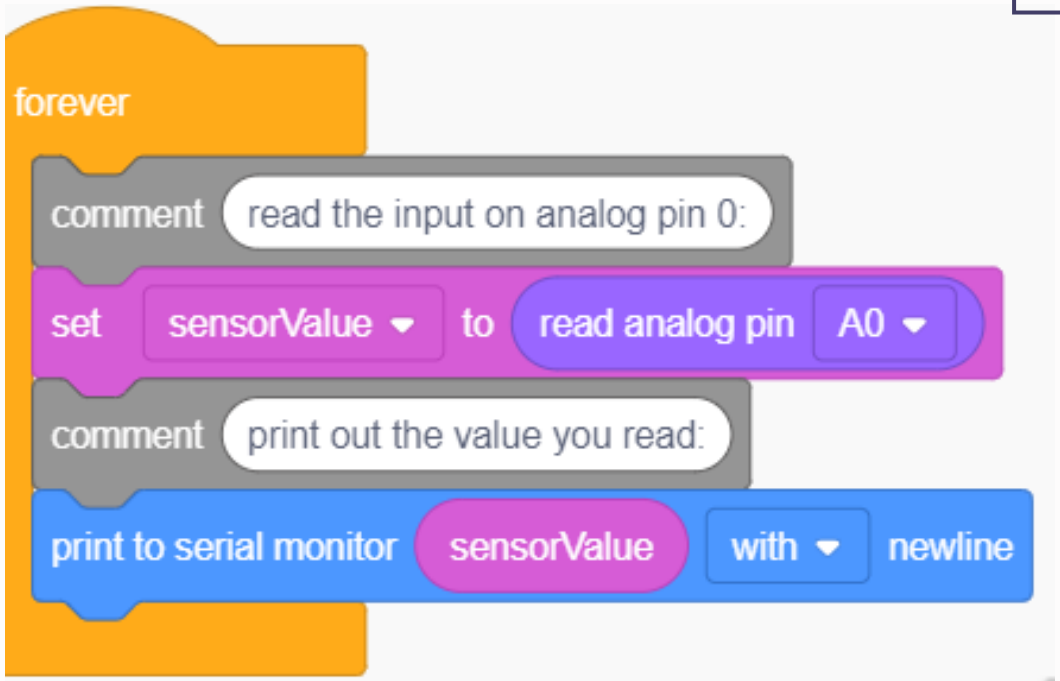


Serial Monitor

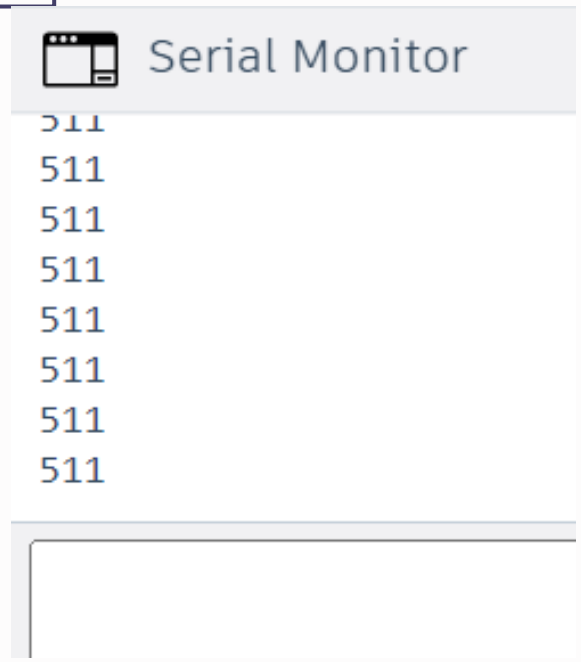
The Circuit



The Code



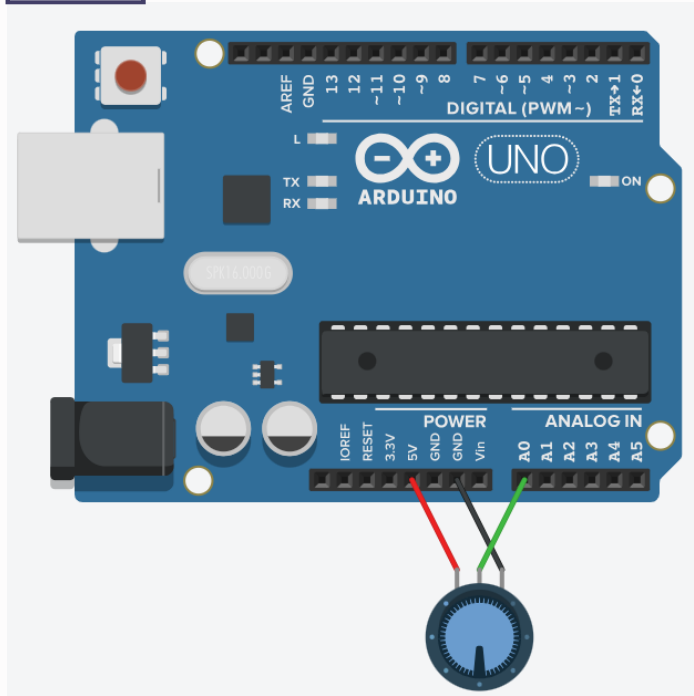
The Monitor



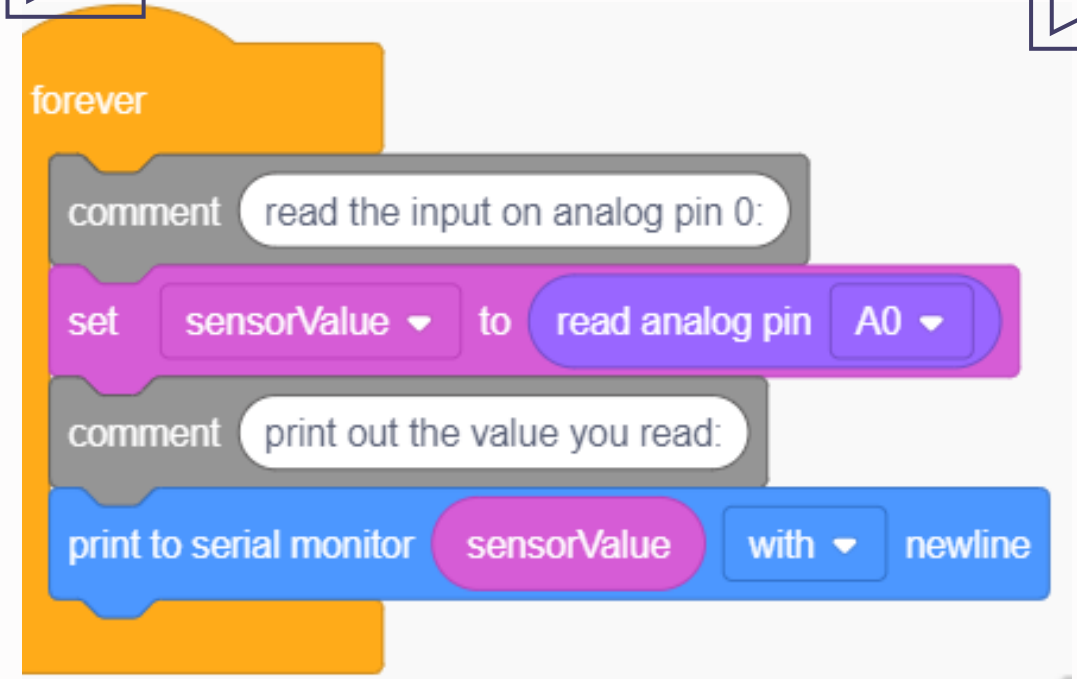
- *print to serial monitor Output command (blue)*: enables sending data to the serial monitor. The newline option skips to the next line after printing the data.

Serial Monitor

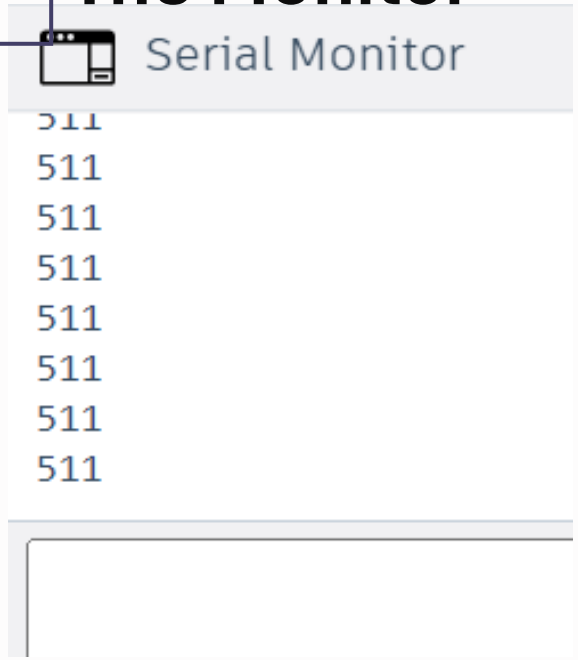
The Circuit



The Code



The Monitor



- As you rotate the central terminal of the potentiometer towards 5V, the value approaches 1023. Meanwhile, as you rotate it towards the GND, the value approaches 0.

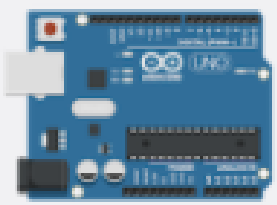
Analogue Outputs

- In this example, you will learn how to write an analogue output. For the Arduino Uno, an analogue output means using a Pulse Width Modulation (PMW). For this purpose, only digital pins with the tilde symbol can be used.

What will I learn?

- Write an analogue output.
- Learn the count code block.

