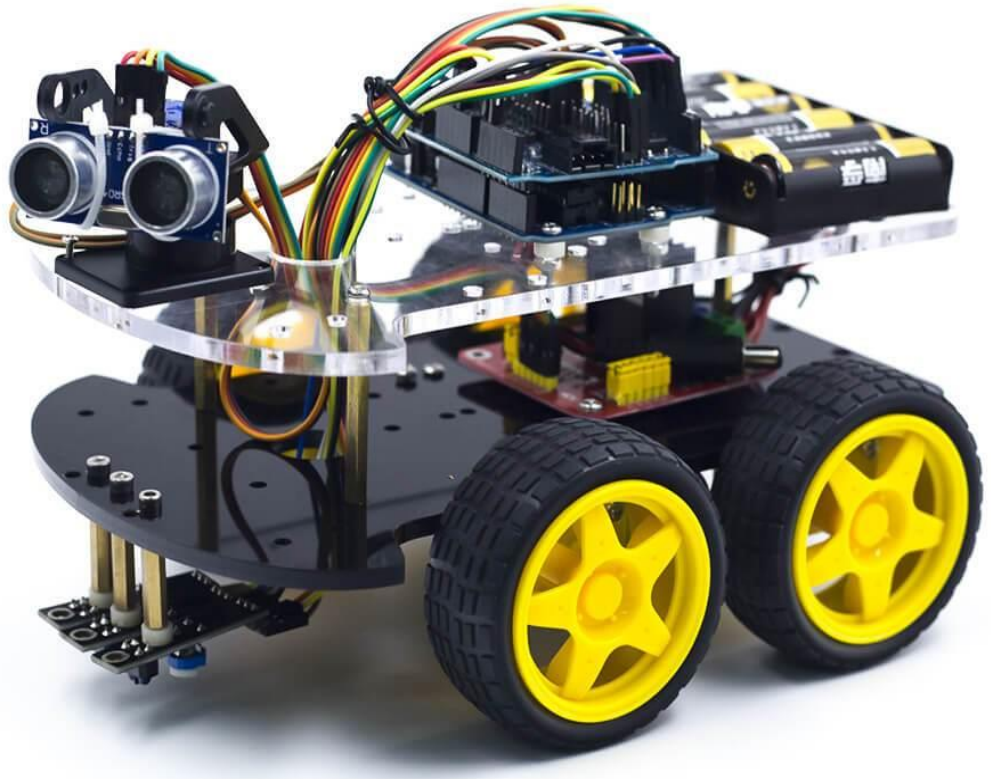
NSDC – Junior Skills Championship
Mobile robotics

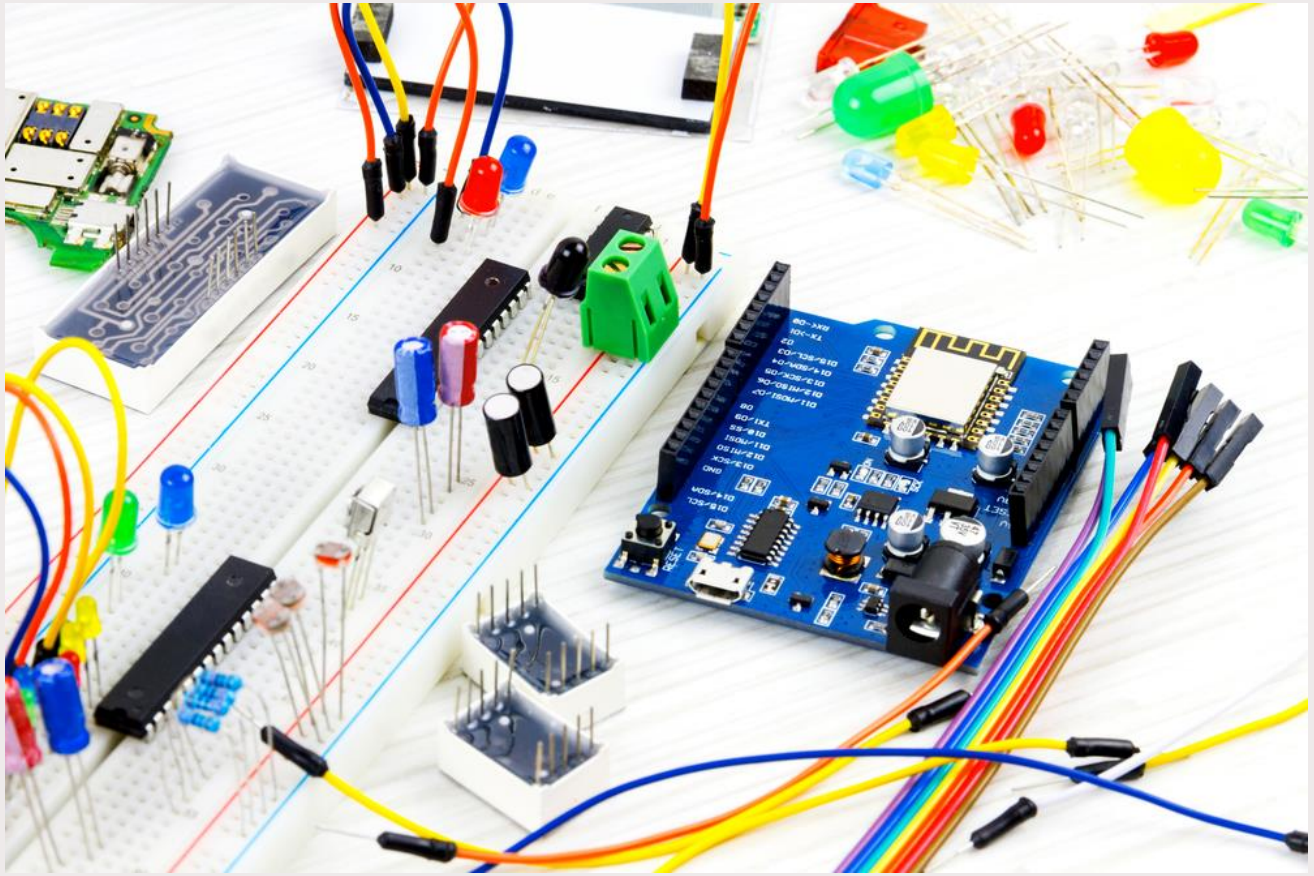


Friday, July 30, 2021

HOLOSUIT

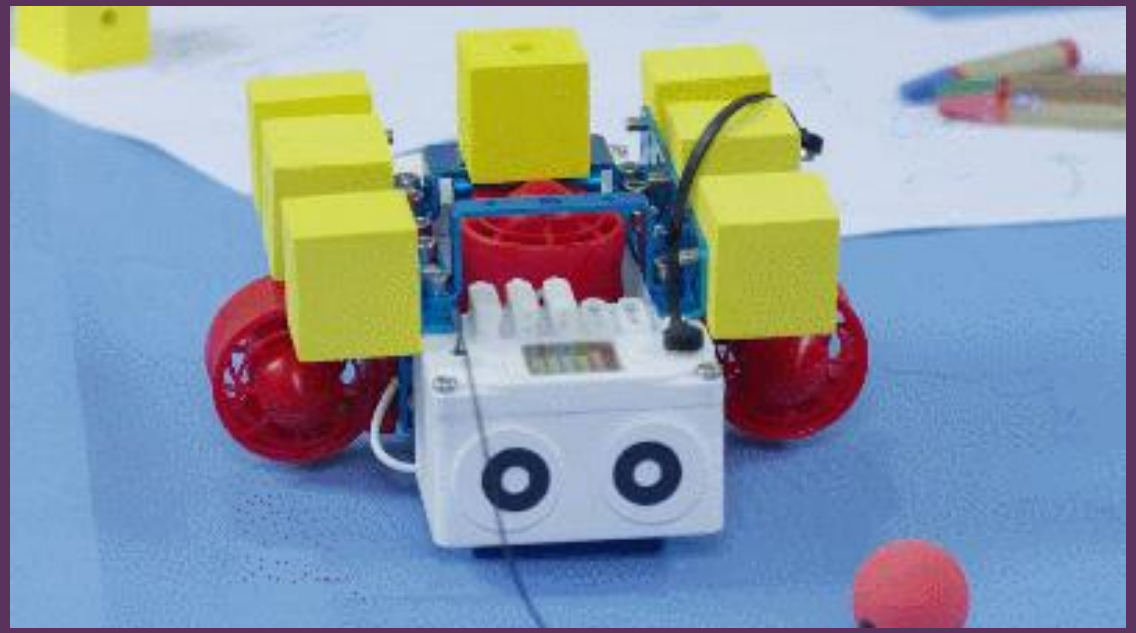
Agenda - Day 3

- Basics of Block programming
  - What Is Block-based Programming?
- Arduino UNO
  - Input and Output of Arduino UNO
- Arduino Block Programming
  - Pins, Controls, Operations
  - Example Programs
- Arduino interfacing with I/O devices
  - Standard interface circuits for Input devices
  - interface circuits for Various Input devices
- H bridge concept for mobile robots
  - Basics of H Bridge
  - Transistor H Bridge
  - L293D Motor Driver IC



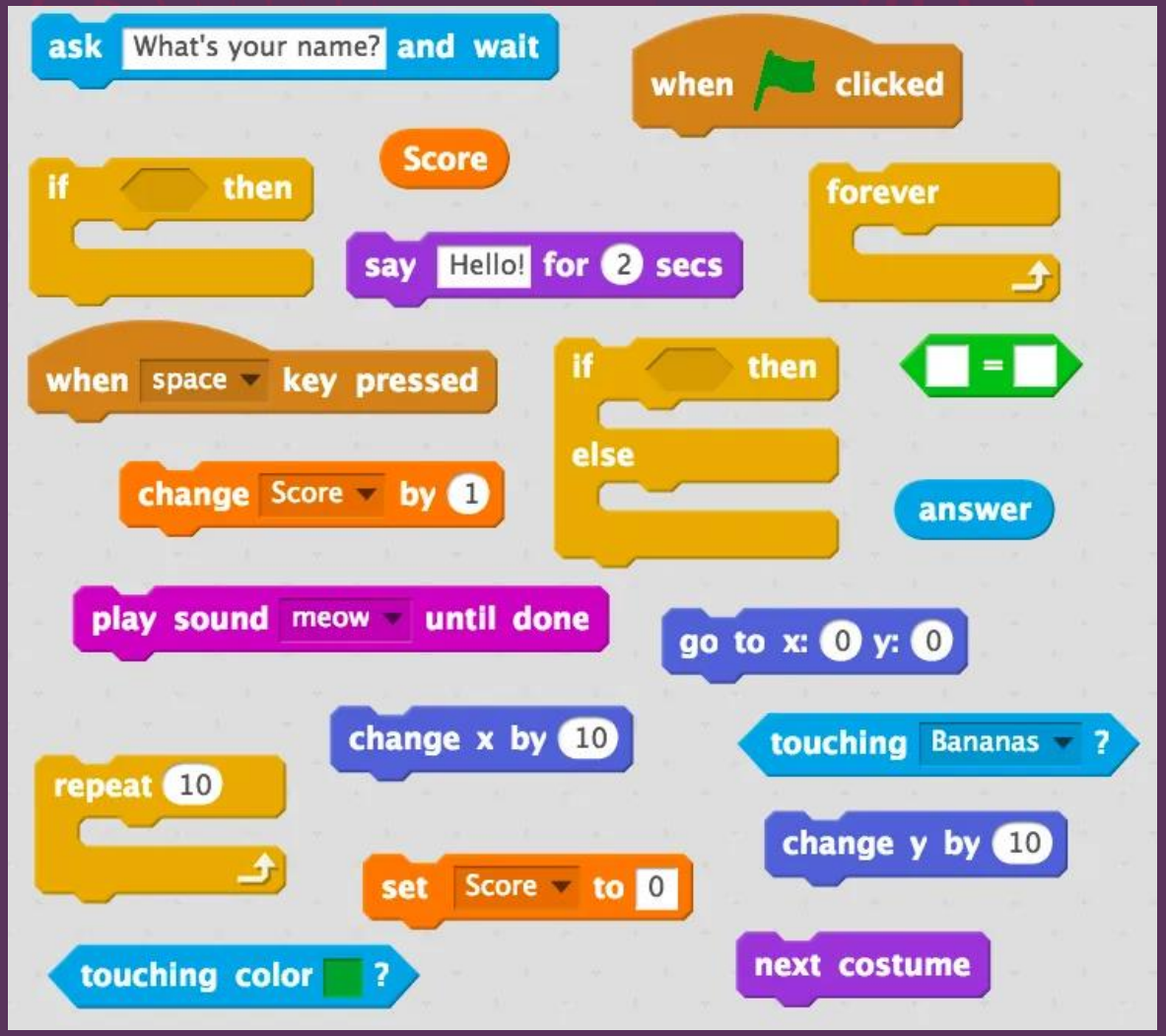
HOLOSUIT

Mobile Robots using bock programming



Block Programming

- Block-based programming has a number of key features that make it distinct from conventional text-based programming and other visual programming approaches.
- Block-based programming uses a programming primitive as puzzle piece metaphor as a means of providing visual cues to the user as to how and where commands may be used.



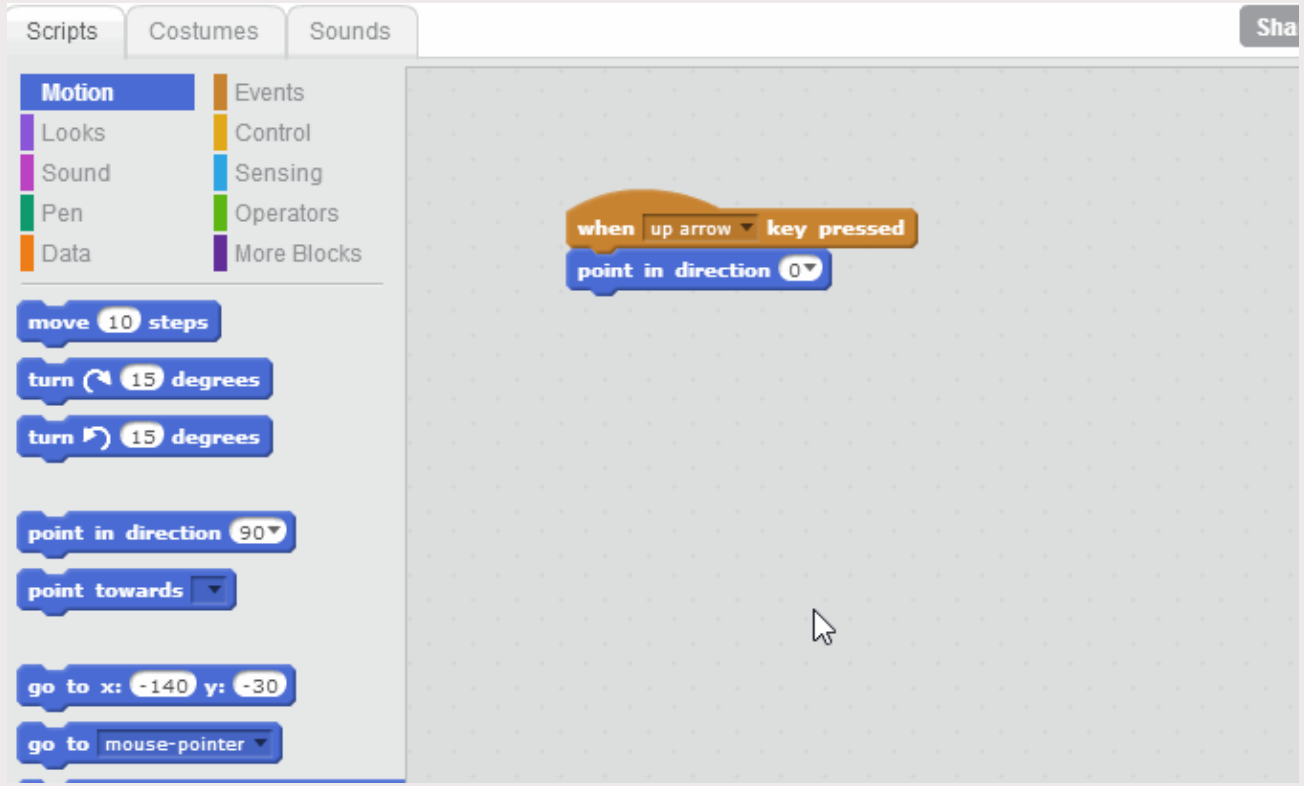
HOLOSUIT

Block Programming

- Block-based programming environments have been designed for children and students.
- Advanced environments are designed for kids ages eight to 16. Writing a program in a block-based environment takes the form of dragging and dropping programming instructions together

What do you mean by “blocks?”

- it’s a way to describe the “chunks” or “pieces” of instructions a user is putting together in order to tell their creation what to do.

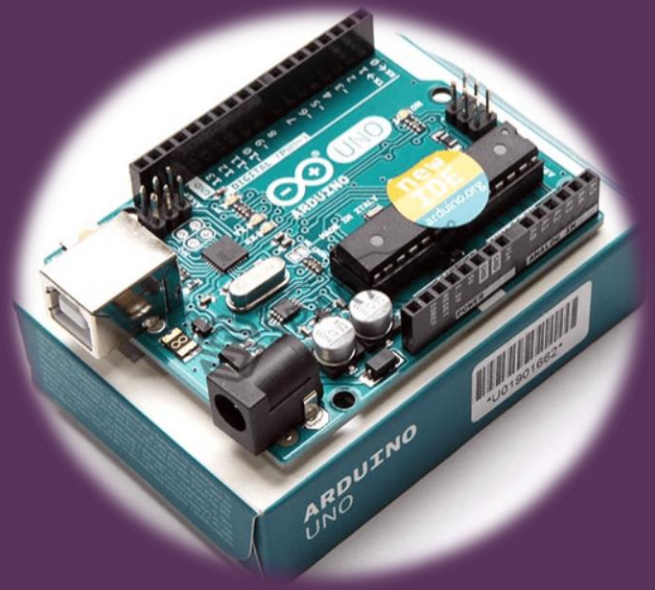


Robotics kits - Block programming

- Makeblock Ultimate 2.0 10-in-1 Robot Kit
- Robo Bit Buggy
- Lego Boost Robot
- Cozmo Robot

Arduino - Microcontroller

- Arduino is an open-source hardware platform for microcontroller prototyping
- Arduino is an open-source electronics platform based on easy-to-use hardware and software
- It is Inexpensive, Simple, clear programming environment, Open source and extensible software / hardware
- Arduino boards are extremely popular as a starting point for using microcontroller technologies

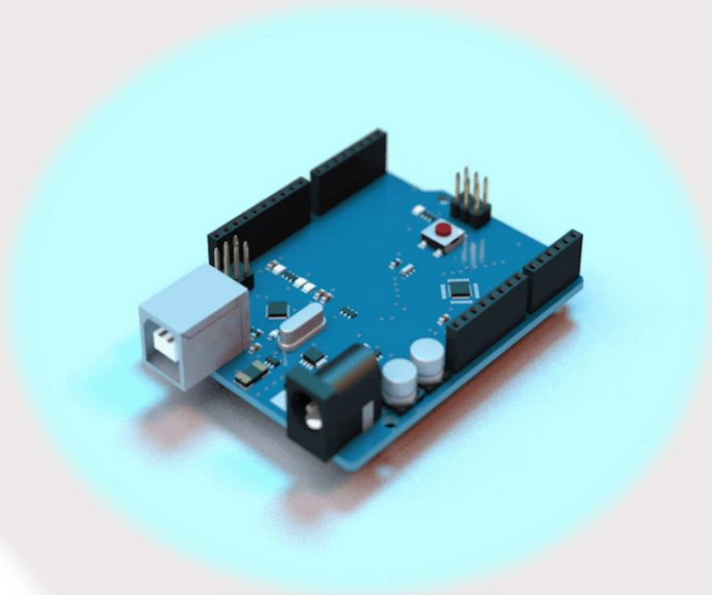


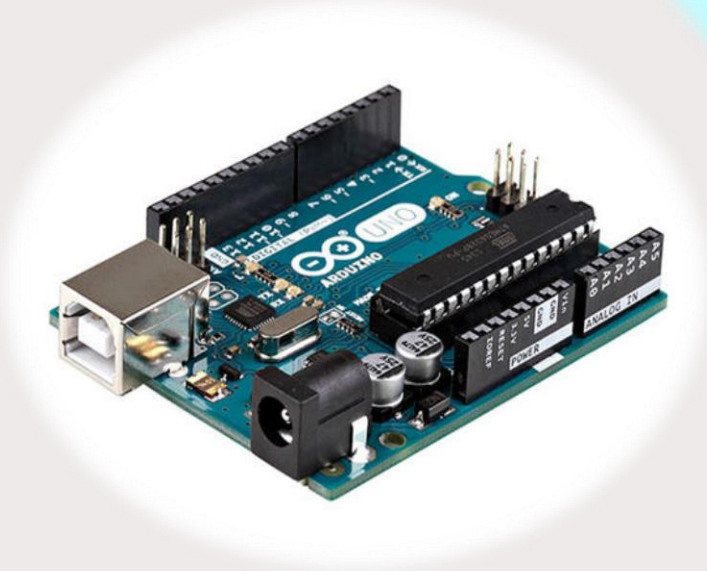
Why Arduino UNO?

