Home Experiment Literature

1. Micro Controller Unit (MCU)

A microcontroller (sometimes called an MCU or Microcontroller Unit) is a single Integrated Circuit (IC) that is typically used for a specific application and designed to implement certain tasks. Products and devices that must be automatically controlled in certain situations, like appliances, power tools, automobile engine control systems, and computers are great examples, but microcontrollers reach much further than just these applications.

Essentially, a microcontroller gathers input, processes this information, and outputs a certain action based on the information gathered. Microcontrollers usually operate at lower speeds, around the 1MHz to 200 MHz range, and need to be designed to consume less power because they are embedded inside other devices that can have greater power consumptions in other areas.

A microcontroller can be seen as a small computer, and this is because of the essential components inside of it; the Central Processing Unit (CPU), the Random-Access Memory (RAM), the Flash Memory, the Serial Bus Interface, the Input/Output Ports (I/O Ports), and in many cases, the Electrical Erasable Programmable Read-Only Memory (EEPROM).

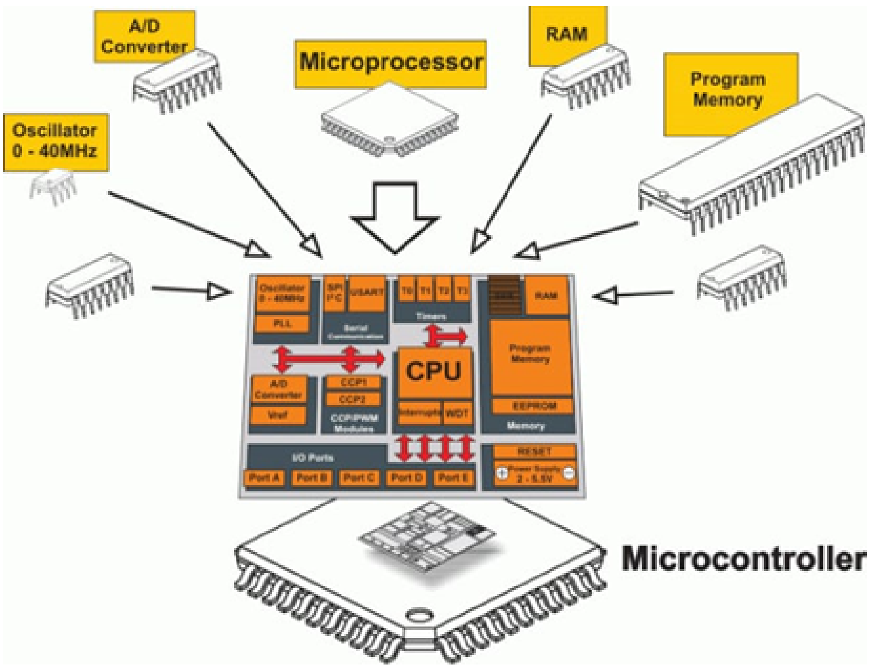

Figure: Parts of a microcontroller. (Source: Max Embedded)

2. Arduino Development Board and Environment

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read inputs, amount of light on a sensor, a press on a button, or the arrival of a Twitter message and turn this into an output, for example, activating a motor, turning on an LED or publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do this use the Arduino programming language and the Arduino Software (IDE).