Components



Arduino Uno R3

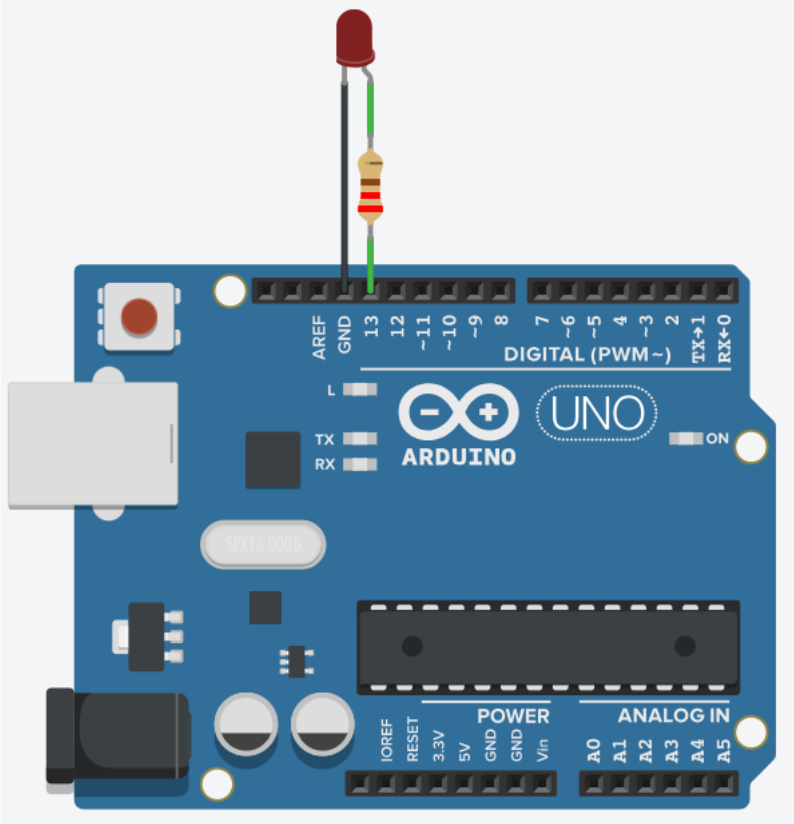
Resistor

LED

Analogue Outputs

The Circuit

- This is the same circuit we have seen for the Hello World with an External LED example.

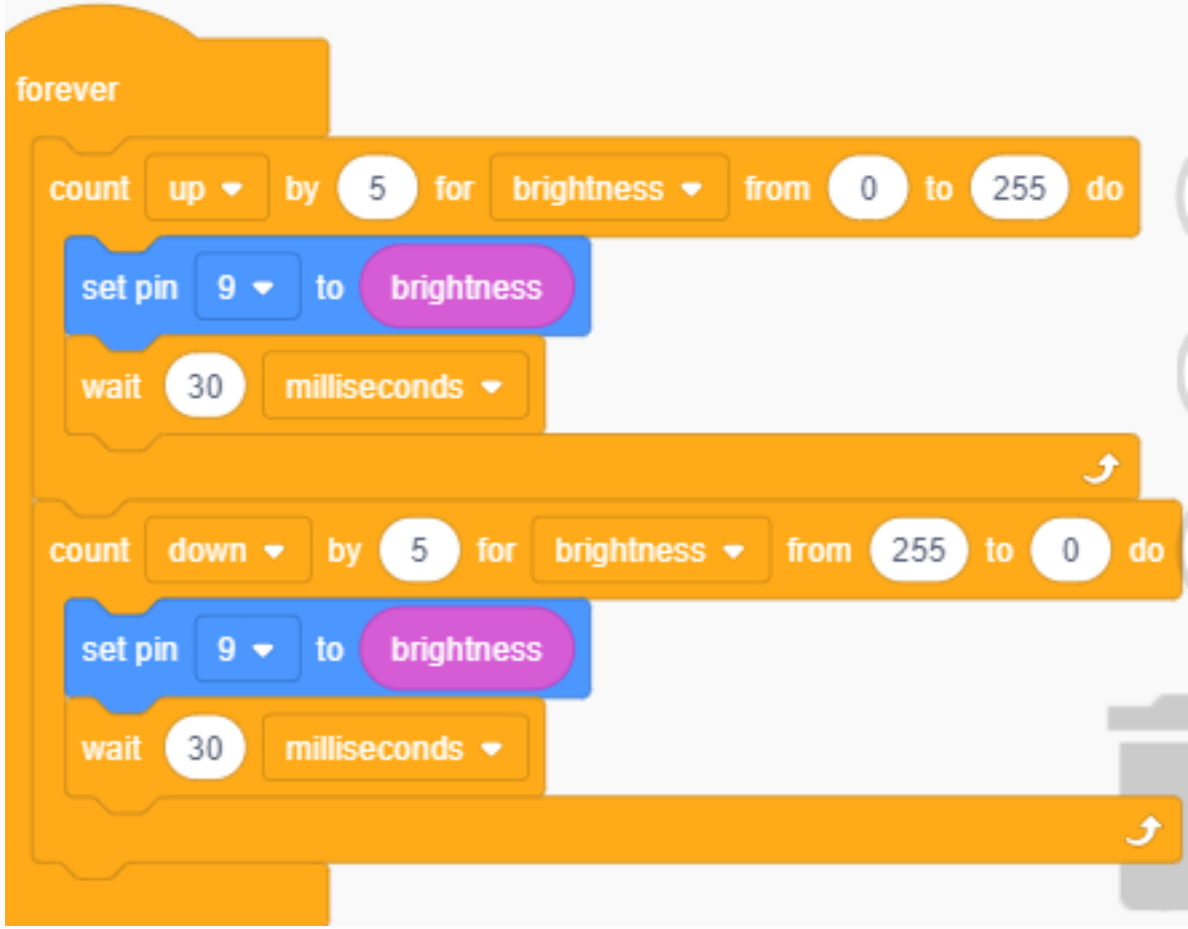


Analogue Outputs

The Code

- *count control command (yellow)*: counts (up or down) the changing step of a variable value. In addition, it implements a for loop that defines the range of values the variable value can assume.