- Low cost (around $30)
- Open source design
- Easy-to-use and cross-platform IDE
- Availability of plug-in Sensors and Shields (expansion hardware)

Arduino UNO

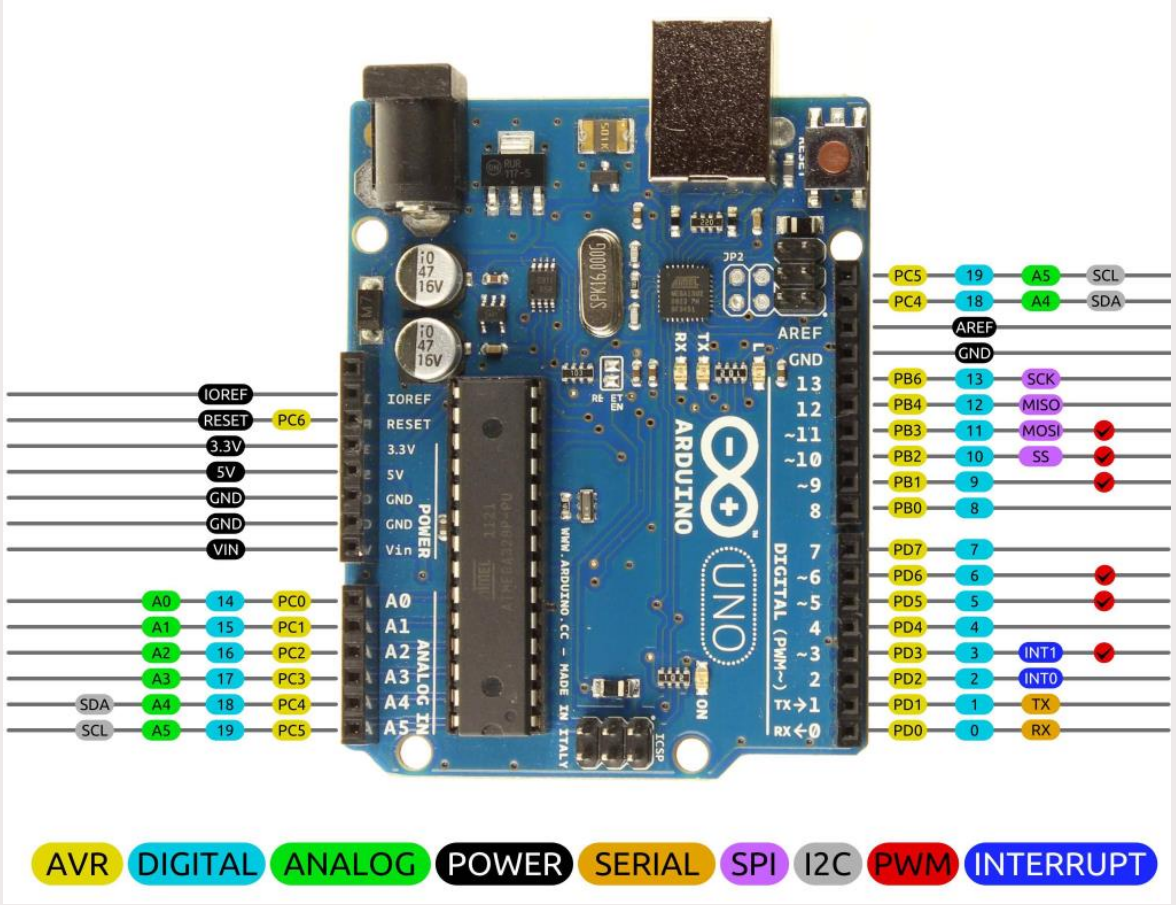
- "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0
- We have a variety of sensors which can be easily added to this board
- Arduino Uno is a microcontroller board based on the ATmega328P.



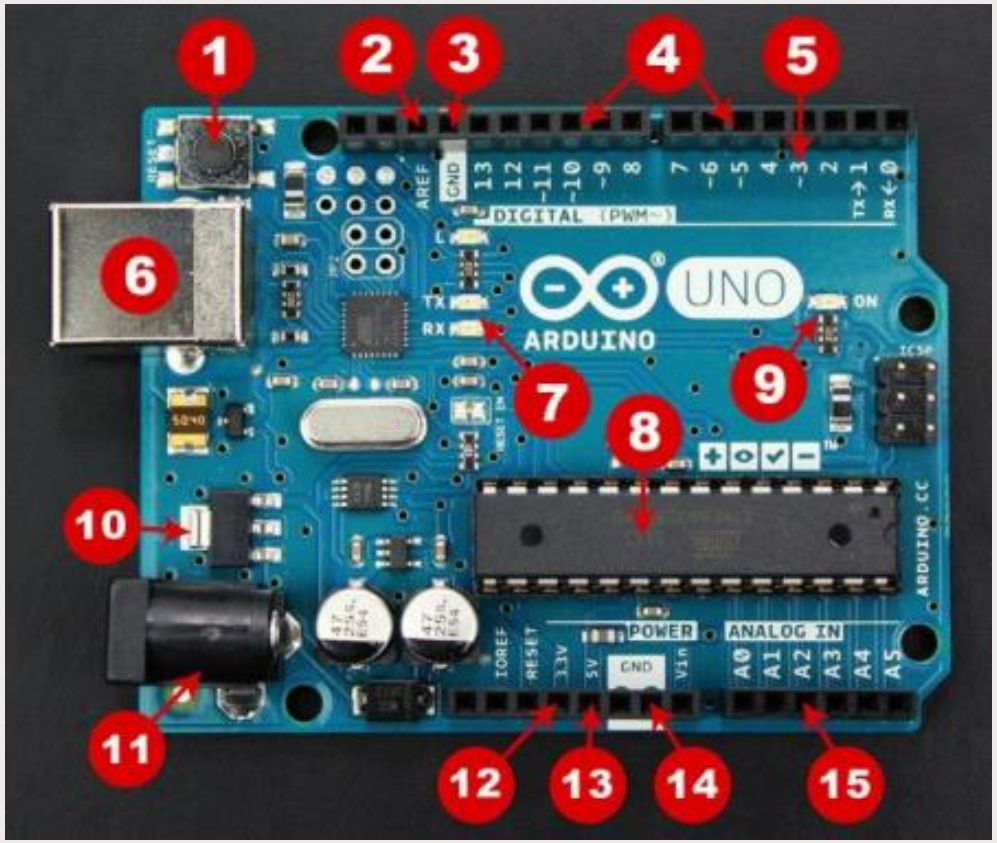


Arduino UNO

- Arduino has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 Analog input/output pins , a 16 MHz quartz crystal
- Each of the 14 digital pins on the Uno can be used as an input or output
- Each of the 6 Analog pins on the Uno can be used as an input or output
- Each of the PWM namely Pin 3, 5, 6, 9, 10, and 1 can be used as PWM output pins
- There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



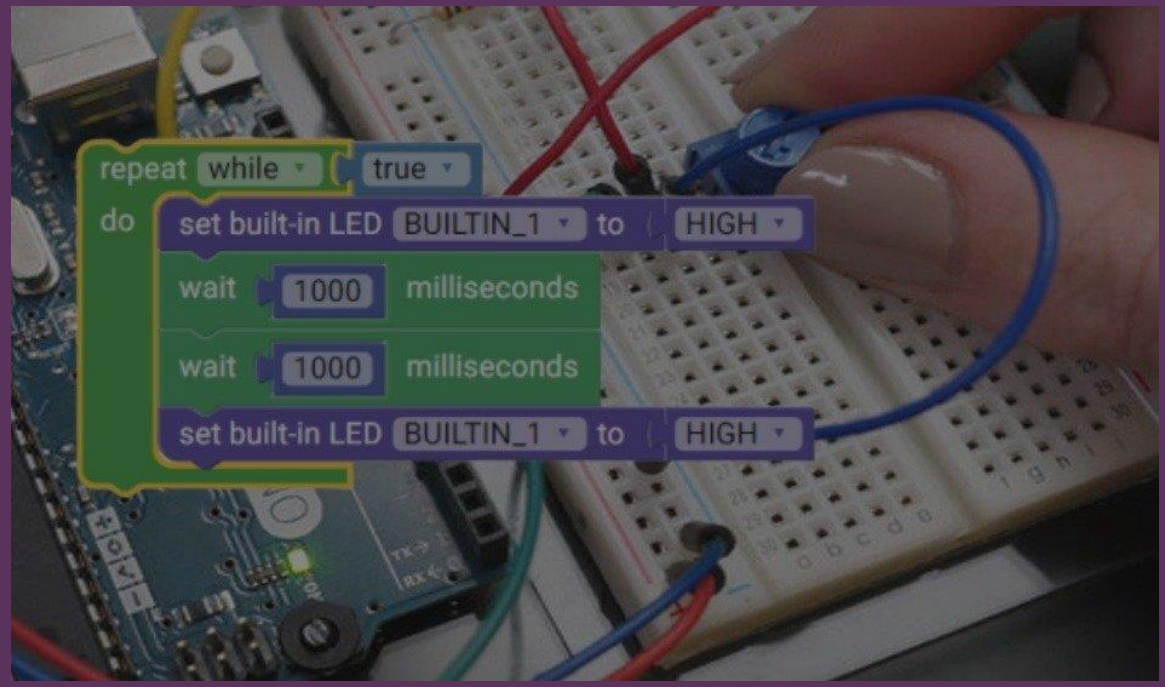
Arduino UNO



1. **Reset Button** – This will restart any code that is loaded to the Arduino board
2. **AREF** – Stands for "Analog Reference" and is used to set an external reference voltage
3. **Ground Pin** – There are a few ground pins on the Arduino and they all work the same
4. **Digital Input/Output** – Pins 0-13 can be used for digital input or output
5. **PWM** – The pins marked with the (~) symbol can simulate analog output
6. **USB Connection** – Used for powering up your Arduino and uploading sketches
7. **TX/RX** – Transmit and receive data indication LEDs
8. **ATmega Microcontroller** – This is the brains and is where the programs are stored
9. **Power LED Indicator** – This LED lights up anytime the board is plugged in a power source
10. **Voltage Regulator** – This controls the amount of voltage going into the Arduino board
11. **DC Power Barrel Jack** – This is used for powering your Arduino with a power supply
12. **3.3V Pin** – This pin supplies 3.3 volts of power to your projects
13. **5V Pin** – This pin supplies 5 volts of power to your projects
14. **Ground Pins** – There are a few ground pins on the Arduino and they all work the same
15. **Analog Pins** – These pins can read the signal from an analog sensor and convert it to digital

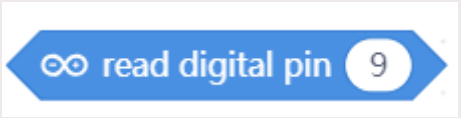
Arduino Block Programming

- The Blocks are high level abstract components which are connected sequentially to perform the desired task
- There are different categories of blocks to read, write, and control, let us learn the blocks we use for programming the Arduino.

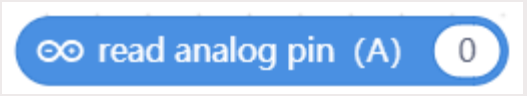


Arduino Block Programming

PINS



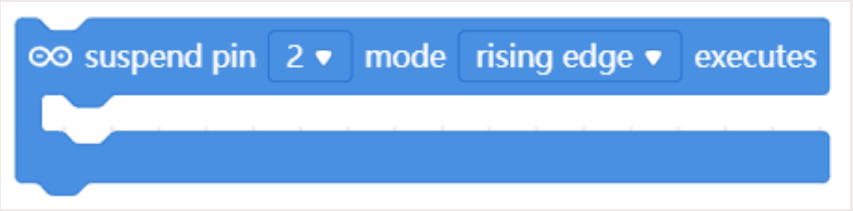
Reads the Digital I/O pins. This block will be used with other blocks wherever snapping is possible



Reads the Analog I/O pins. This block will be used with other blocks wherever snapping is possible.

Set the respective digital I/O's either as input or output

To assign the values to PWM pin at output mode

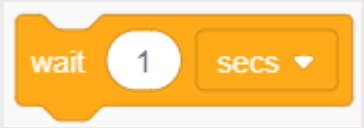


Block for Interrupt pins 2&3, if any interrupt to be raised, this block will be executed in the program

Used to send the data to the serial monitor

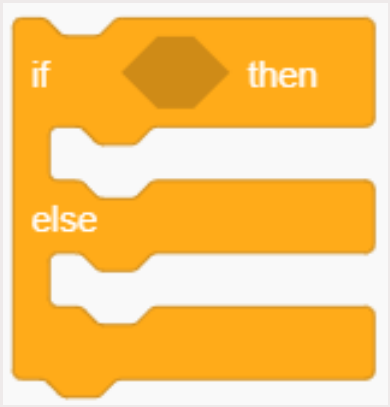
Arduino Block Programming

CONTROLS



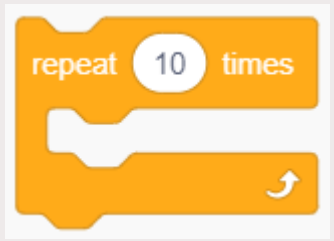
To introduce delay in the program sequence

Loop block with infinite iteration

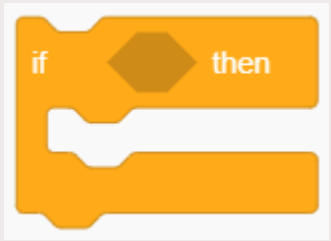


Conditional control block, if the condition is TRUE the block/s inside IF will execute else block/s inside ELSE will execute

Delay with Condition, the block will introduce delay till the condition is TRUE.



Loop block with specific iteration input and input should be integer

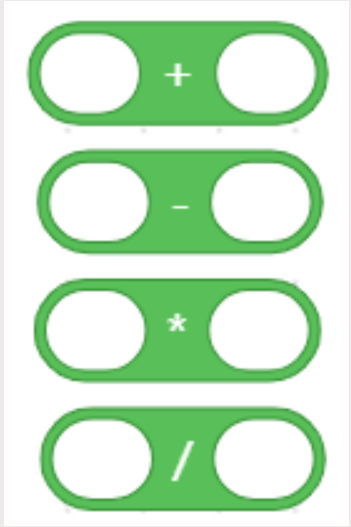


Conditional control block, if the condition is TRUE the block/s inside IF will execute.

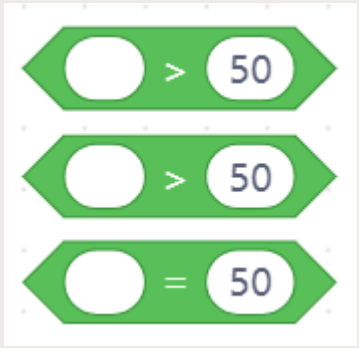
Loop block with the condition, the block repeatedly executes the inner blocks till the condition is TRUE.

Arduino Block Programming

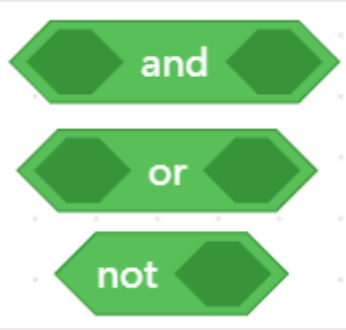
OPERATIONS



Arithmetic operation blocks, the inputs can be key in values or variables, or I/Os These blocks will be used in other blocks wherever snapping is possible.



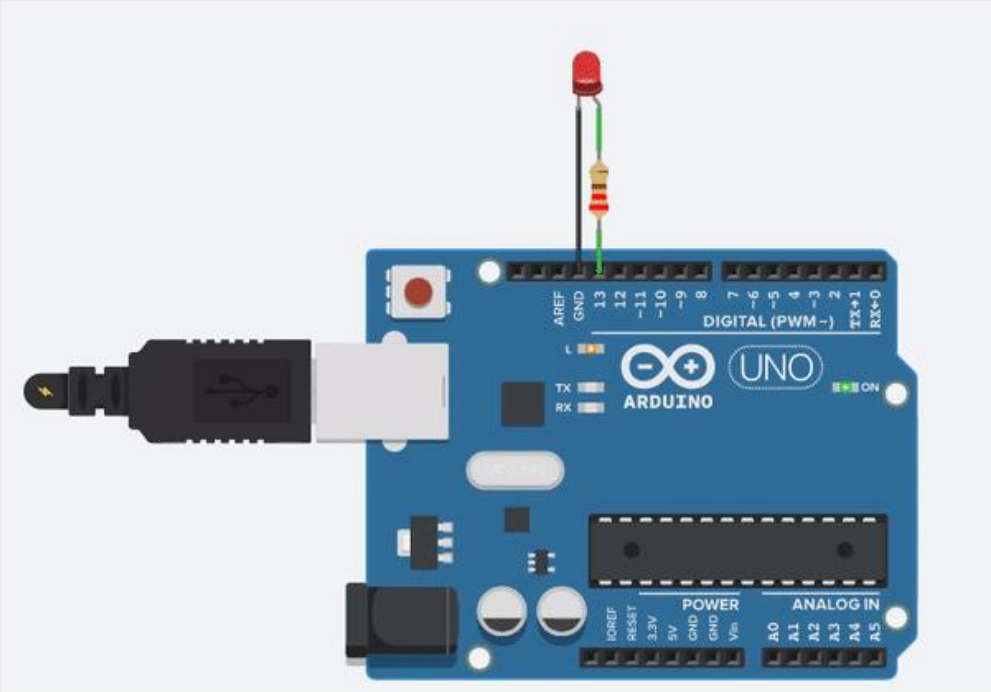
Relational operator blocks used for comparison. These blocks will be used in other blocks where ever snapping is possible.