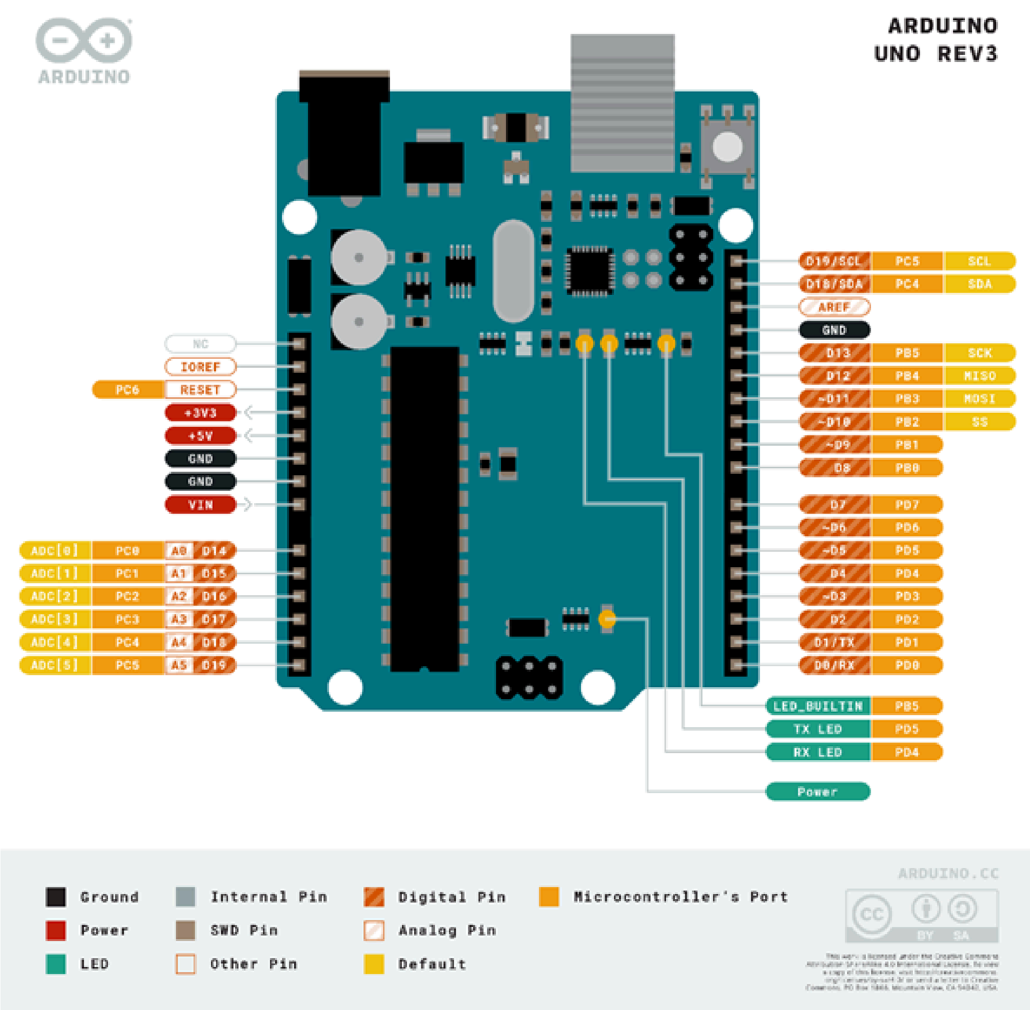

Figure: Arduino Pinout (Source: arduino.cc)

Arduino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.

Arduino Uno specifications:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz

Figure: Arduino Uno Specifications (Source: arduino.cc)

The Arduino environment (IDE) has extensive support for various microcontrollers and includes a core library, which simplifies the programming of microcontrollers.

Figure: Arduino IDE (Integrated Development Environment)

An introduction to the programming environment is available at
https://www.arduino.cc/en/Guide/Environment
The core function and variables descriptions are available at
https://www.arduino.cc/reference/en/

3. DC Motor

A DC (Direct Current) motor is a type of motor that will cause the motor shaft to rotate around its longitudinal axis when applying an electric current between its terminal pins. Thus, the DC motor is a type of actuator that transforms electrical current into rotational motion.

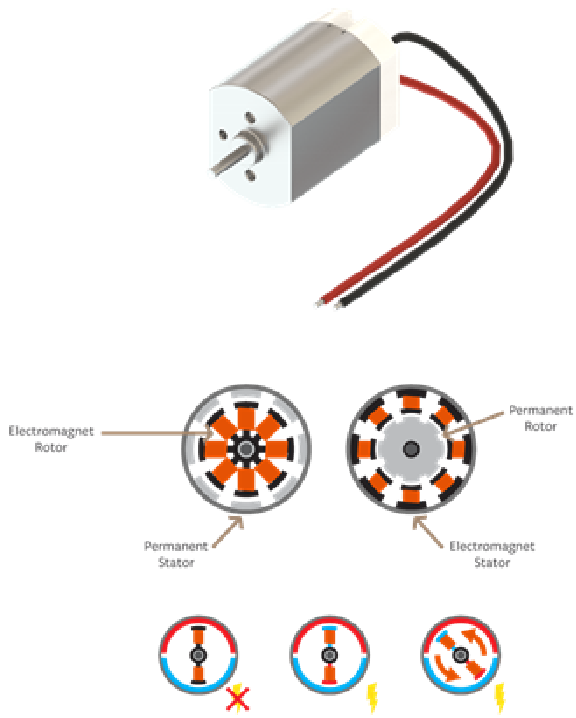

Figure: DC Motor (Source: arduino.cc)