- This creates a PWM. A PWM changes the voltage applied to a load allowing us to control its parameters. For example, if the load is an LED, we can control its brightness.



Summary

We introduced the Arduino Platform and Microcontrollers.

We simulated prototypes to practice digital and analogue input/outputs.

We learned the electronic components: LED, resistor, potentiometer, breadboard, and pushbutton.

We learned programming: creating variables, if conditionals, for loops, reading and writing to digital pins.

We learned how to send data to the serial monitor

02

Arduino: Sensors and Actuators

UNIVERSITY OF WARWICK

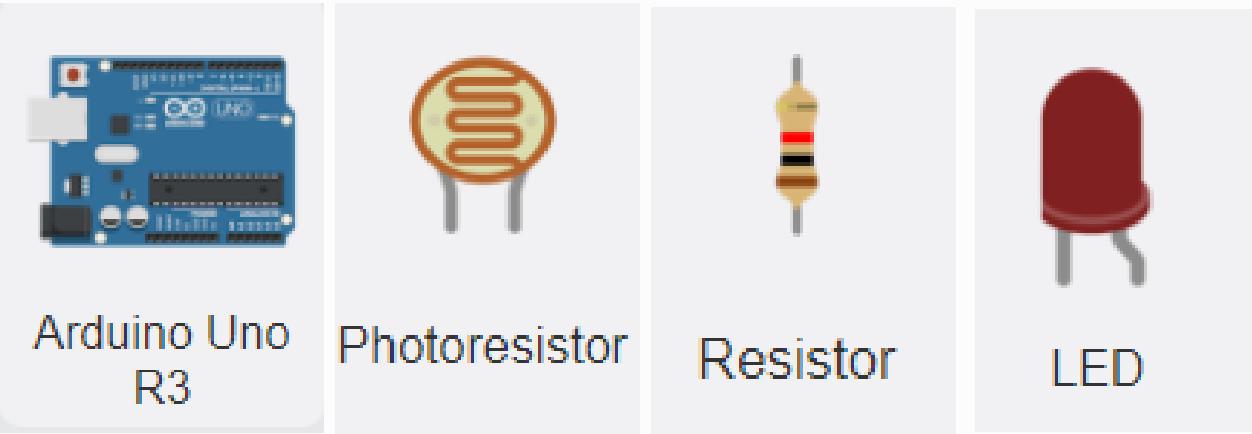
Light-Dependent Resistor (LDR)

- In this example, you will learn how to read the LDR sensor. LDR is a photoresistor whose resistance value varies according to the incident light. It has a low resistance value in the presence of light and a high value in its absence.

What will I learn?

- Write an analogue output.
- Use and LDR.