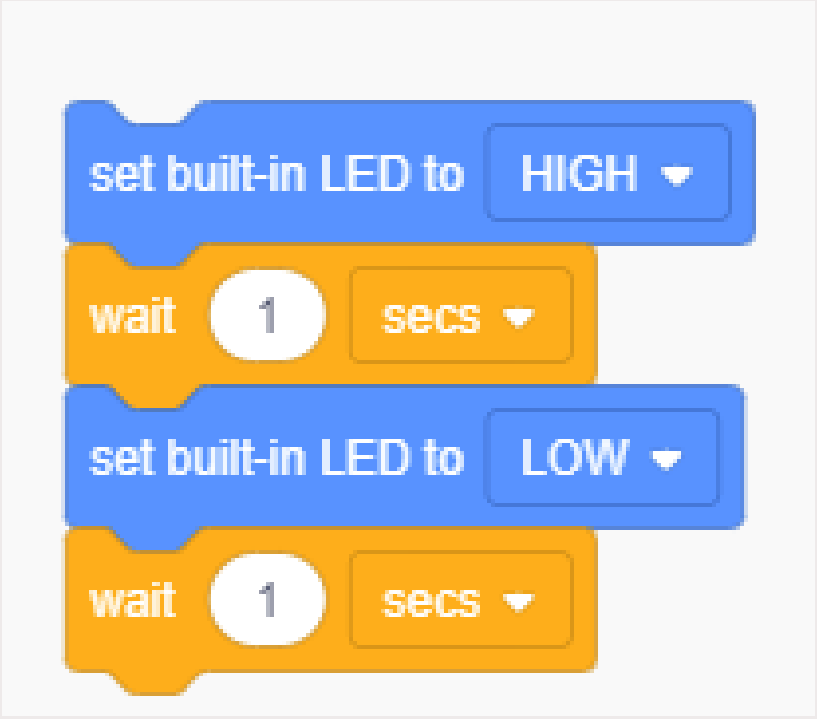


Logical Operation blocks. These blocks will be used in other blocks wherever snapping is possible.

Arduino Block Programming

Block Program: LED Blinking

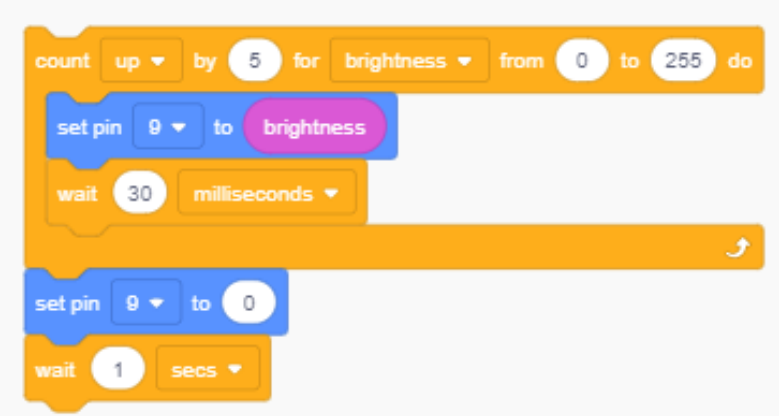


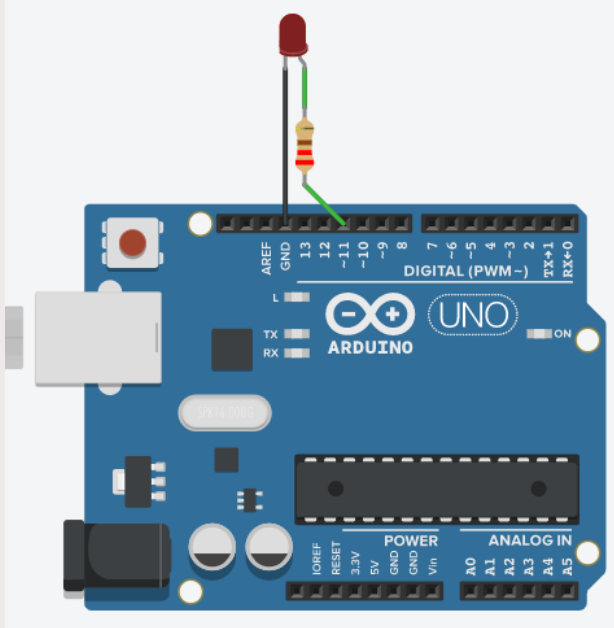


Arduino Block Programming

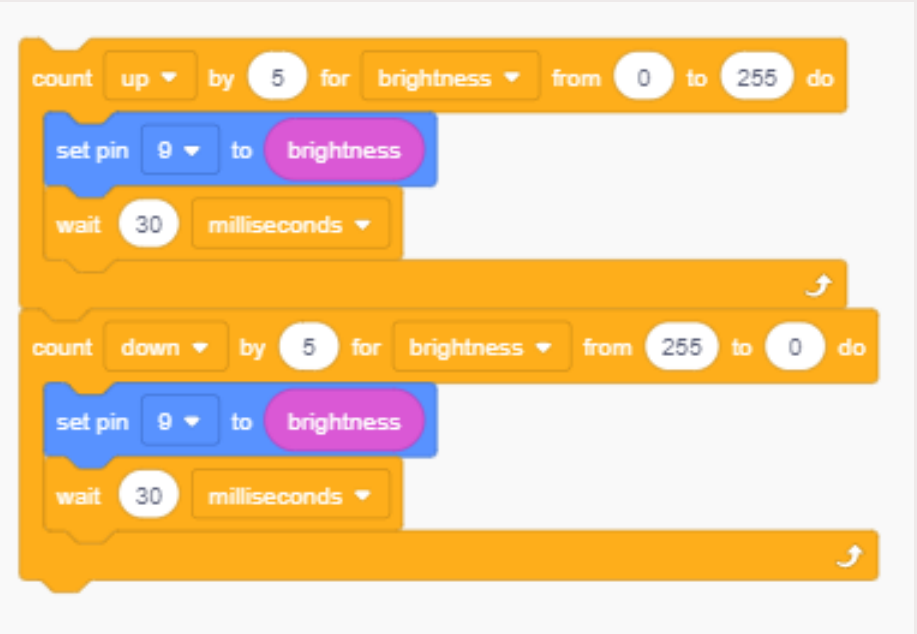
Fade-in and fade-out the LED

Block Code – Fade in





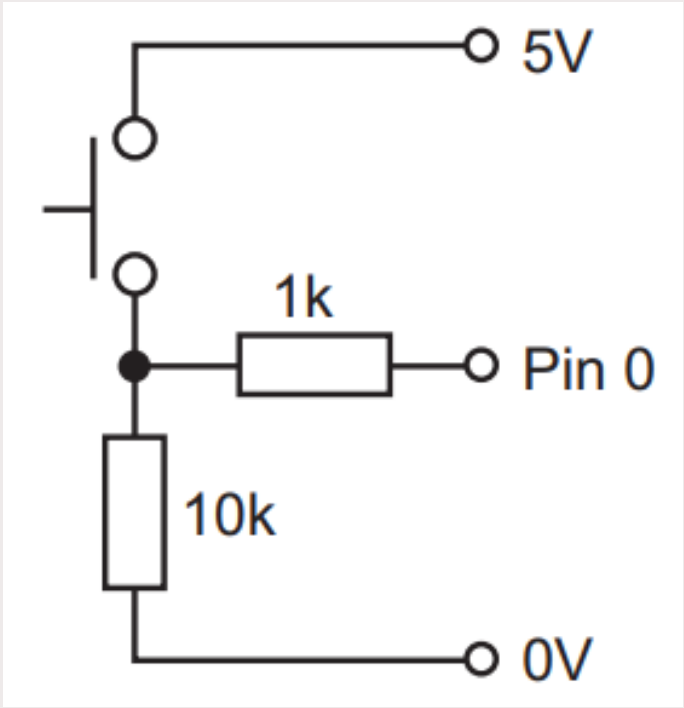
Block Code – Fade in and Fade out

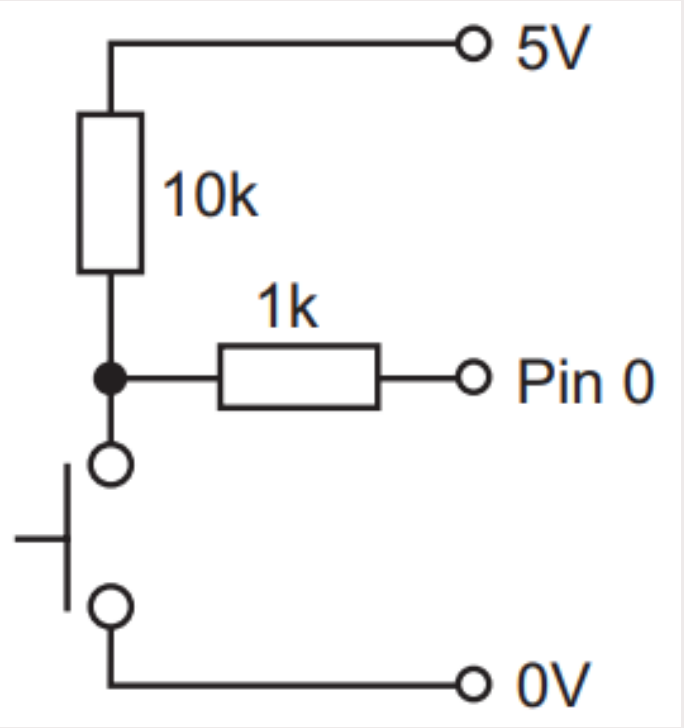


Arduino interfacing with I/O devices

Input Device - Switches

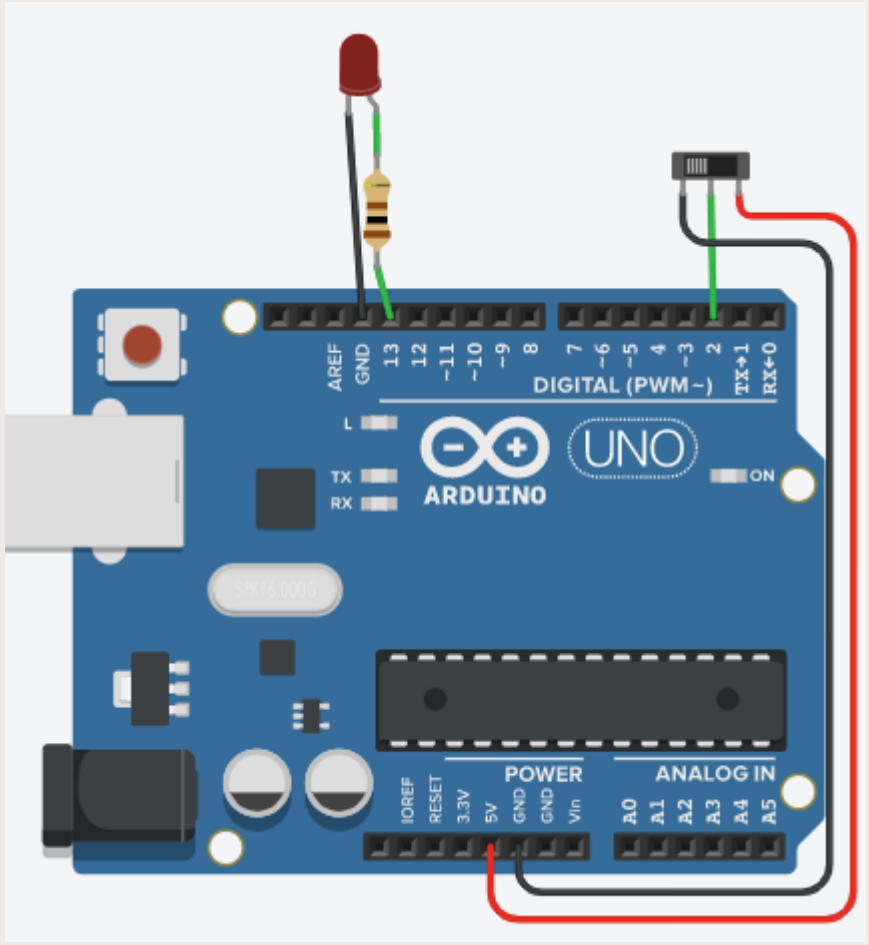
There are a large variety of switches available, but the majority all have two ‘contacts’ which are either ‘open’ (off) or ‘closed’ (on). The two circuits shown below can be used with almost all switches

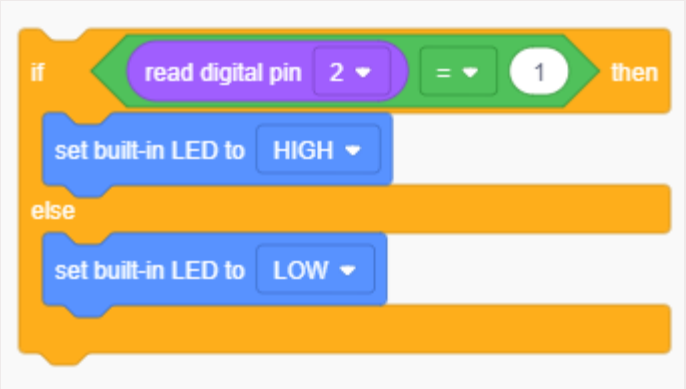




Arduino Block Programming

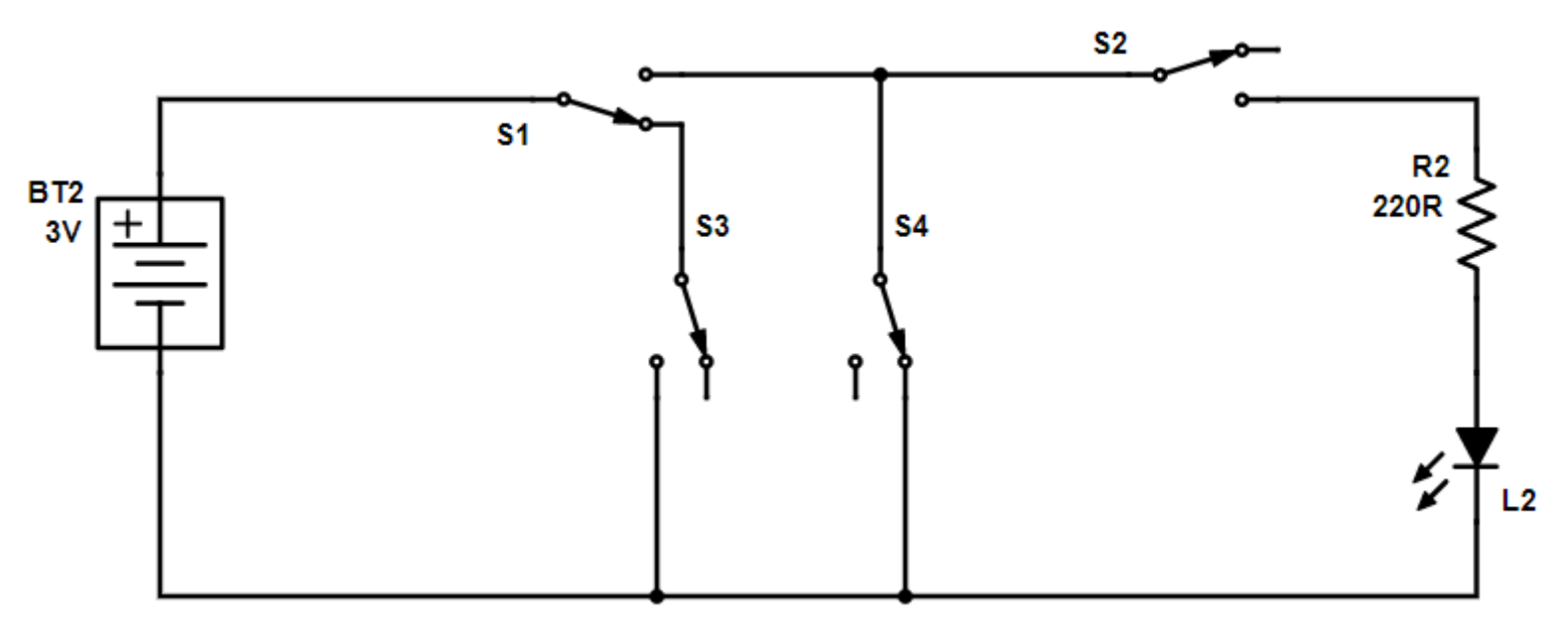
Arduino - LED with Switch





Arduino interfacing with I/O devices

LED Control



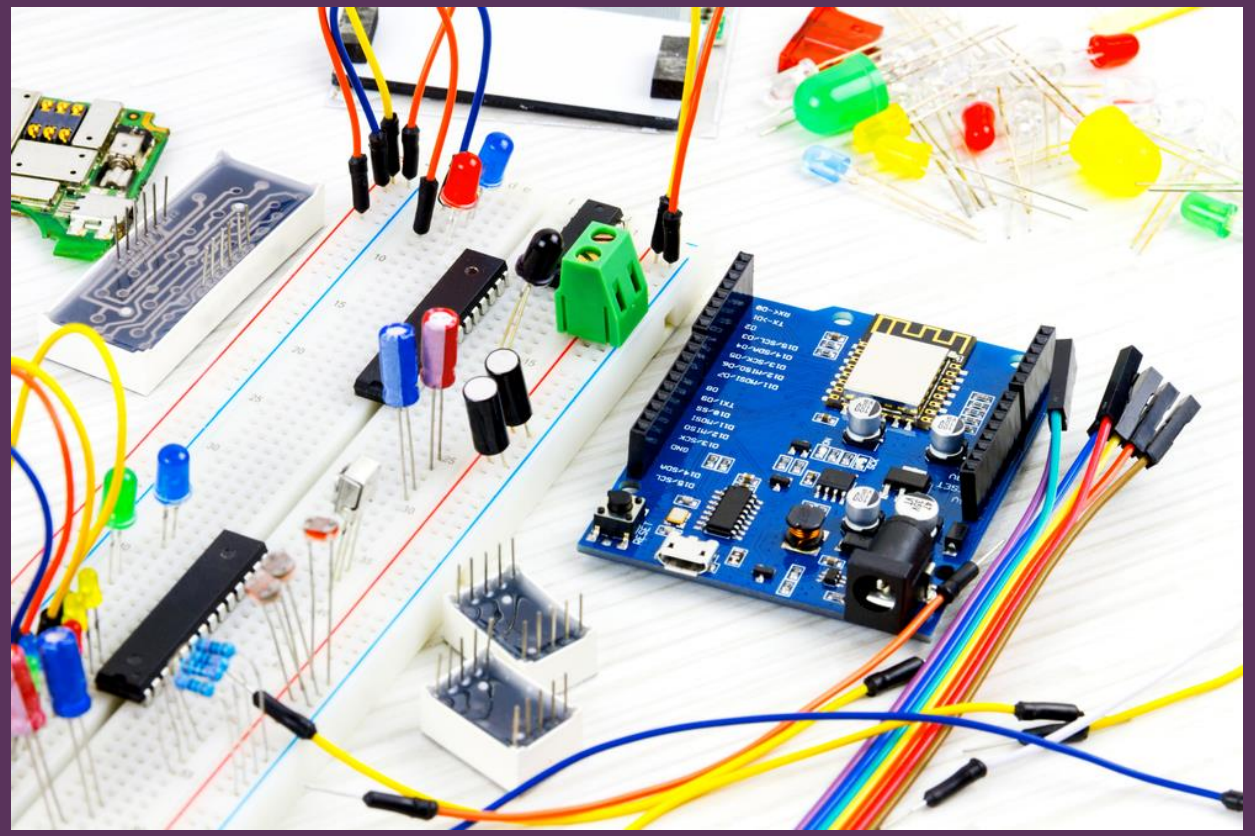
Arduino interfacing with I/O devices

In addition to the basic programming concepts let us move to the peripheral interface.