There are two parts inside a motor: the rotor (the shaft is part of this) and the stator. Looking at the cross-section of a motor, you can see that the rotor is the moving part and the stator is the static part. The stator and the rotor use both permanent magnets and electromagnets. Depending on the type of motor, the stator can be a permanent magnet while the rotor is an electromagnet, or vice-versa. Turning on the electromagnet creates attraction and repulsion forces that make the motor spin.

The DC motor spins when we apply DC voltage through its two terminal pins. We can vary the speed of the motor by changing the voltage level. Motors can run in both directions just by reversing the direction of the current.

4. Motor Driver H-Bridge

Most electric motors need a higher voltage than can be provided by a microcontroller. To control a DC motor from a microcontroller, you'll need to use a driver. This can be a transistor, a relay or an H-Bridge. The Arduino Motor Carrier uses H-Bridges to drive the DC motors.

The H-Bridge is an electronic circuit that contains four transistors arranged so that the current can be driven to control the direction of the spin and the angular speed. It is common practice to use PWM (Pulse Width Modulation) signals to control the speed of a motor instead of providing analog voltages.

With the Arduino IDE, if you want to just turn a DC motor on and off, you can use a digitalWrite command with HIGH for ON and LOW for OFF. If you instead want to modify the speed of the motor, you can use the analogWrite command and one of the PWM pins.

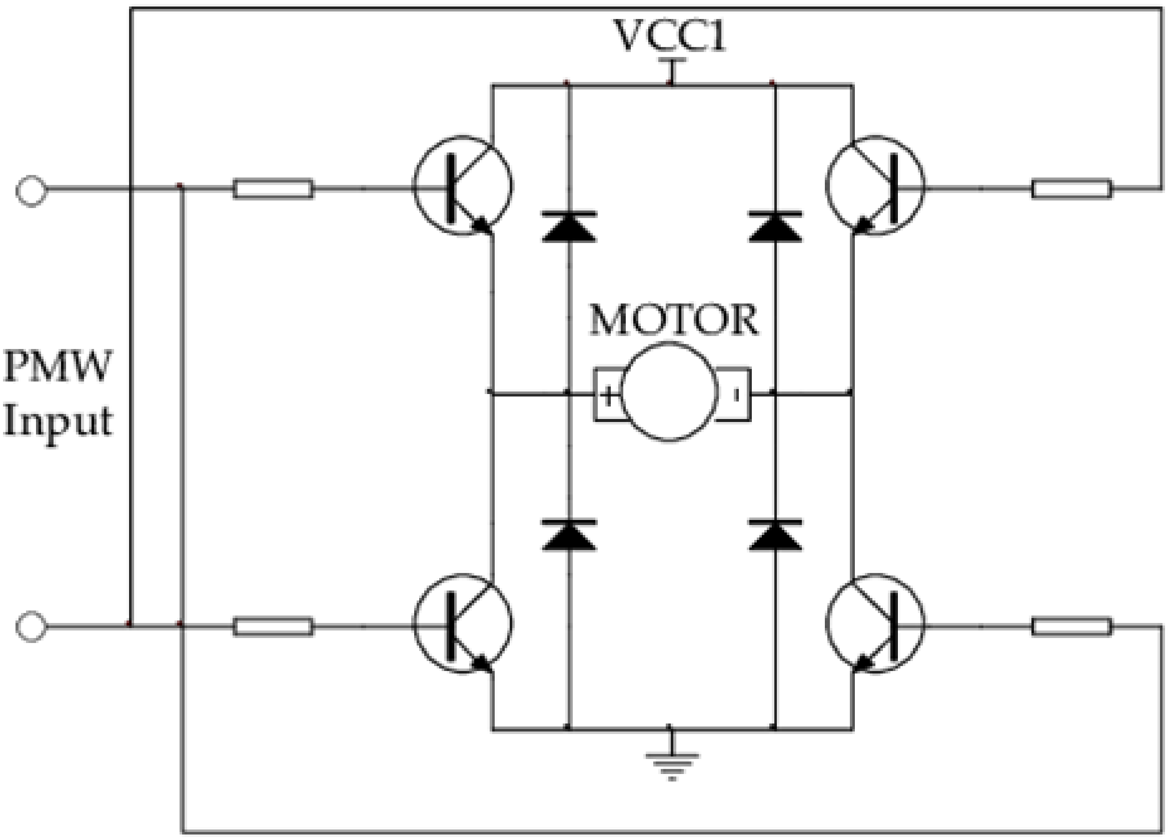

Figure: H-Bridge (Source: arduino.cc)