Components



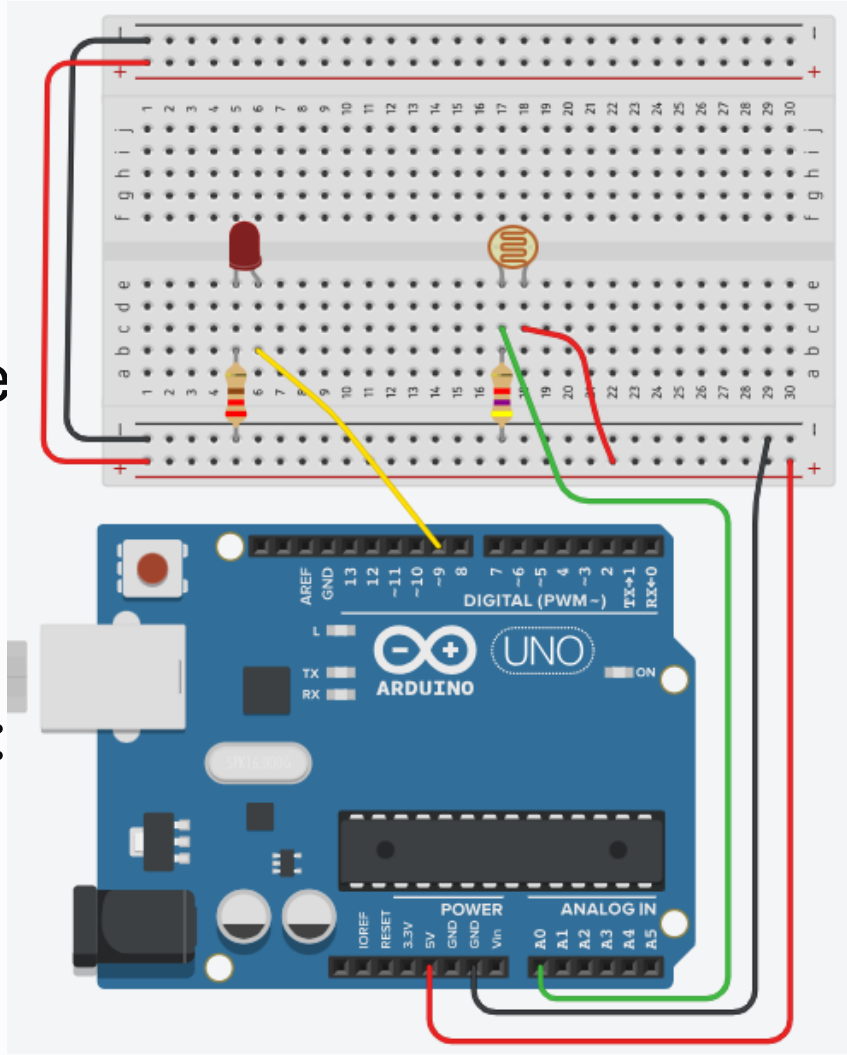
LDR Sensor

The Circuit

You will need:

- 1 Arduino Uno board.
- 1 LED.
- 2 Resistors (220Ω for the LED and 4.7kΩ for the LDR).
- 1 Breadboard.

Connect all components like in the following image:

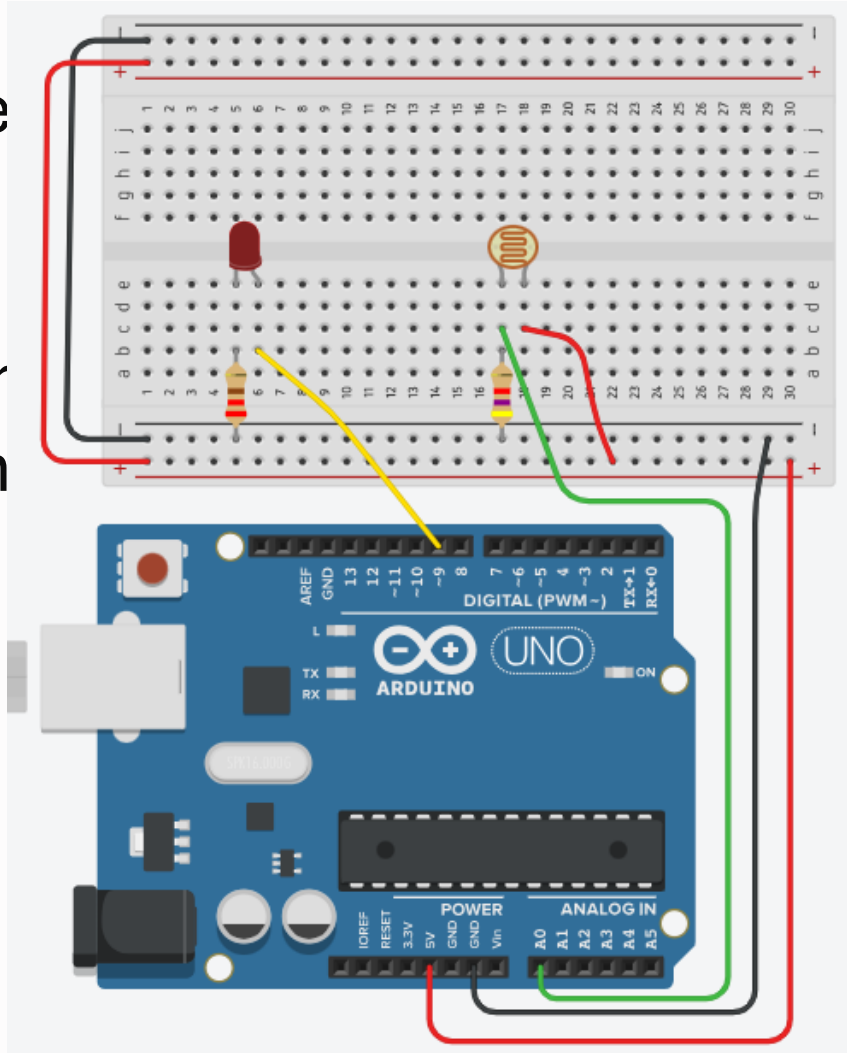


LDR Sensor

The Circuit

- Note that the LDR has two terminals and can be connected in any way, like any other resistor.

- The LDR must be connected in series with another resistor and connect the Arduino pin in between them.