Arduino can be interface with many different input and output devices

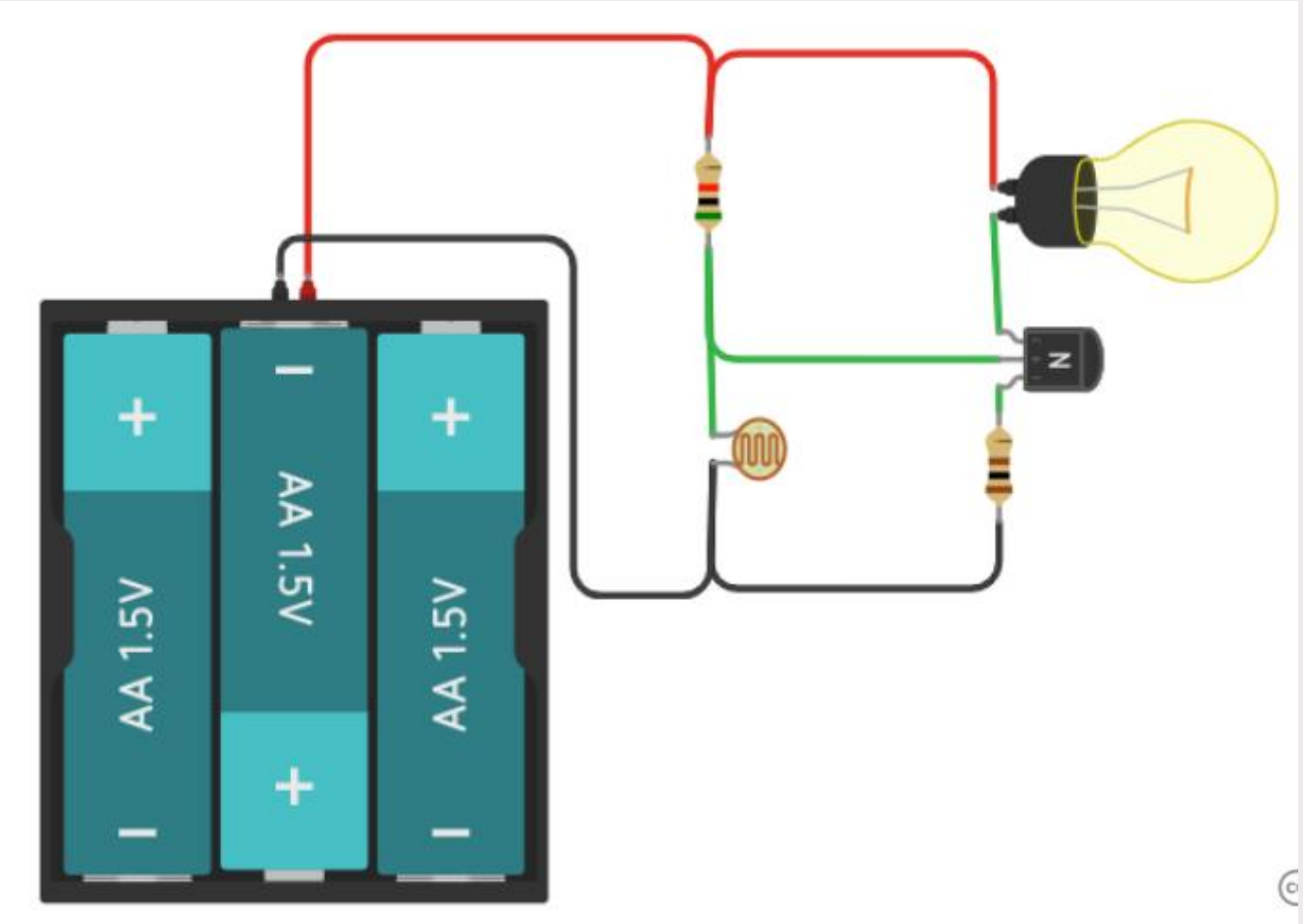
The interfacing circuits can be spited into four subsections:

- Introduction to ‘standard’ interfacing circuits
- Output Device Interfacing
- Input Device Interfacing
- Advanced Component Interfacing



Arduino interfacing with I/O devices

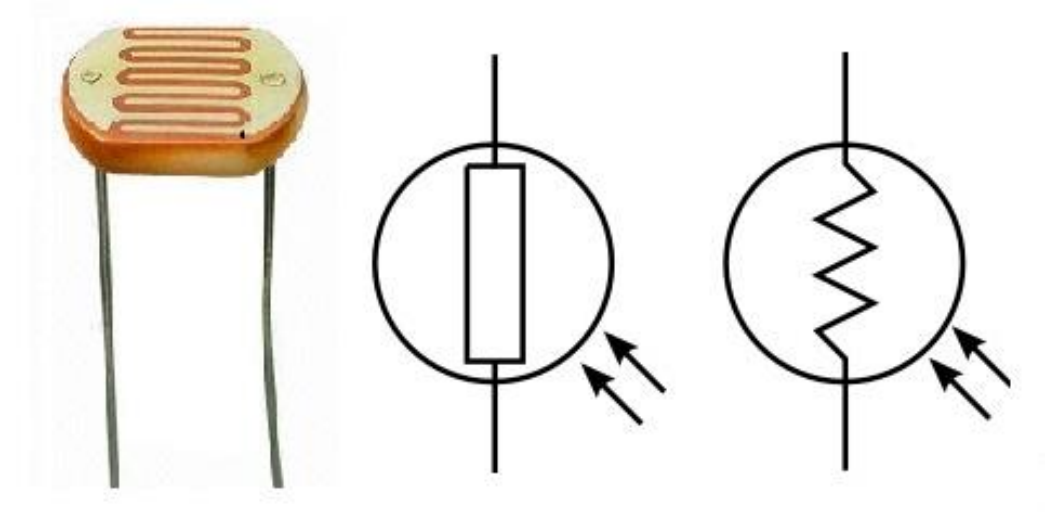
Light Dependant Resistor (LDR)

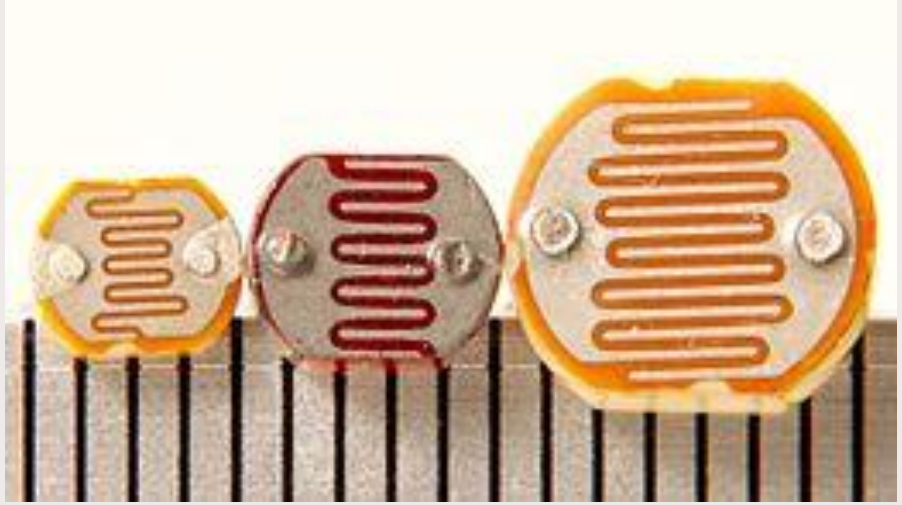


Arduino interfacing with I/O devices

Light Dependant Resistor (LDR)

- A Light Dependant Resistor (LDR) is a resistor that changes in value according to the light falling on it
- Light Dependant Resistor (LDR) are light- controlled variable resistors
- A commonly used device, the ORP-12, has a high resistance in the dark, and a low resistance in the light.

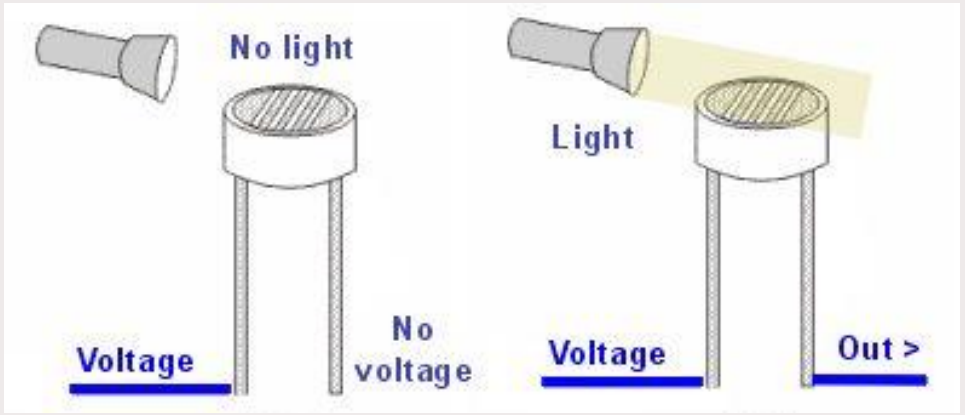


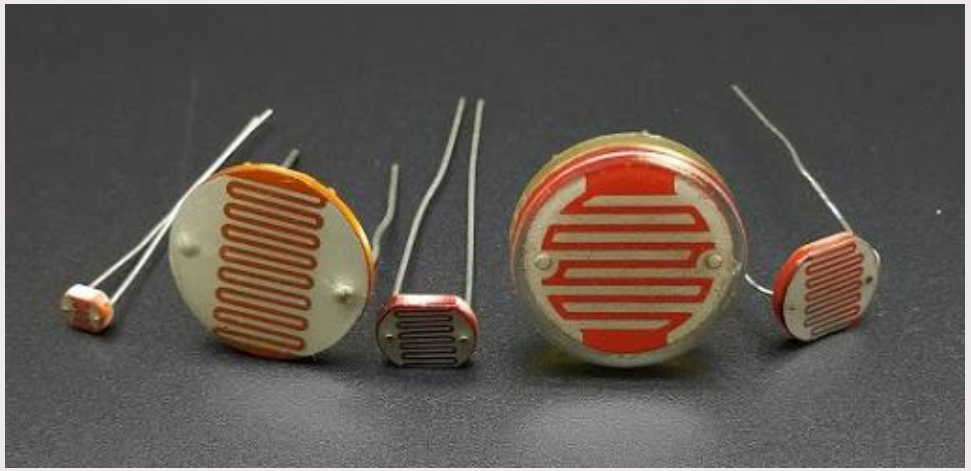


Arduino interfacing with I/O devices

Light Dependant Resistor (LDR)

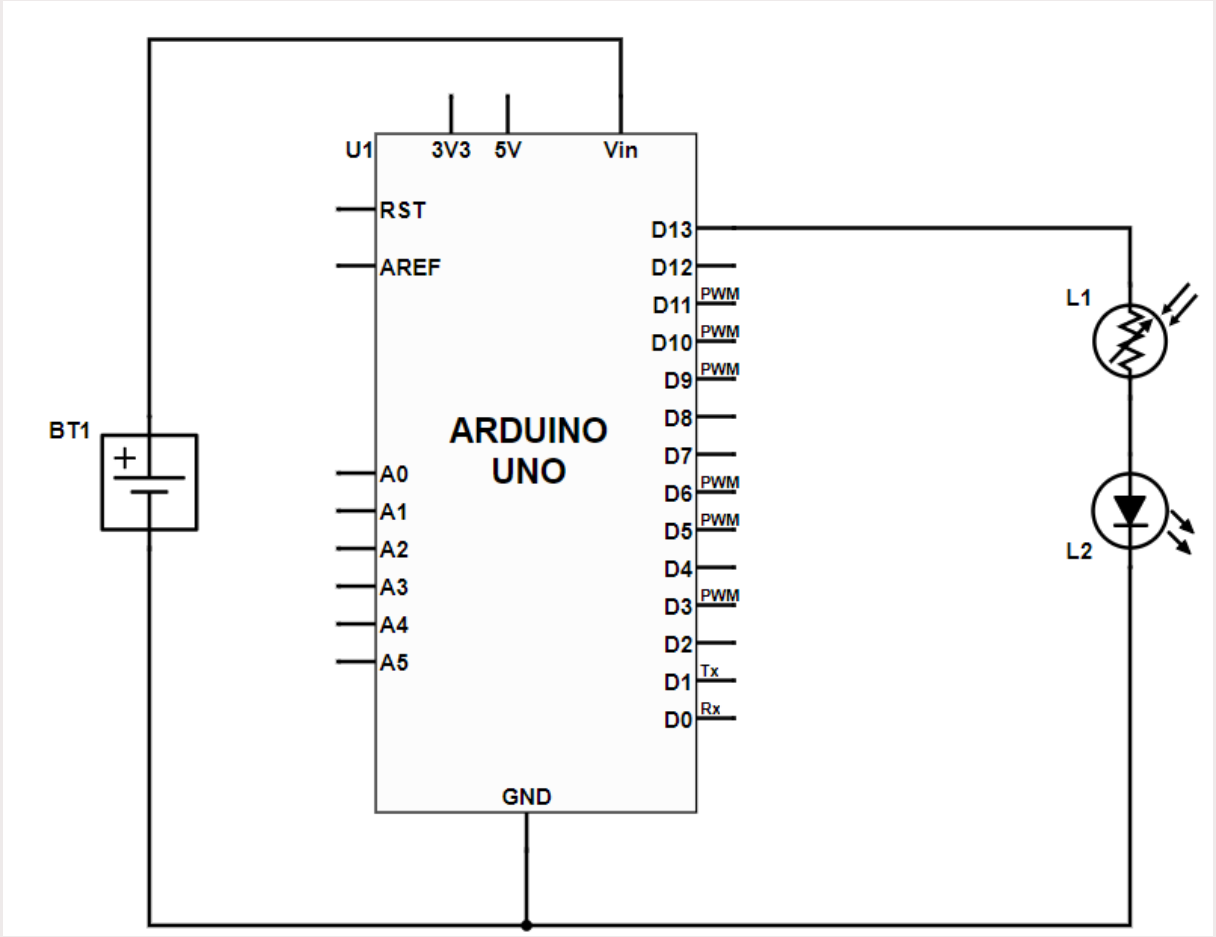
- Photoresistors may require a few milliseconds or more to fully respond to changes in light intensity and may require a few seconds to return to their normal dark resistance once light is removed
- Photoresistor acts as the light sensing element in a simple light meter
  - when dark - the photoresistor is very resistive, and little current flows through the series loop; the meter is at its lowest deflection level
  - When an increasingly bright light source is shown on the photoresistor, the photoresistor's resistance begins to decrease, and more current begins to flow through the series loop





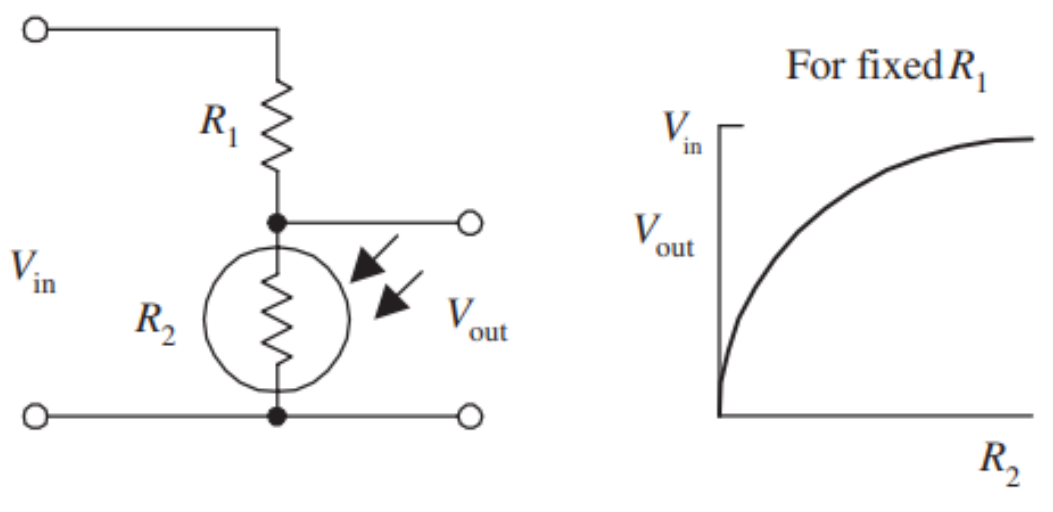
Arduino interfacing with I/O devices

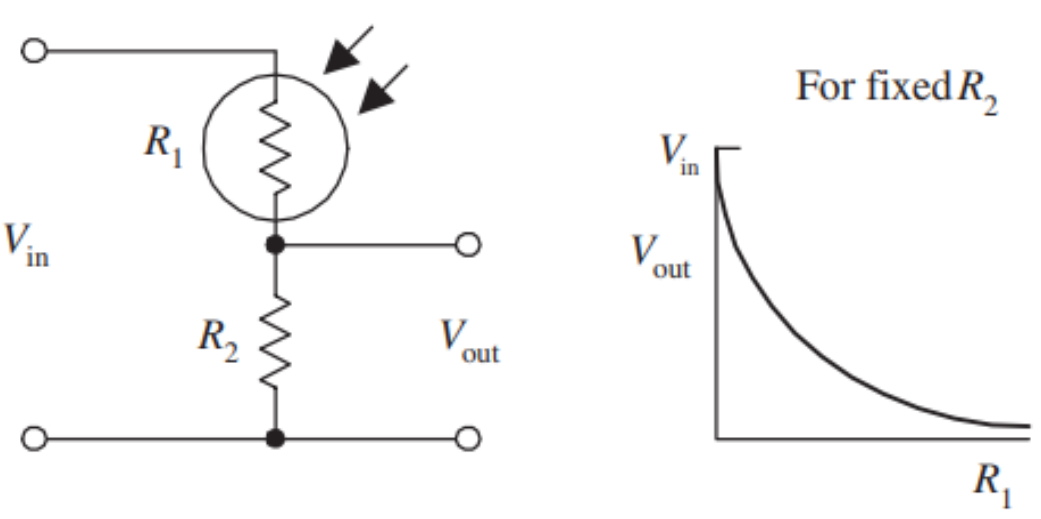
Light Dependant Resistor (LDR)