5. PWM

Pulse Width Modulation, or PWM, is a digital modulation technique commonly used to control the power supplied to electrical devices, like motors. The modulation technique consists of changing the width of a periodic signal's pulse. The width of the pulse is referred to as the duty-cycle and goes from 0% (minimum width) to 100% (maximum width).

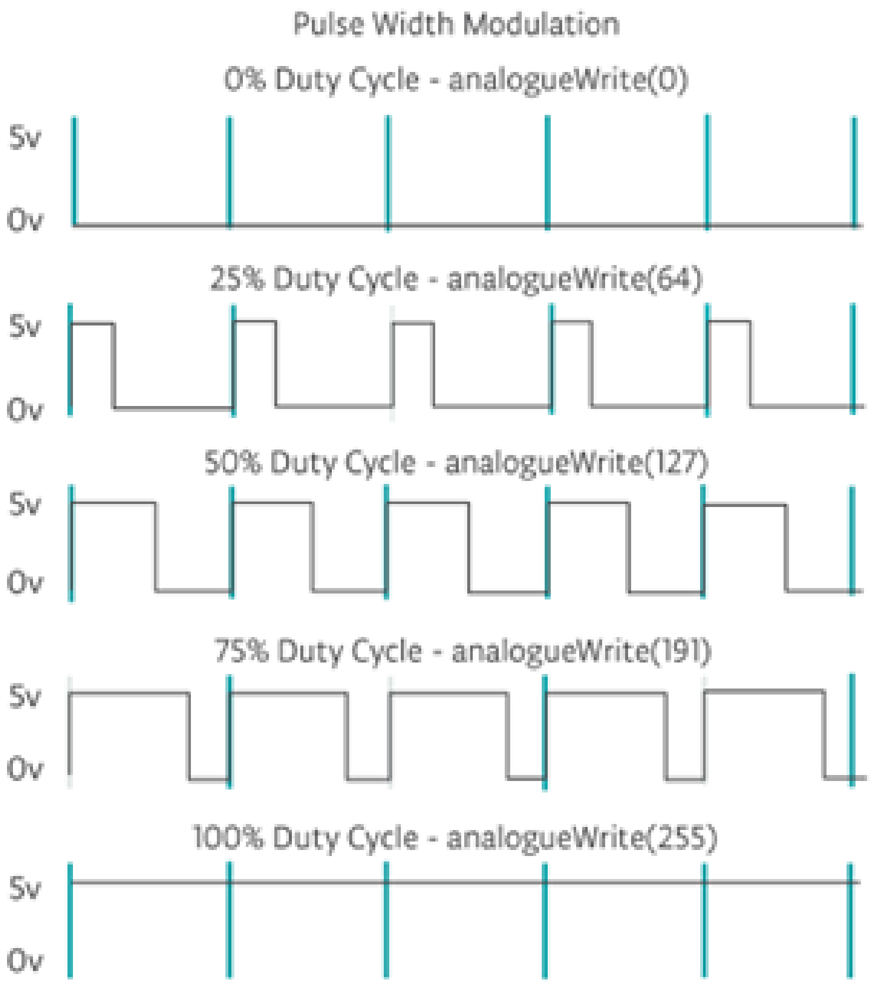

Figure: PWM (Source: arduino.cc)

6. Encoder

Magnetic encoders are sensors that can report information about the rotation speed and the spinning direction of the motor when mounted on a motor. The magnetic encoders are composed of a module with two Hall-effect sensors and magnetic discs. As the motor turns, the disc rotates past the sensors. Each time a magnetic pole passes a sensor, the encoder outputs a digital pulse, also called a “tick”. By counting those ticks, the speed of the motor can be determined.