LDR Sensor

The Code

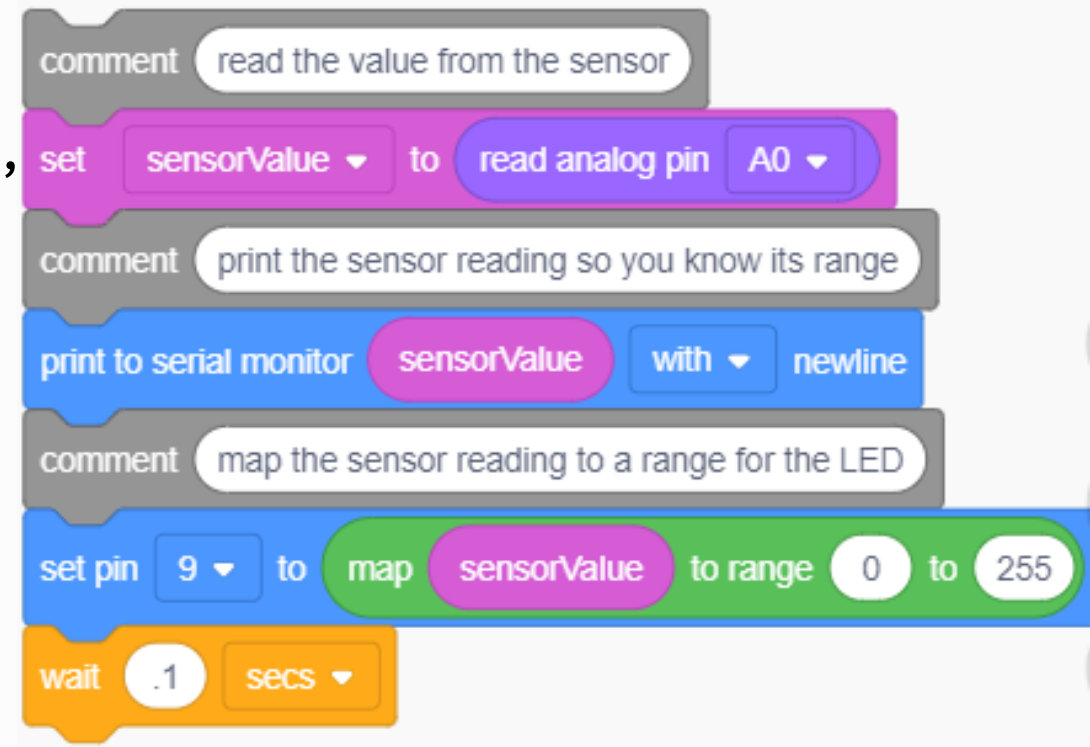
- The map function converts any value to a desirable range.

- In this example, we are converting ‘sensorValue’ in the range 0 to 255, as follows:

$$value = \frac{(sensorValue - analogPin_{min}) * (map_{high} - map_{min})}{(analogPin_{max} - analogPin_{min}) + map_{min}}$$

Example:

$$value = \frac{(1023 - 0) * (255 - 0)}{(1023 - 0) + 0} = 255$$



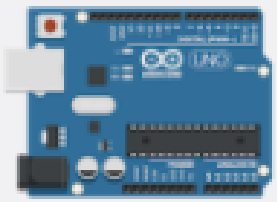
Ultrasonic Sensor

- In this example, you will learn how to read the Ultrasonic sensor. It allows measuring distances based on sound signals.

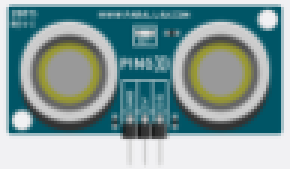
What will I learn?

- Use an Ultrasonic sensor.

Components



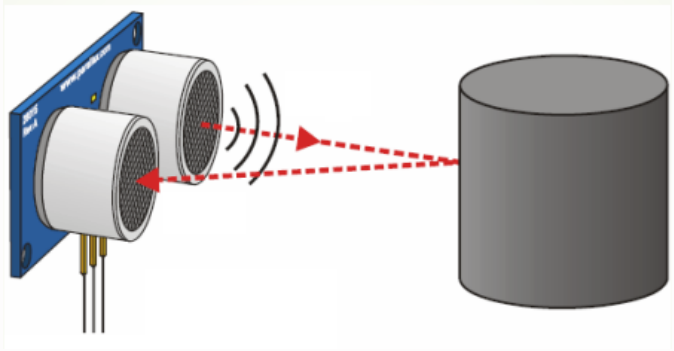
Arduino Uno R3



Ultrasonic Distance...

Ultrasonic Sensor

- The ultrasonic distance sensor consists of a transmitter and receiver capable of measuring distances based on the speed of sound in the air.