Arduino interfacing with I/O devices

Light Dependant Resistor (LDR)

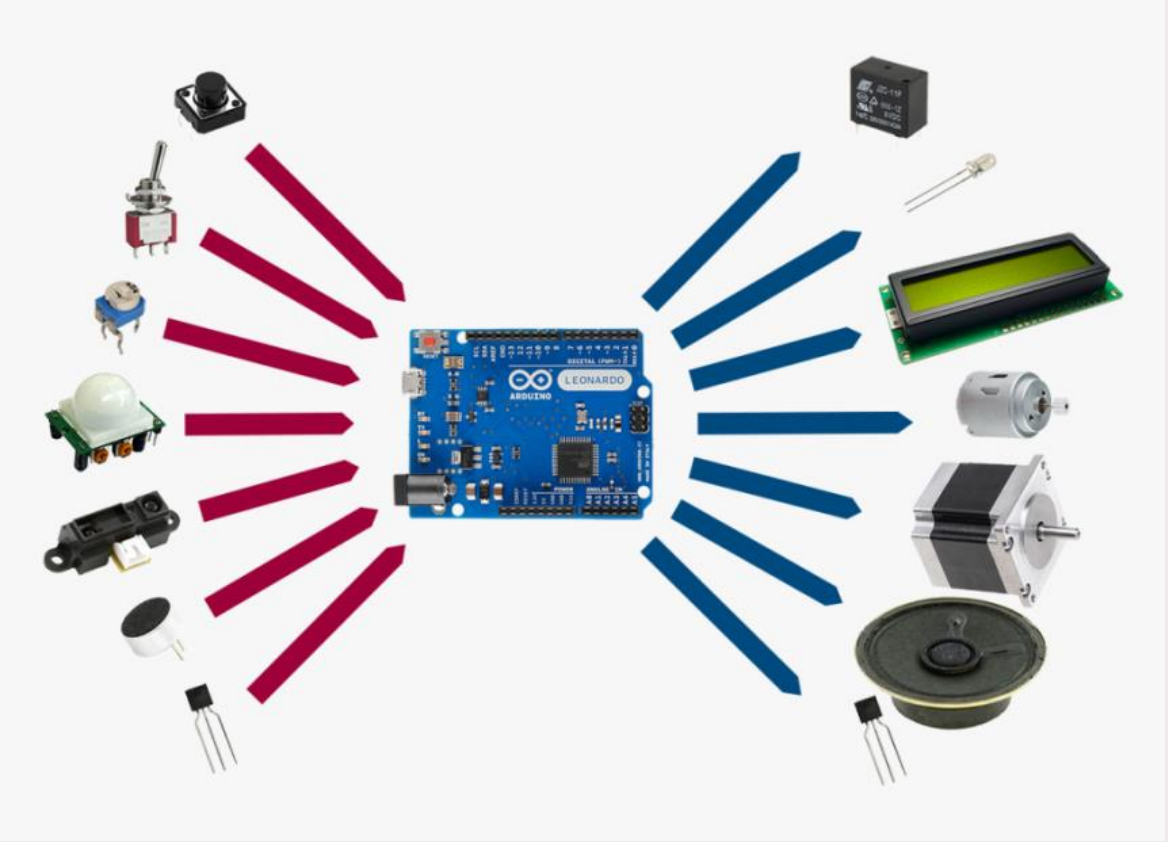




Arduino interfacing with I/O devices

When using block code from basic examples. it must be remembered that:

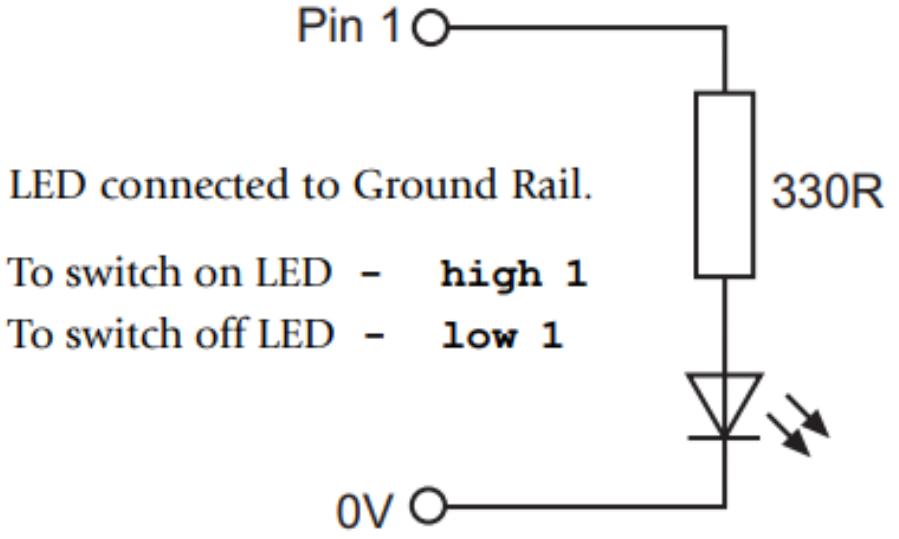
- Each pin should be set up as an input or output before using the code
- If the hardware pins are changed from those given in the circuit diagrams it will be necessary to modify the pin numbers in the code

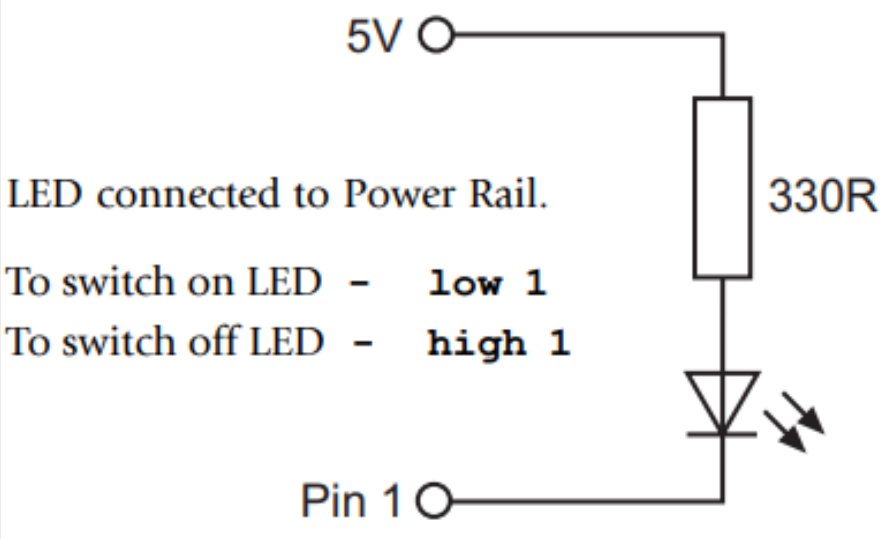


Arduino interfacing with I/O devices

Light Emitting Diode (LEDs)

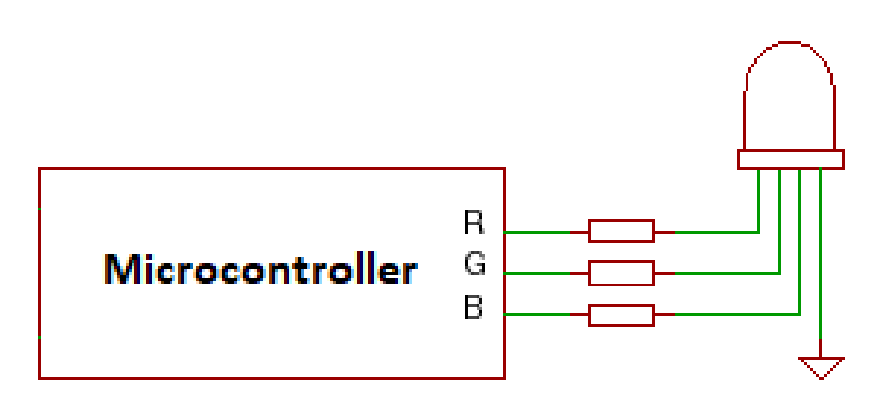
The Microcontroller can sink (“absorb”) or source (“give out”) a small amount of current, which means that an LED can be connected directly to the output pin. A series resistor (value 330Ω) is also required to limit the current.

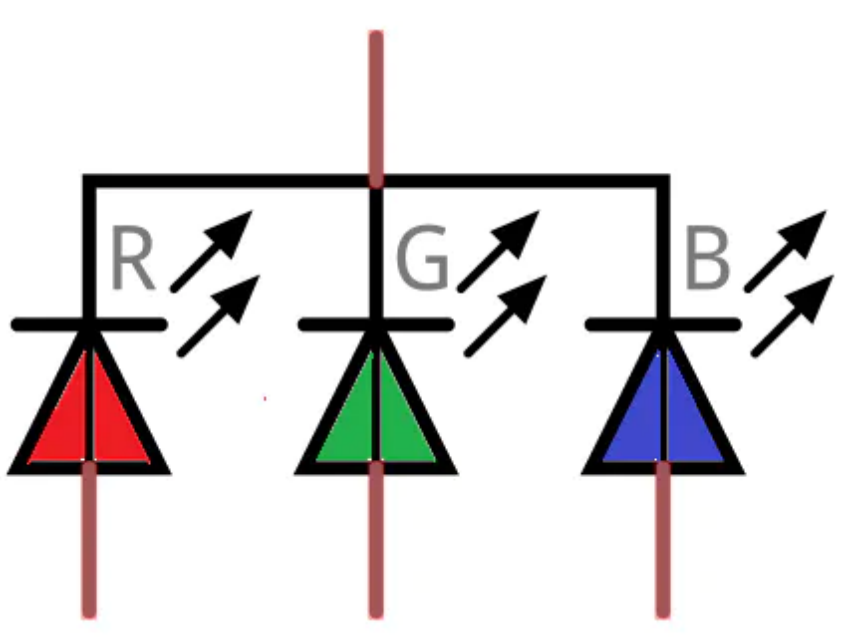




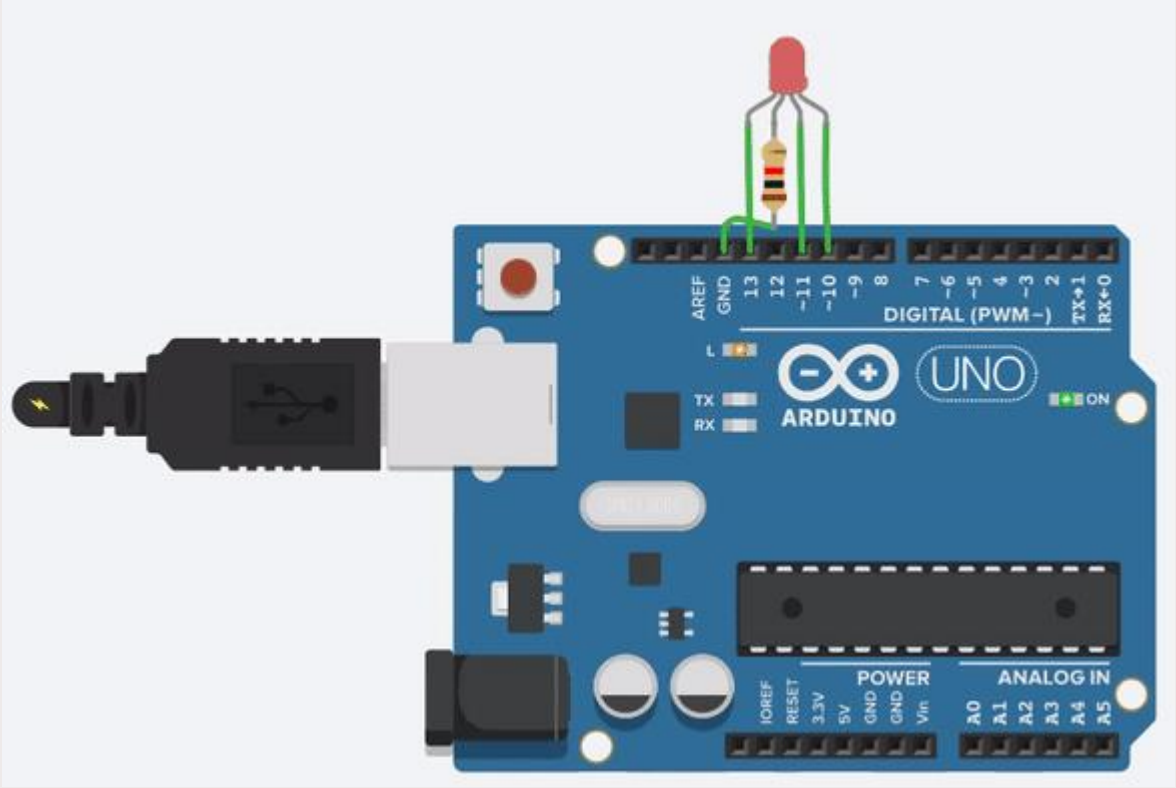
Arduino interfacing with I/O devices

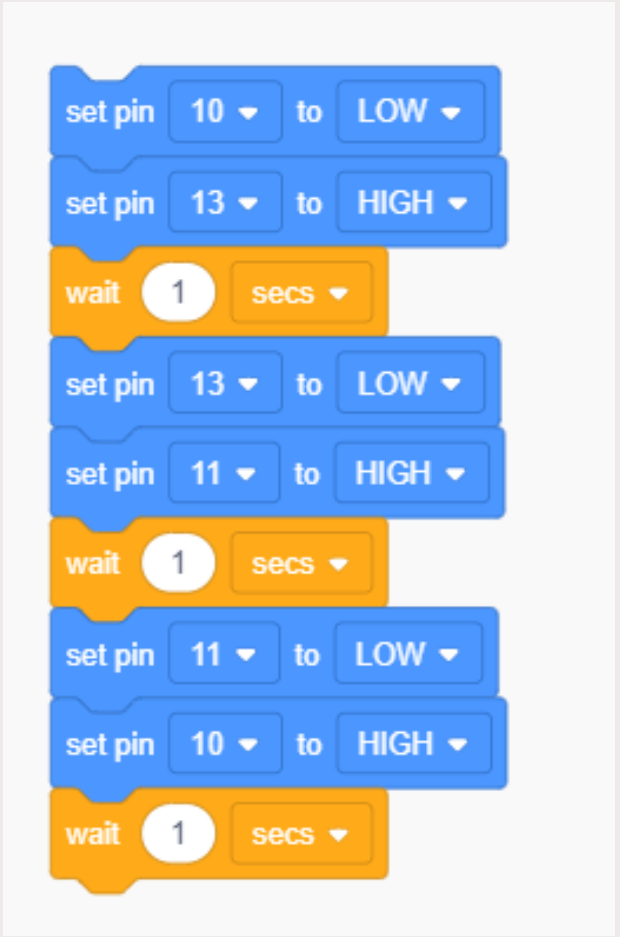
- The RGB LEDs consists of three different LED's, from the name you can guess that these LEDs are red, green and blue
- We can obtain many other colors by mixing up these colors
- The Arduino microcontroller has an Analog pins varying from 0 to 255 which will help us in obtaining different colors for RGB LED





Arduino interfacing with I/O devices



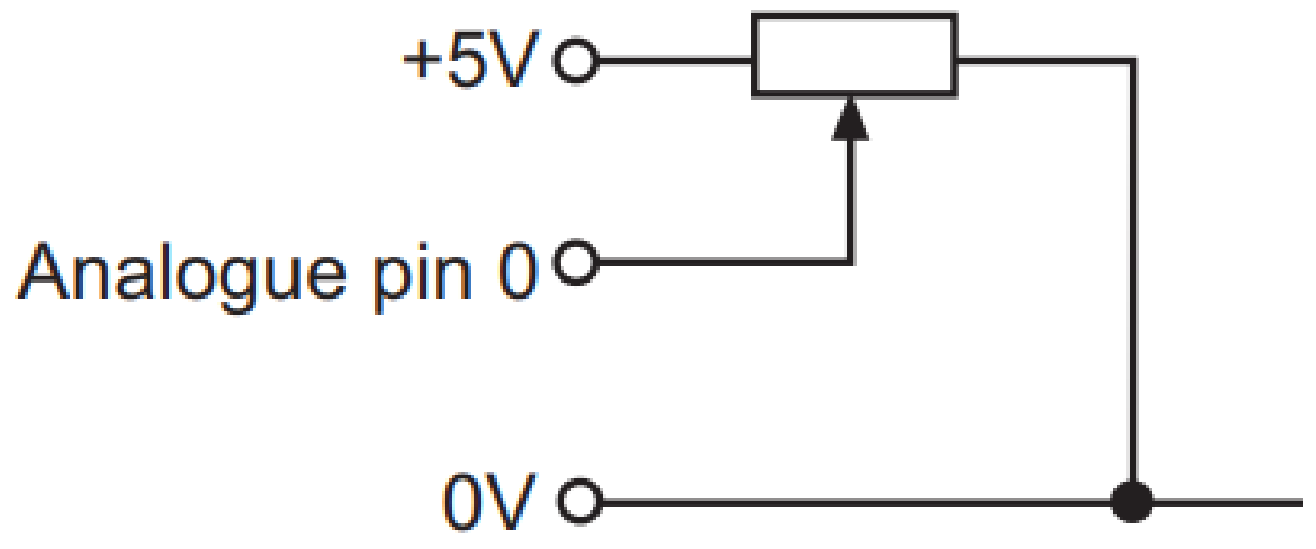


Arduino interfacing with I/O devices

Input Device - Potentiometer

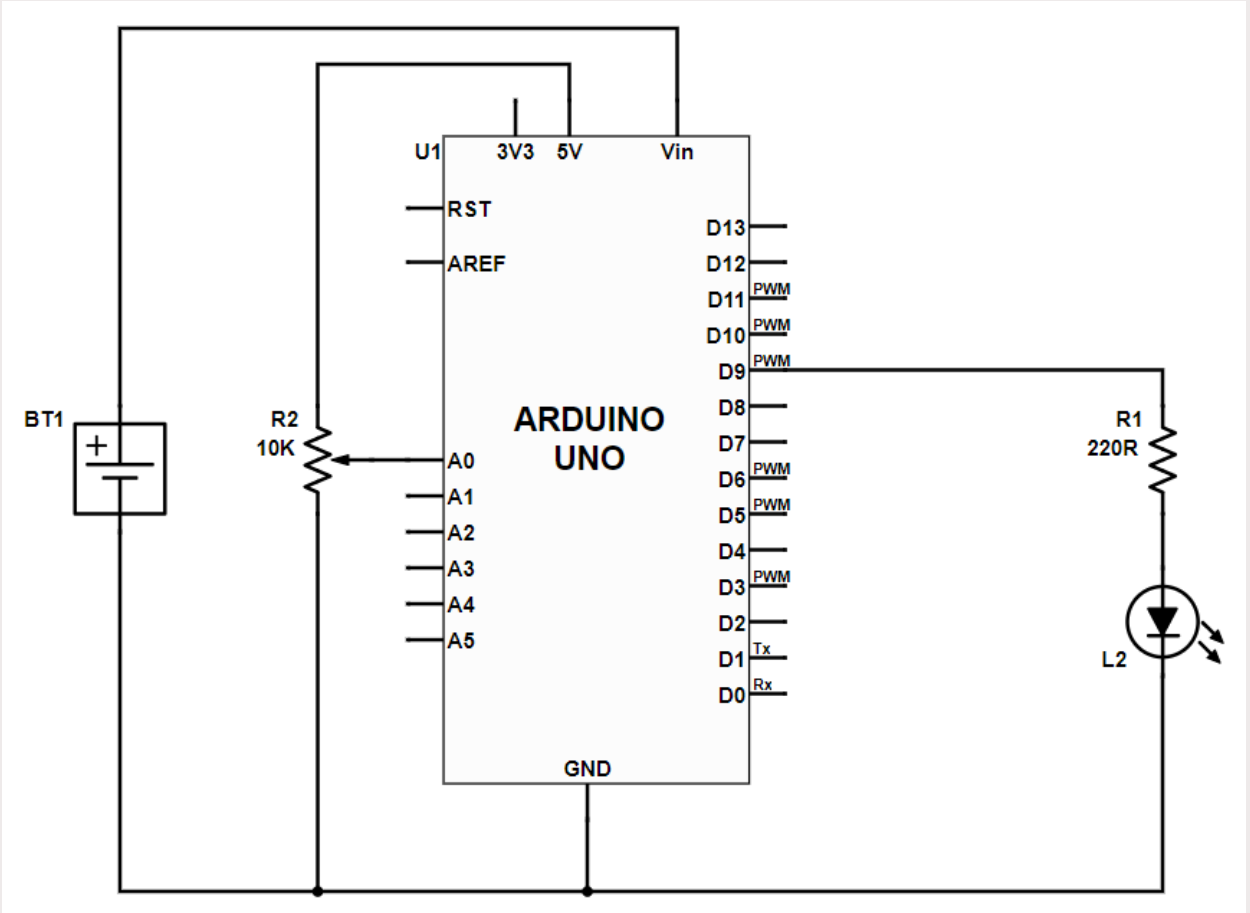
A potentiometer (or 'variable resistor') has a spindle that can be moved to change the resistance value of the potentiometer. This can be used to measure rotational or linear movement

The Analog pins used to measure the value of the resistance by carrying out an Analog to Digital Conversion. The value of the resistance is given a 'value' between 0 and 1023 which is then stored in a variable.



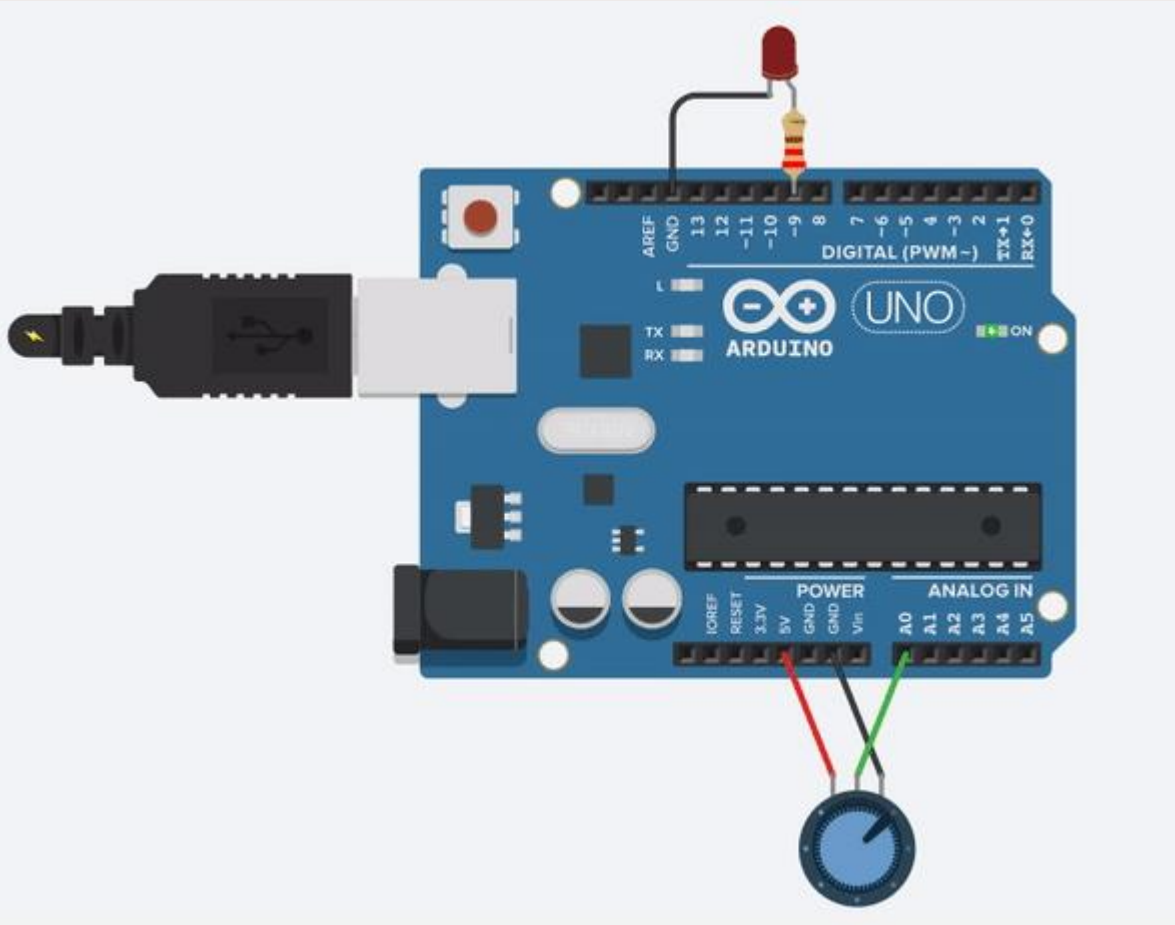
Arduino interfacing with I/O devices

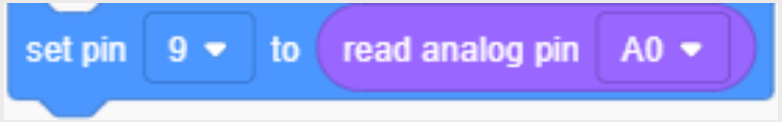
Input Device - Potentiometer

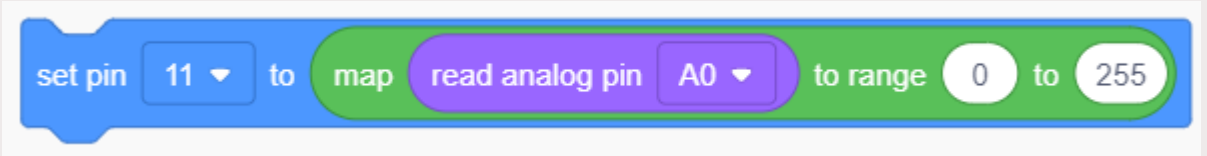


Arduino interfacing with I/O devices

Input Device - Potentiometer



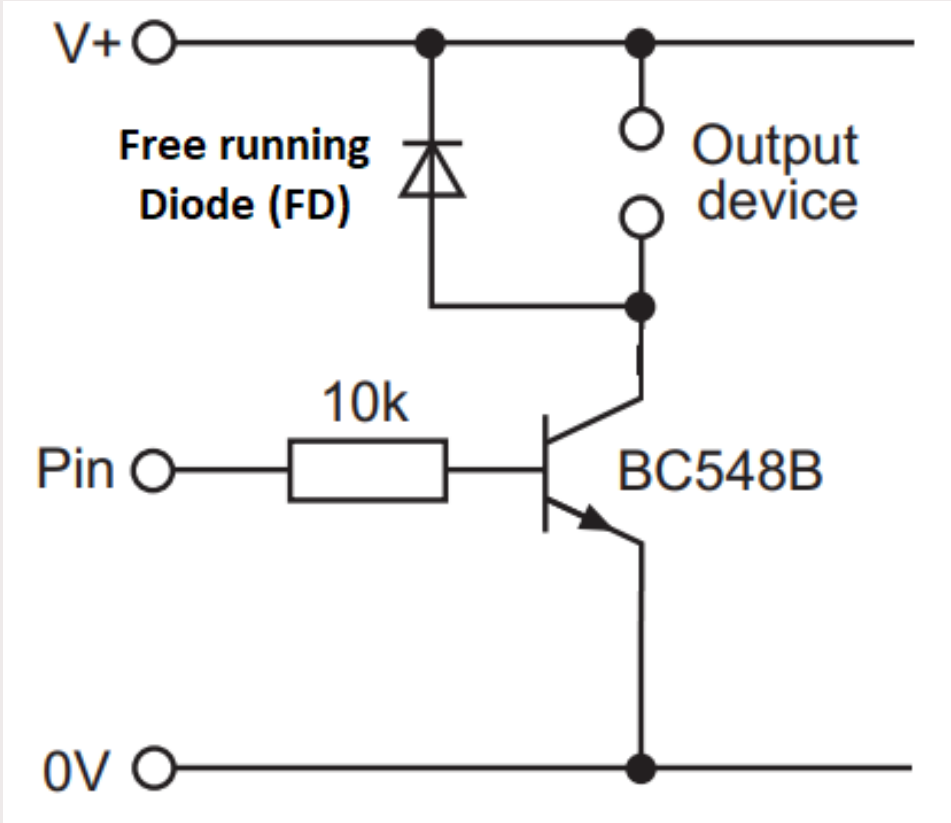




Arduino interfacing with I/O devices

Transistor Interfacing Circuit

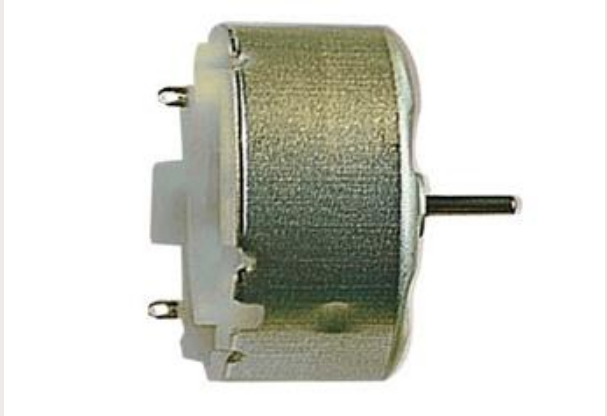
- The transistor can be switched ON or OFF by changing the base.
- There are a few applications of switching circuits operated by transistors
- Considered NPN transistor to explain few applications which are using transistor switch



Arduino interfacing with motors

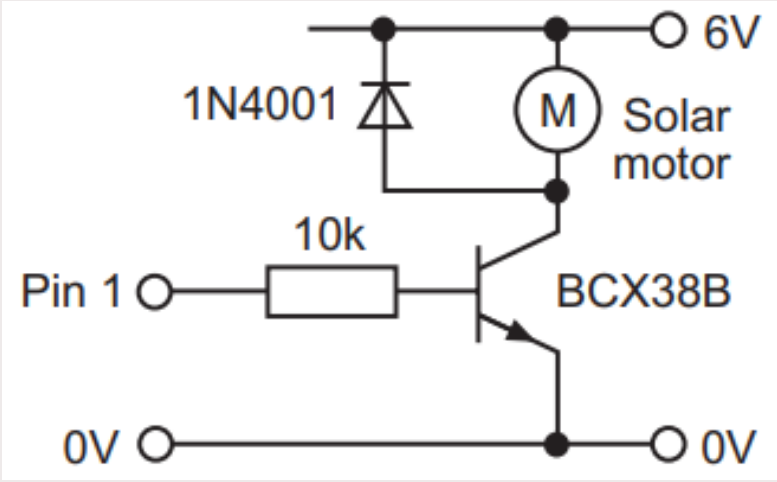
DC Motors

Many projects require the use of a DC motor to create rotational movement. There are several ways motors can be interfaced with microcontroller



Using transistor drive

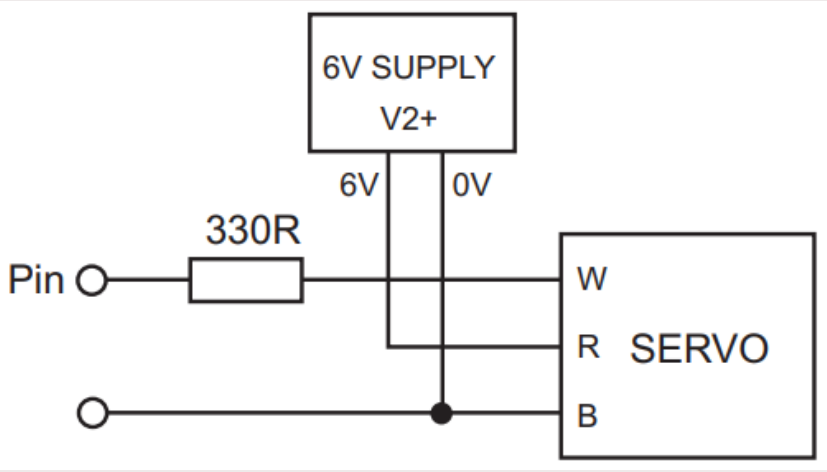
- Circuit uses a transistor to switch the motor on and off.
- It will work with ‘solar’ motors, DC Motor
- This type of motor requires very less current and introduces a lot of electrical ‘noise’ on to the power rails



Arduino interfacing with motors

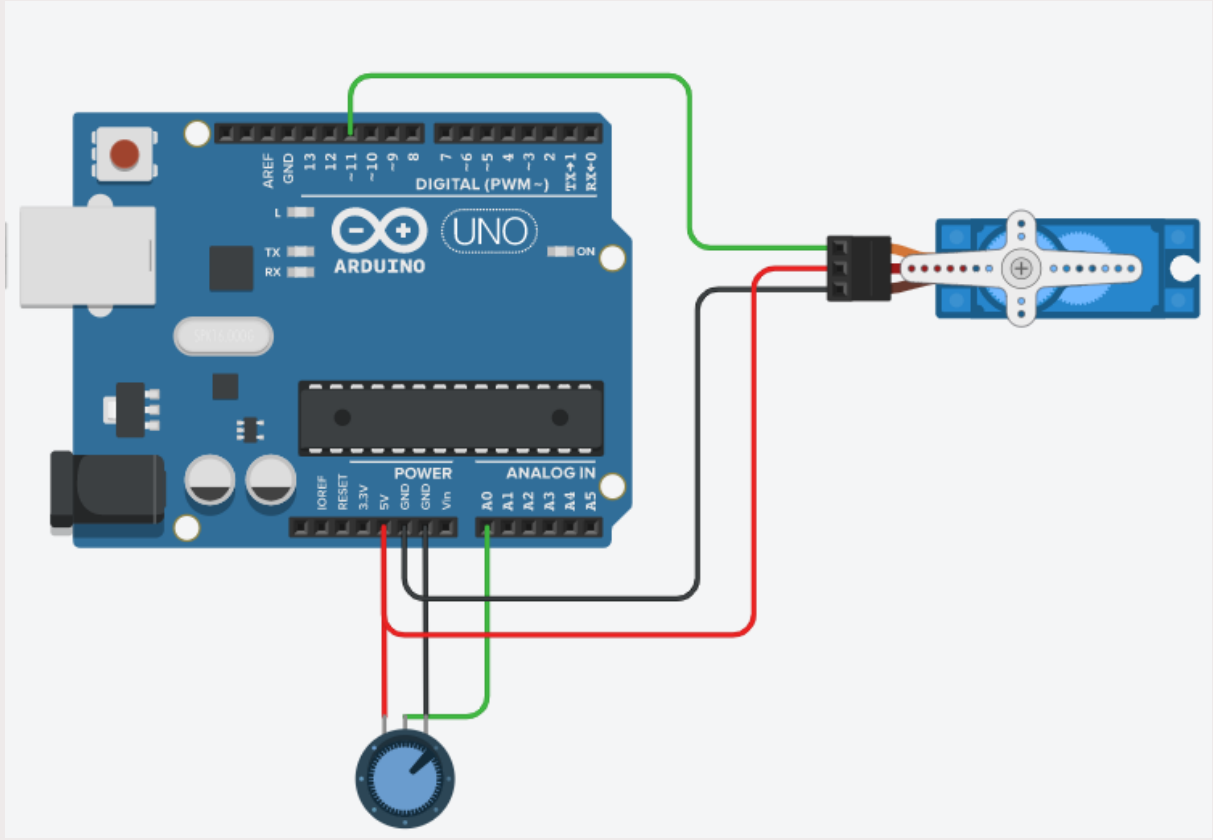
Control Servo motor

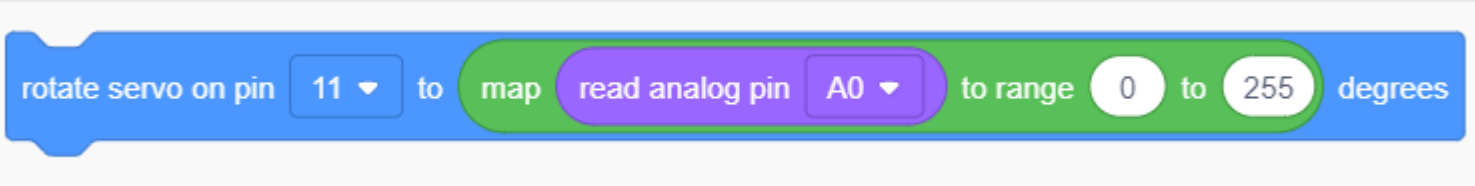
- Servos are used in most industrial robots and planes to control the mechanism
- They are accurate devices that always rotate the same amount for a given signal, and so are ideal for use in many automated machines
- A typical servo has just three connection wires, normally red, black and white (or yellow)
- The red wire is the 5V supply, the black wire is the 0V supply, and the white (or yellow) wire is for the positioning signal