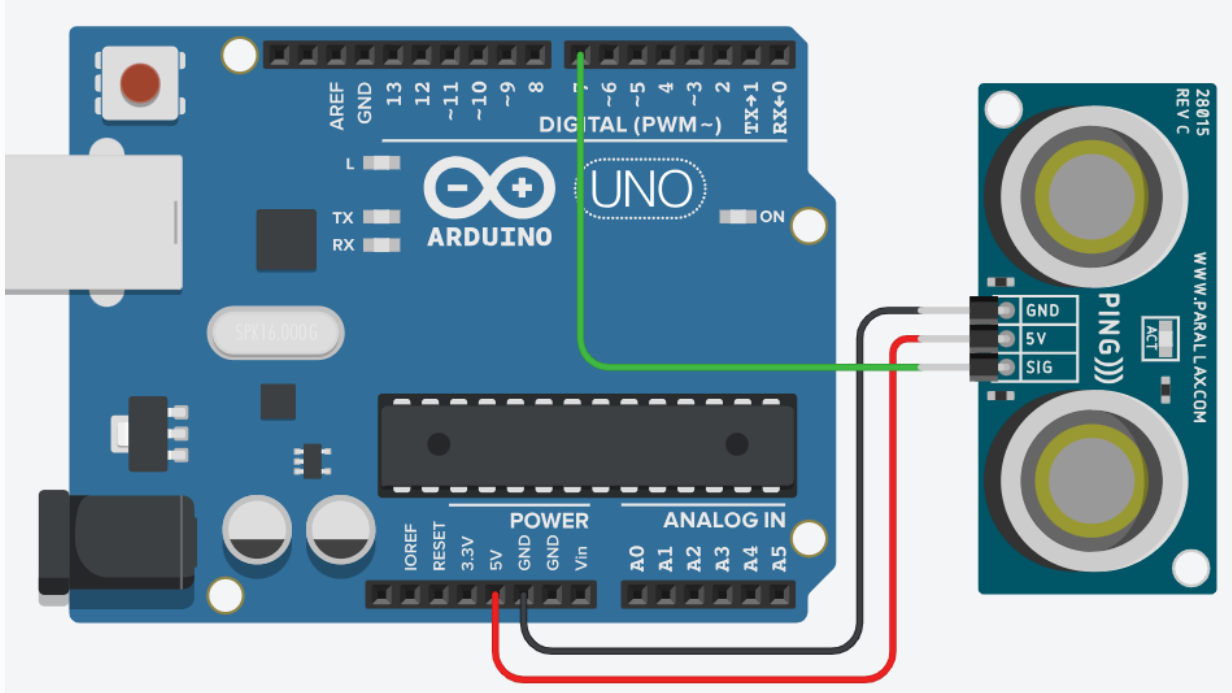


- When given a pulse on the trigger pin, the transmitter emits a sound signal. When hitting an object, this sound signal is reflected to the sensor, where the receiver captures it (echo pin). The time taken from the transmission of the sound signal to its capture by the receiver can be used to determine the object's distance based on the sound propagation speed in the air (340m/s).

Ultrasonic Sensor

The Circuit

- In this example, the transmitter and receiver are represented by the signal (SIG) pin, which is connected to pin 7.

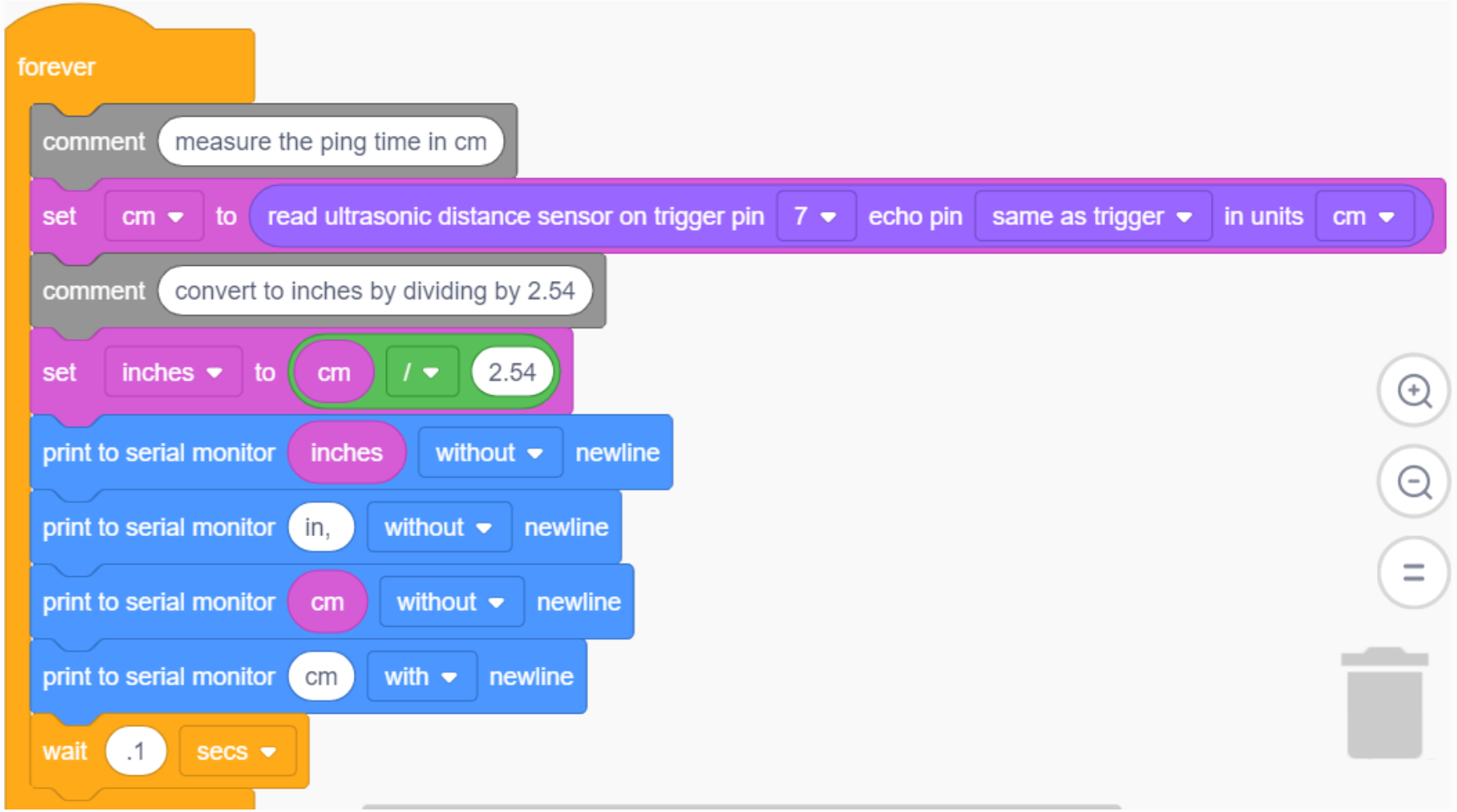


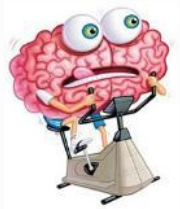
- Note the need to connect to the 5V and GND.