Arduino interfacing with I/O devices

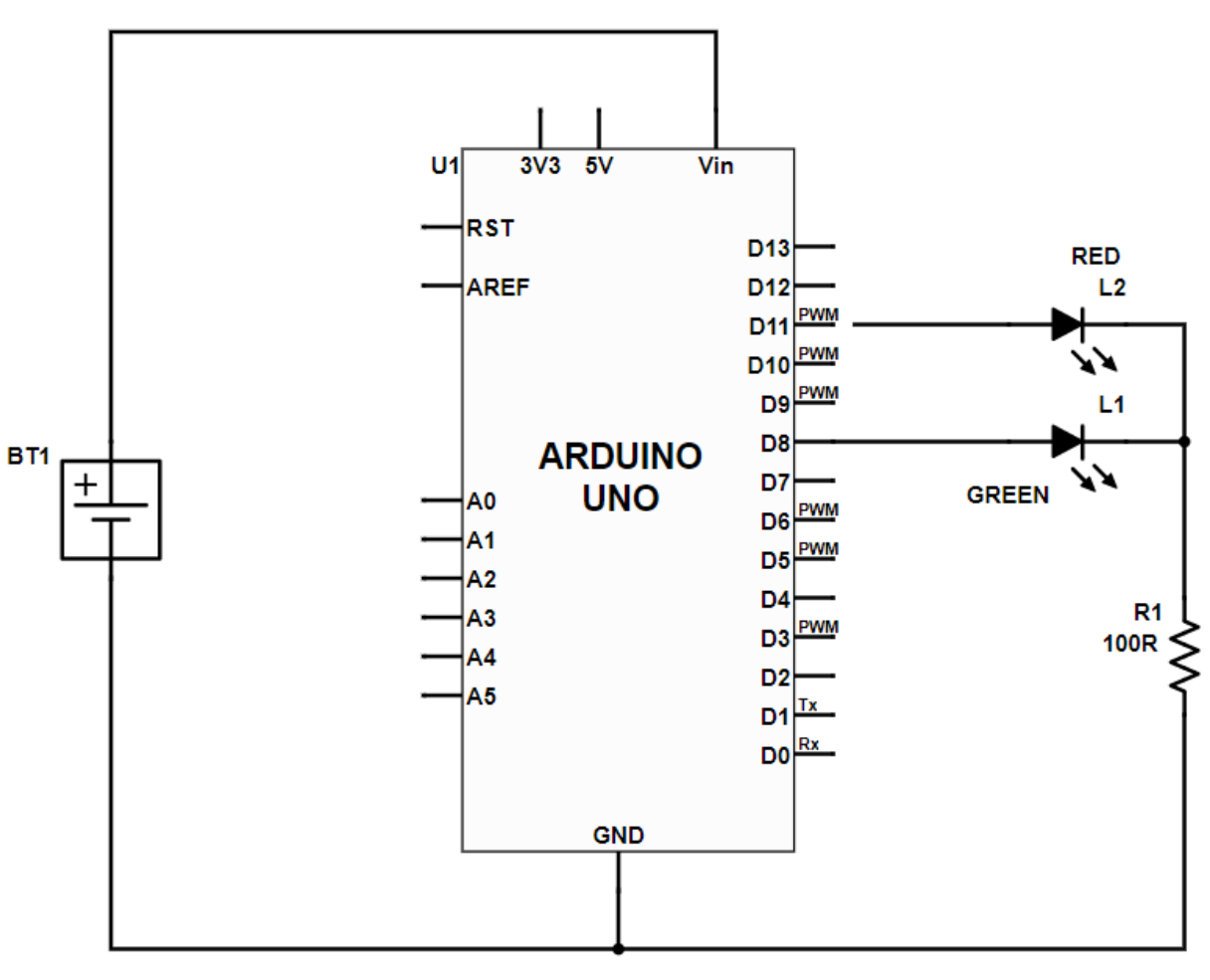
Servo Control





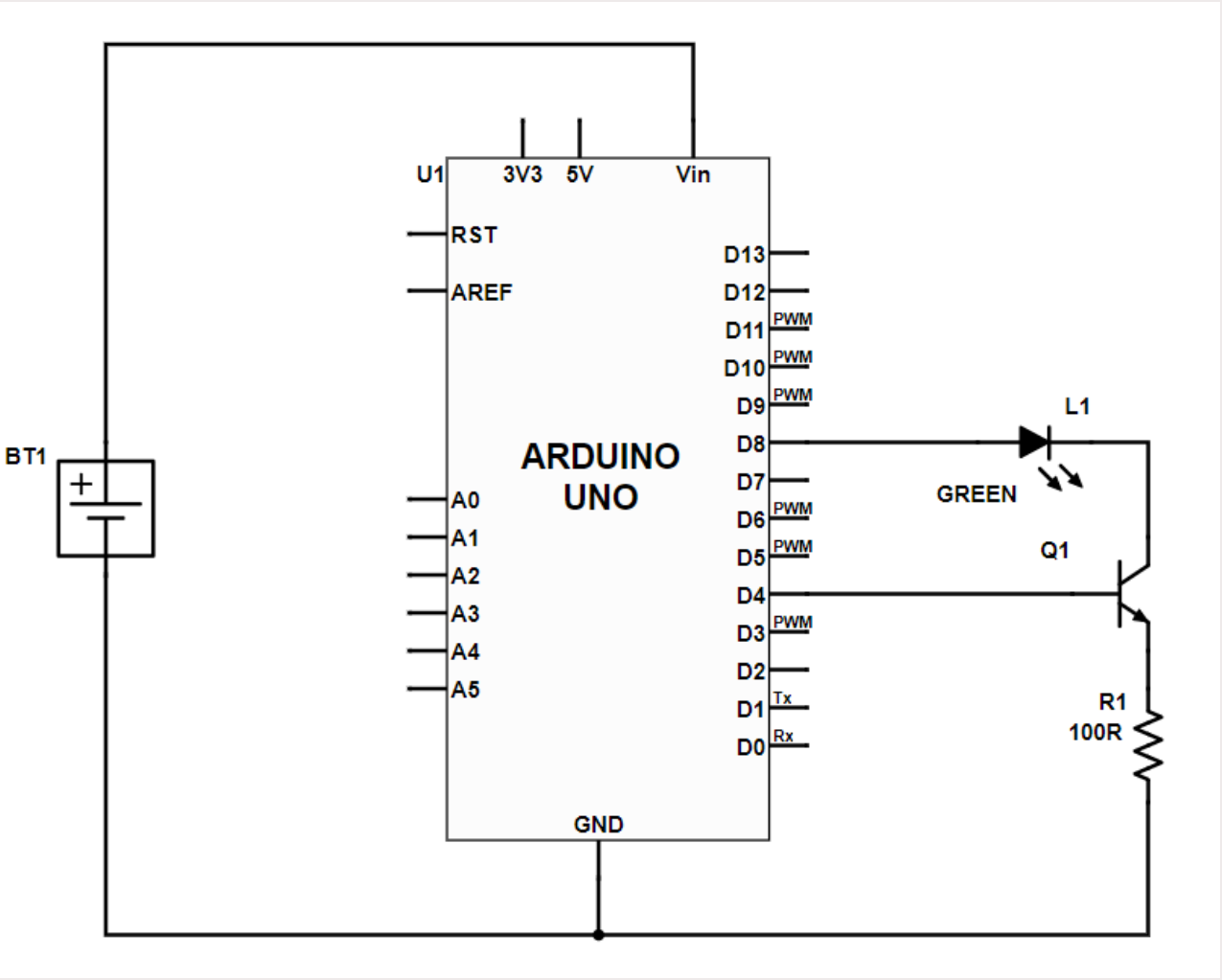
Arduino interfacing with I/O devices

LED Control



Arduino interfacing with I/O devices

LED Control



Types of sensors

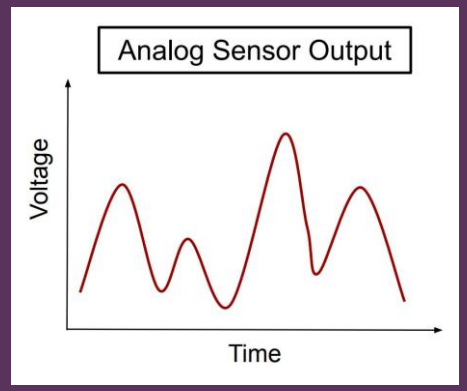
Classification of sensors

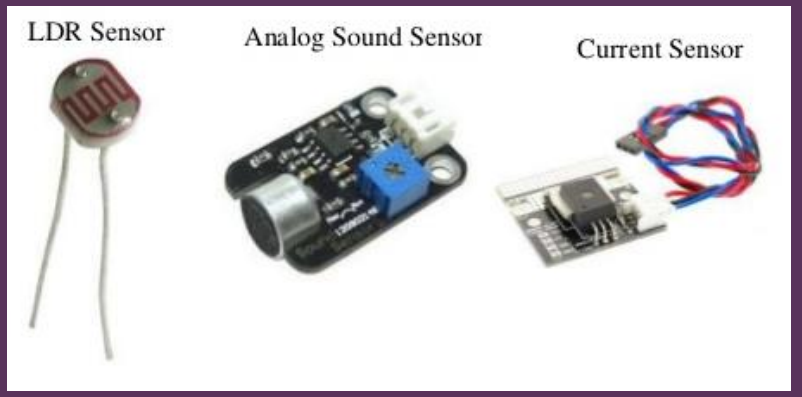
- **Analog Sensor**

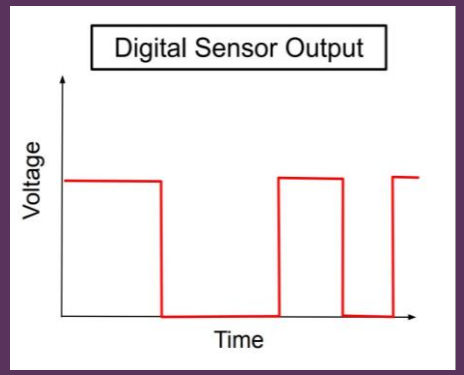
  Analog sensors are those that convert the quantity being measured into an Analog signal, one whose values change continuously along an interval.

- **Digital Sensor**

  Digital sensors are those whose output is a digital signal, that is a logic 0 or logic 1. An IR sensor is an example of a digital sensor

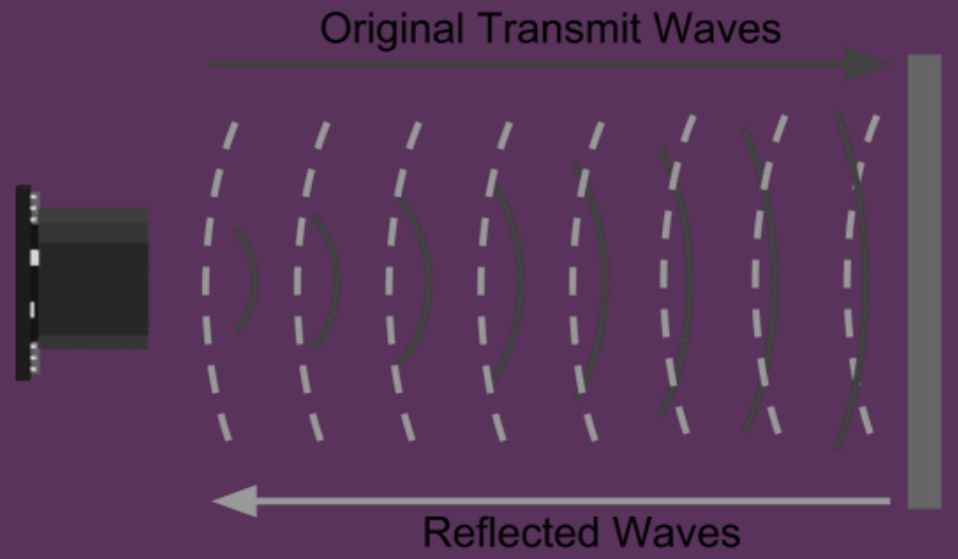






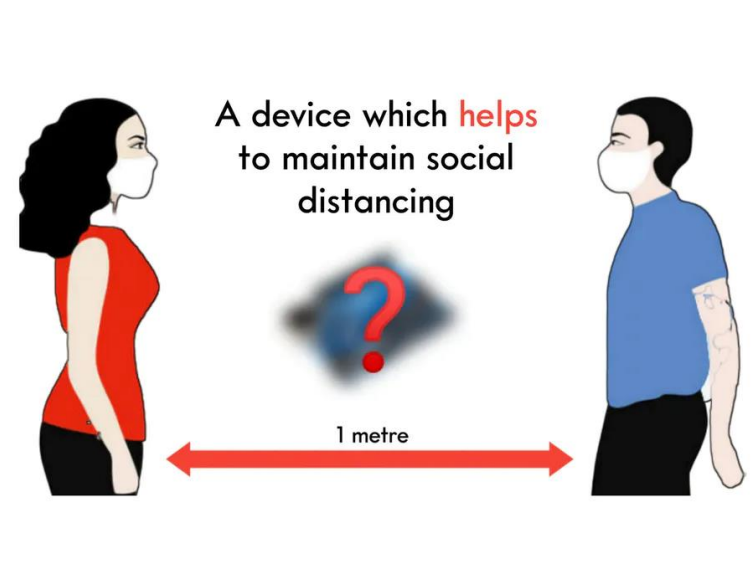
Ultrasound sensor

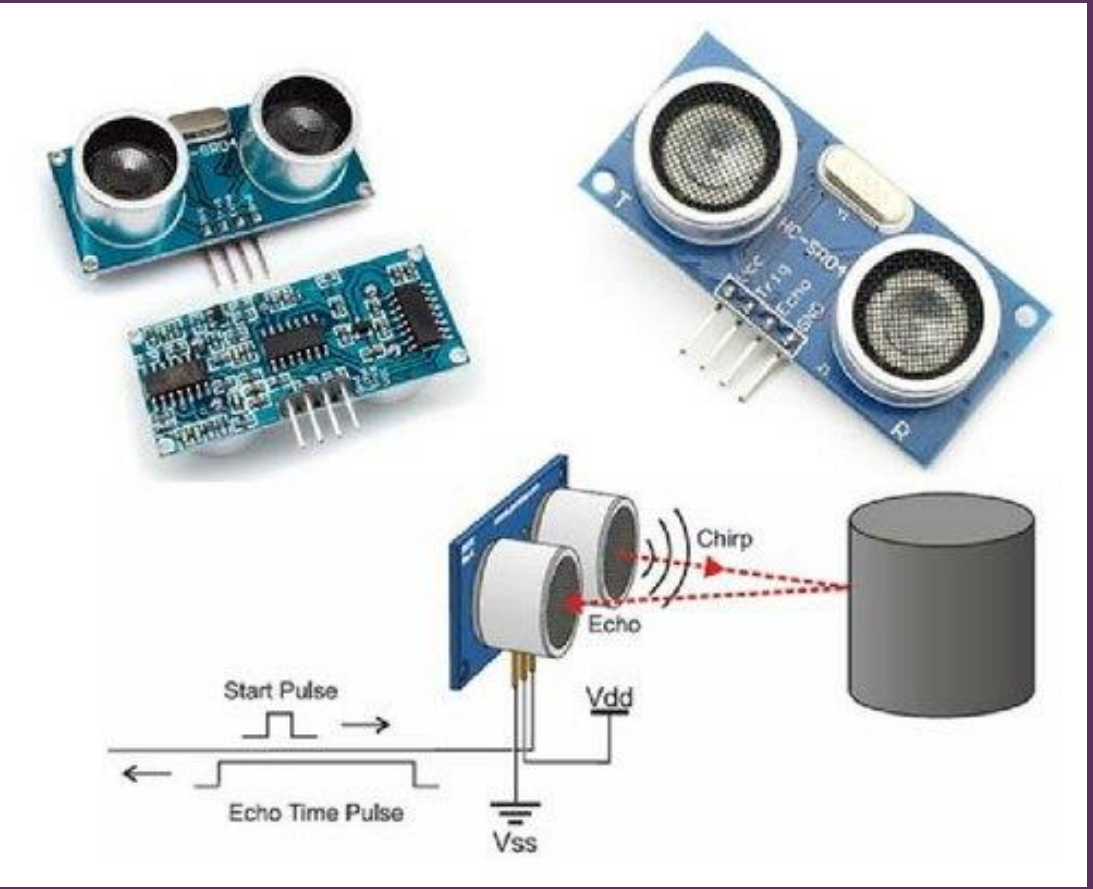
- An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves and converts the reflected sound into an electrical signal.
- Ultrasonic waves travel faster than the speed of audible sound
- Ultrasonic sensors have two main components: the transmitter and the receiver



Ultrasound sensor Electronic Components

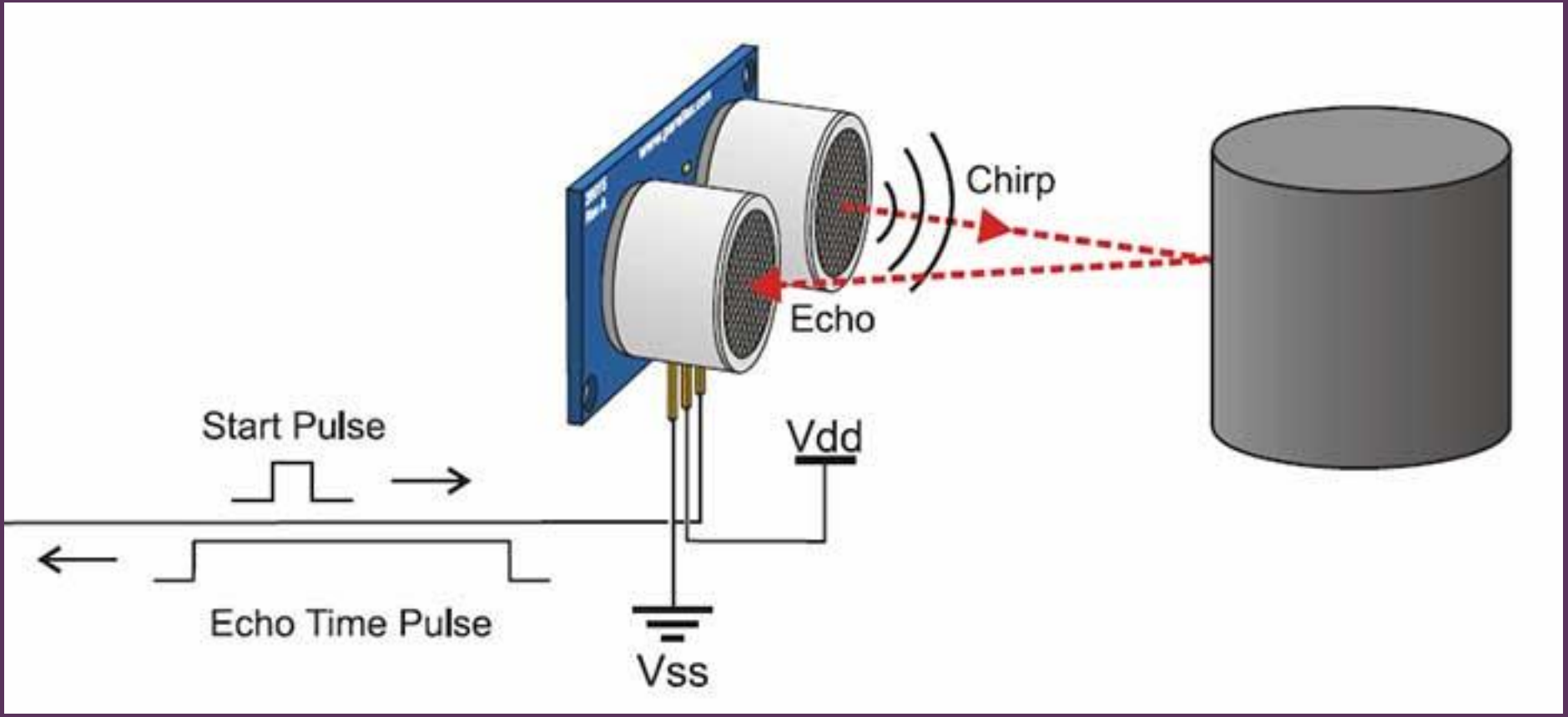
- An ultrasonic sensor can measure the distance of a target object.
- This feature is used here in this experiment to alert somebody if an object comes too close.
- For instance, it can be used to build a COVID Social Distancing Alarm.





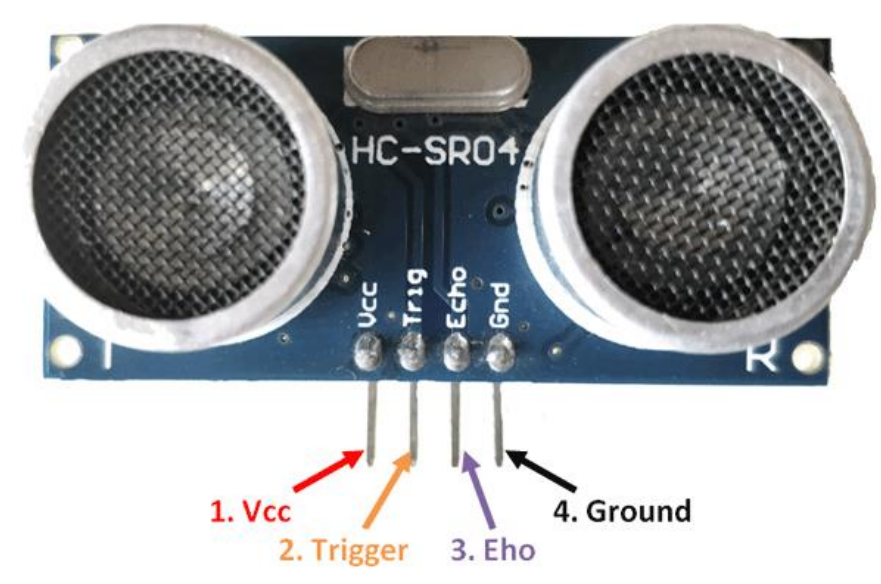
Ultrasound sensor Electronic Components

- HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc.



Ultrasound sensor Electronic Components

Pin Configuration



Pin Number	Pin Name	Description
1	VCC	The VCC pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period which will be equal to the time taken for the US wave to return to the sensor.
4	Ground	This pin is connected to the Ground of the system.

Ultrasound sensor Electronic Components

How to use the HC-SR04 Ultrasonic Sensor

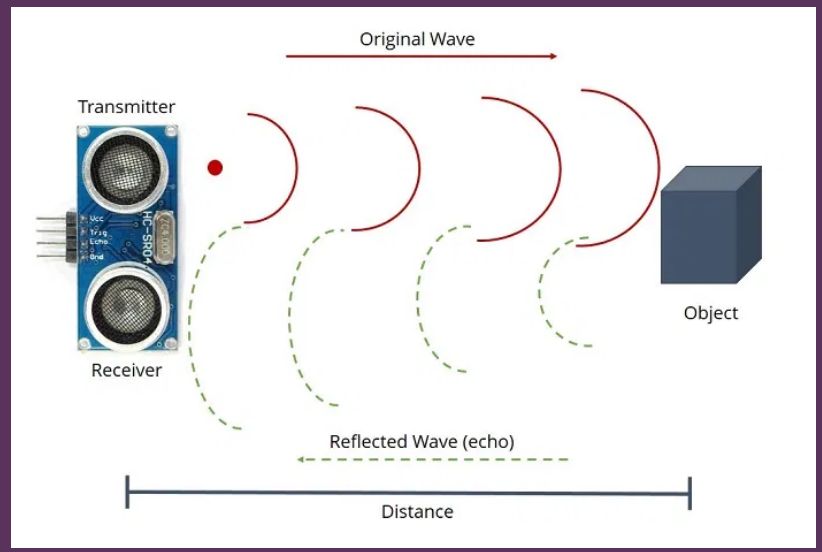
- The word 'sonic' refers to sound. Ultrasonic waves are those that travel much faster than sound.
- The sensor emits ultrasonic waves and once these are reflected back, the sensor lets us know using one of its pins going low.
- We can then measure the time difference it took for the wave to return and use this to calculate the signal.
- This is done using the formula,

  D = ½ * T *V

  where, D is the distance

  T is the time and

  V is the speed of sound (343 m/s in air).



It needs to be divided by 2 because the sound waves first get emitted and then get reflected back.

Ultrasound sensor Electronic Components

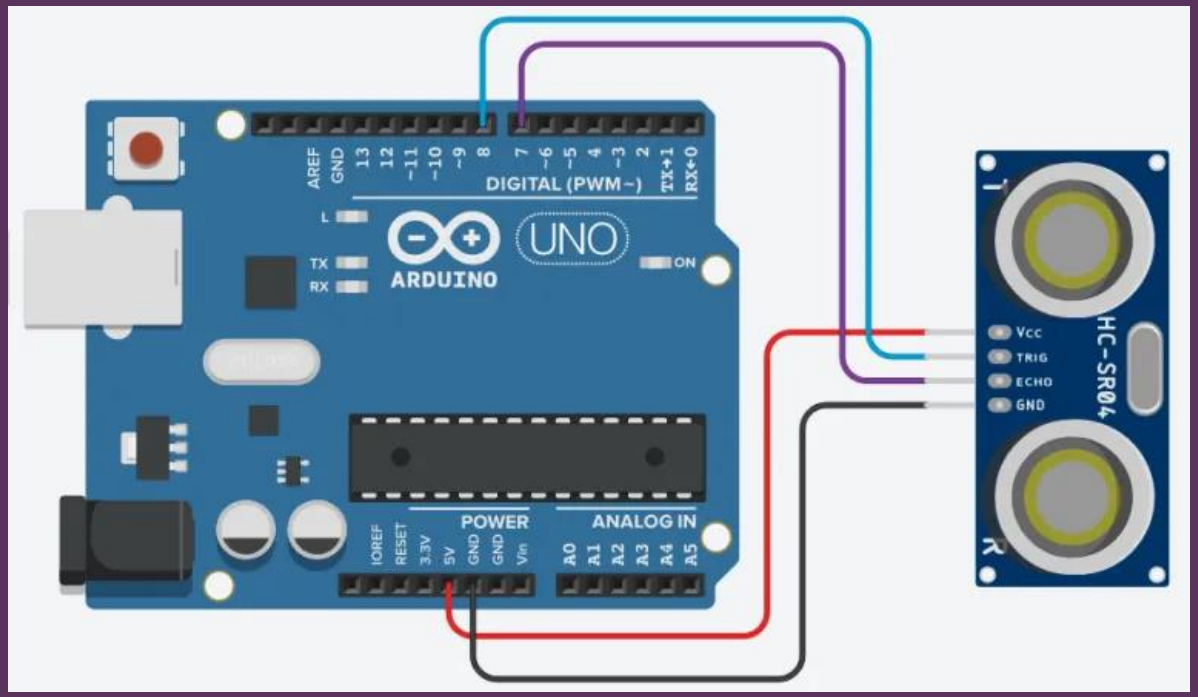
Ultrasound Sensor Applications

- Robotic sensing
- Liquid level control
- Counting people / people detection
- Presence detection

Arduino with sensors

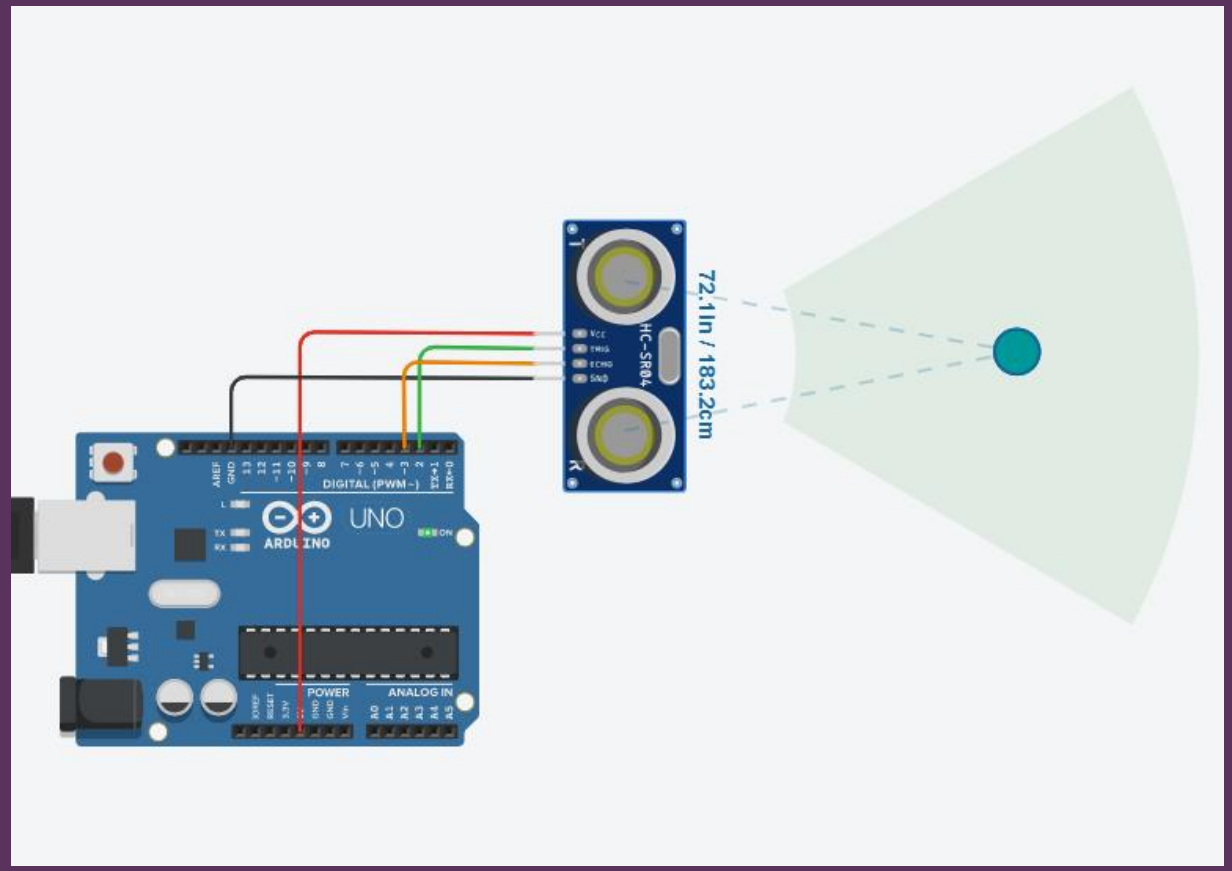
Interfacing ultrasound sensor with Arduino

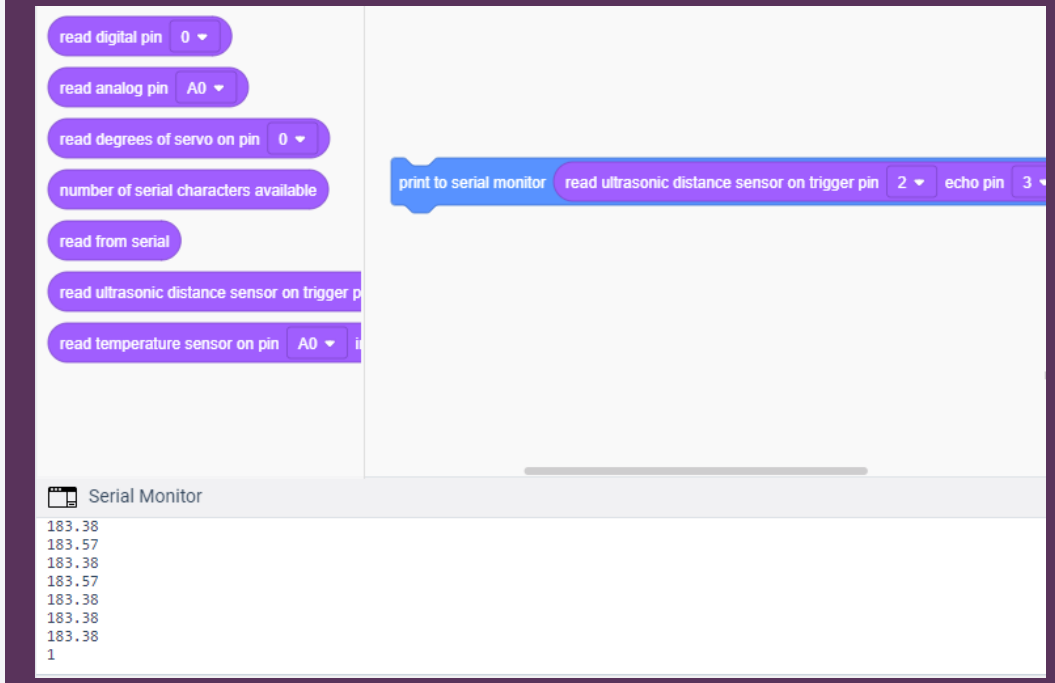
- An ultrasonic sensor is an electronic component used to find the range of a targeted object by emitting ultrasonic waves (sound waves).
- This sensor mainly consists of two parts, a transducer that produces ultrasonic sound waves and another that listens for its echo.

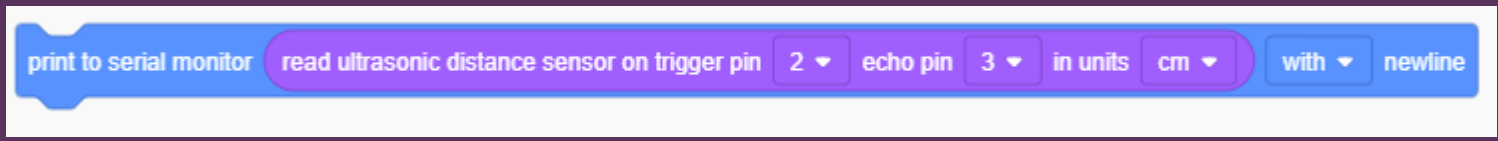


Arduino with sensors

Interfacing ultrasound sensor with Arduino

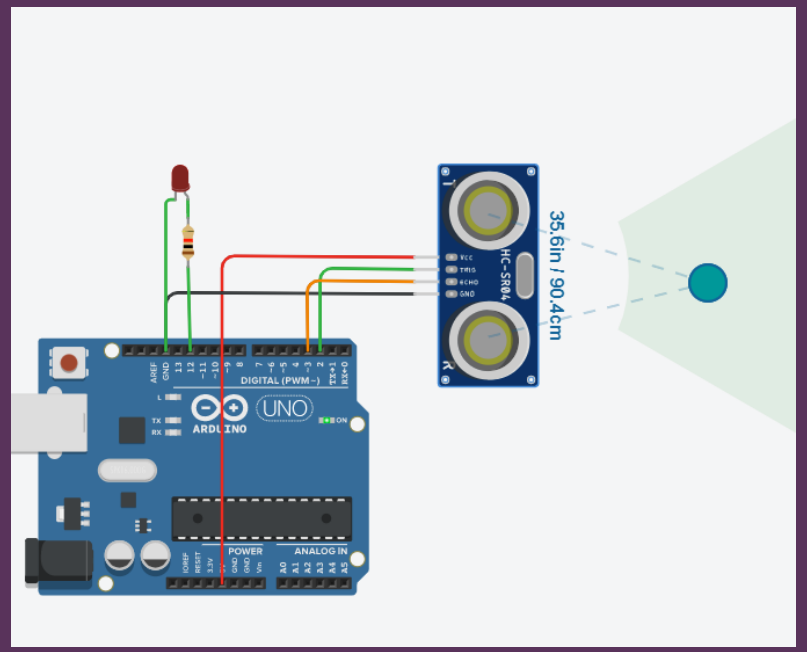


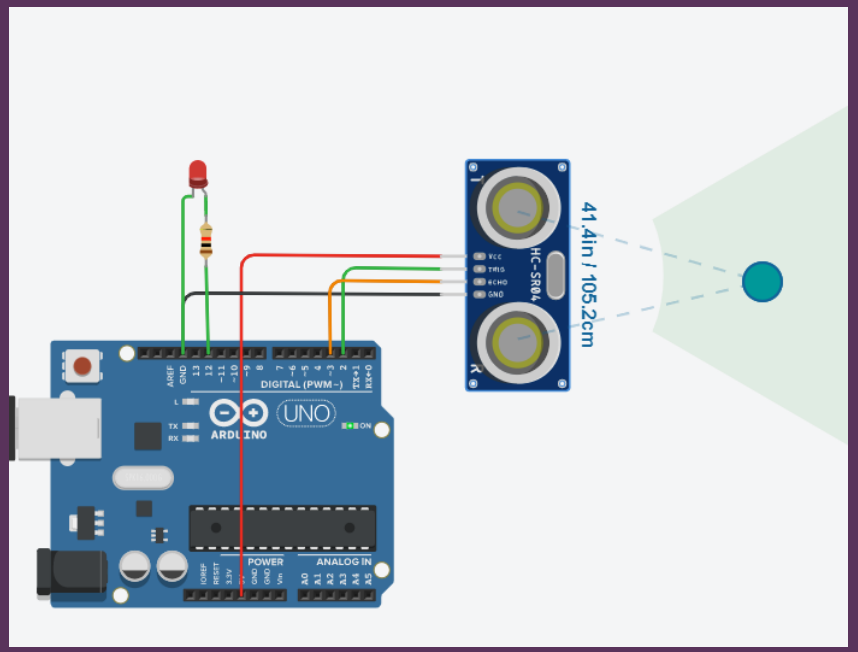


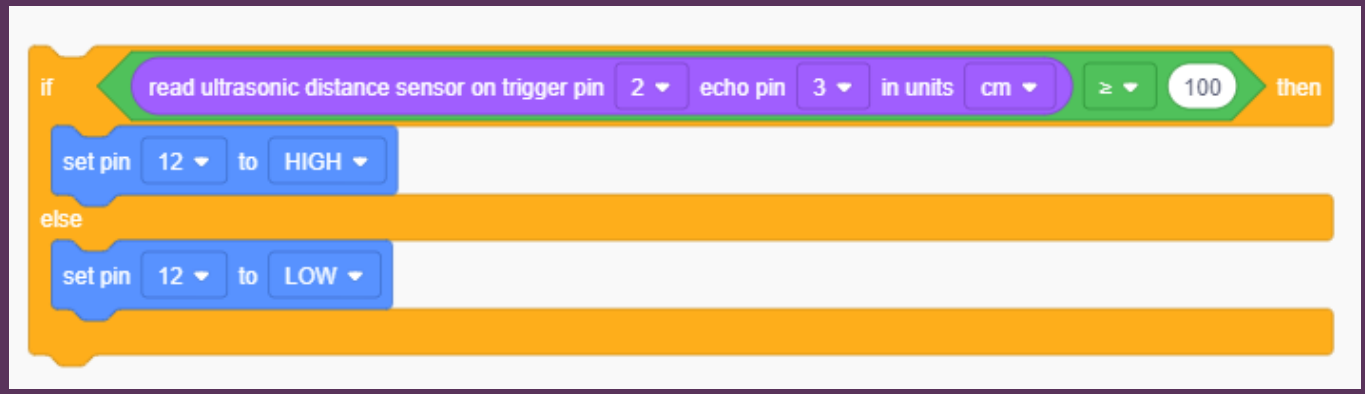


Arduino with sensors

Interfacing ultrasound sensor with Arduino







Arduino with sensors

Interfacing ultrasound sensor with Arduino