Ultrasonic Sensor

The Code

- The read ultrasonic input command reads the object's distance in cm.
- The value in cm is then converted to inches and stored in the variable *inches*.





Practice Questions

1. Use an Ultrasound sensor to light up 2 LEDs according to the distance between an object and the sensor. If the object is close, a red LED should turn ON; otherwise, a green LED should turn ON.
2. Repeat the previous question and replace the red LED with a buzzer (search on Tinkercad for Piezo).
3. Now, use an LDR sensor to light up 2 LEDs according to the light's intensity. If the intensity is low, a red LED should turn ON; otherwise, a green LED should turn ON.

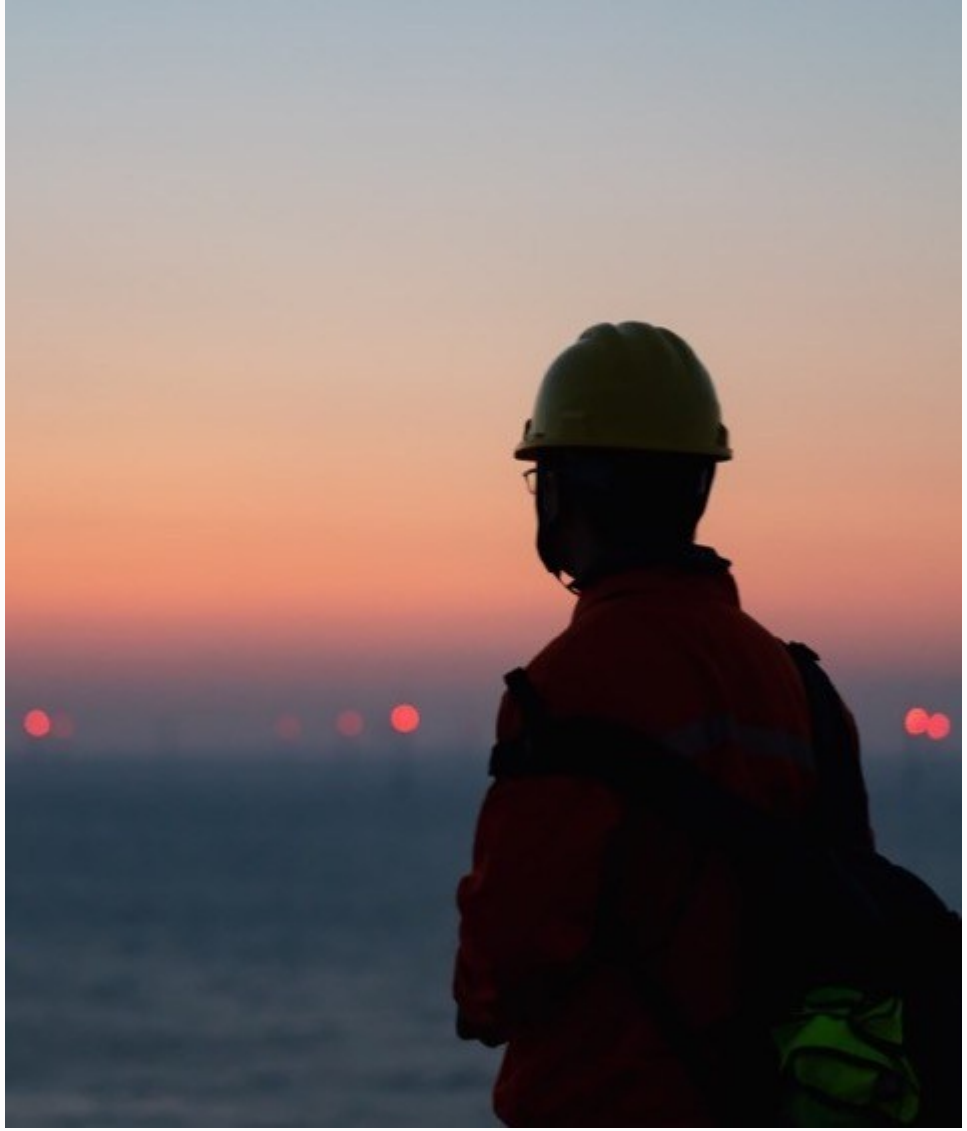
Summary

We introduced the LDR and Ultrasonic sensors.

We introduce loads such as LED and Piezo

Thank you

Leonardo.Alves-Dias@warwick.ac.uk
Liping.Zheng.1@warwick.ac.uk



UNIVERSITY OF WARWICK