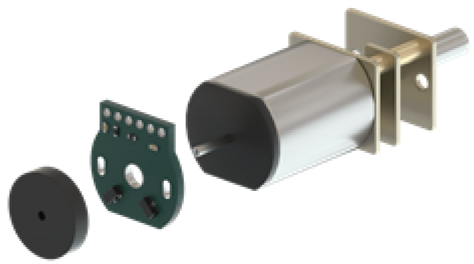

Figure: Encoder (Source: arduino.cc)

The encoder has two outputs, one for each Hall effect sensor. The sensors are positioned so that there is a phase of 90 degrees between them. This means that the square wave outputs of the two Hall effect sensors on one encoder are 90 degrees out of phase. This is called a quadrature output.

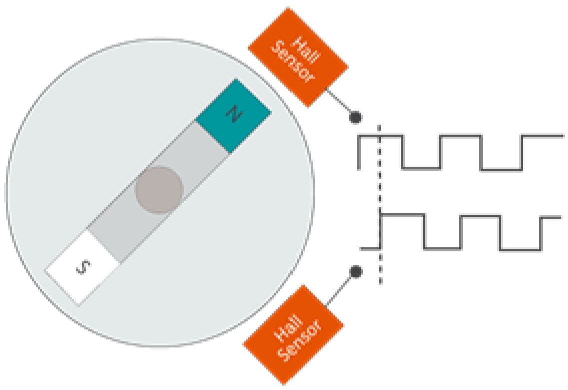

Figure: Encoder (Source: arduino.cc)

7. Potentiometer / Analog to Digital Converter (ADC)

A potentiometer (also known as a pot or potmeter) is defined as a 3 terminal variable resistor in which the resistance is manually varied to control the flow of electric current. A potentiometer acts as an adjustable voltage divider. Potentiometers work by varying the position of a sliding contact across a uniform resistance. In a potentiometer, the entire input voltage is applied across the whole length of the resistor, and the output voltage is the voltage drop between the fixed and sliding contact as shown below.

$$V_{out} = V_{in} * \frac{R_2}{R_1+R_2}$$

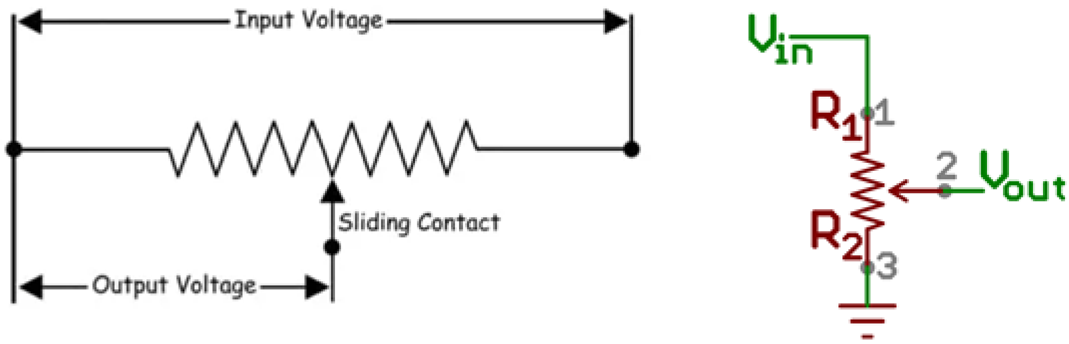

Figure: Potentiometer

An Analog to Digital Converter (ADC) converts an analog voltage on a pin to a digital number. One of the most common techniques uses the analog voltage to charge up an internal capacitor and then measure the time it takes to discharge across an internal resistor. The microcontroller monitors the number of clock cycles that pass before the capacitor is discharged. This number of cycles is the number that is returned once the ADC is complete.

8. I2C communication protocol

The Inter-Integrated Circuit (I2C) Protocol is a protocol intended to allow multiple "peripheral" digital integrated circuits ("chips") to communicate with one or more "controller" chips. It is only intended for short distance communications within a single device. Like Asynchronous Serial Interfaces (such as RS-232 or UARTs), it only requires two signal wires to exchange information. It can support multiple peripheral devices. Each I2C bus consists of two signals: SDA and SCL. SDA (Serial Data) is the data signal and SCL (Serial Clock) is the clock signal. The clock signal is always generated by the current bus controller. The I2C bus drivers are "open drain", meaning that they can pull the corresponding signal line low, but cannot drive it high. Thus, there can be no bus contention where one device is trying to drive the line high while another tries to pull it low.

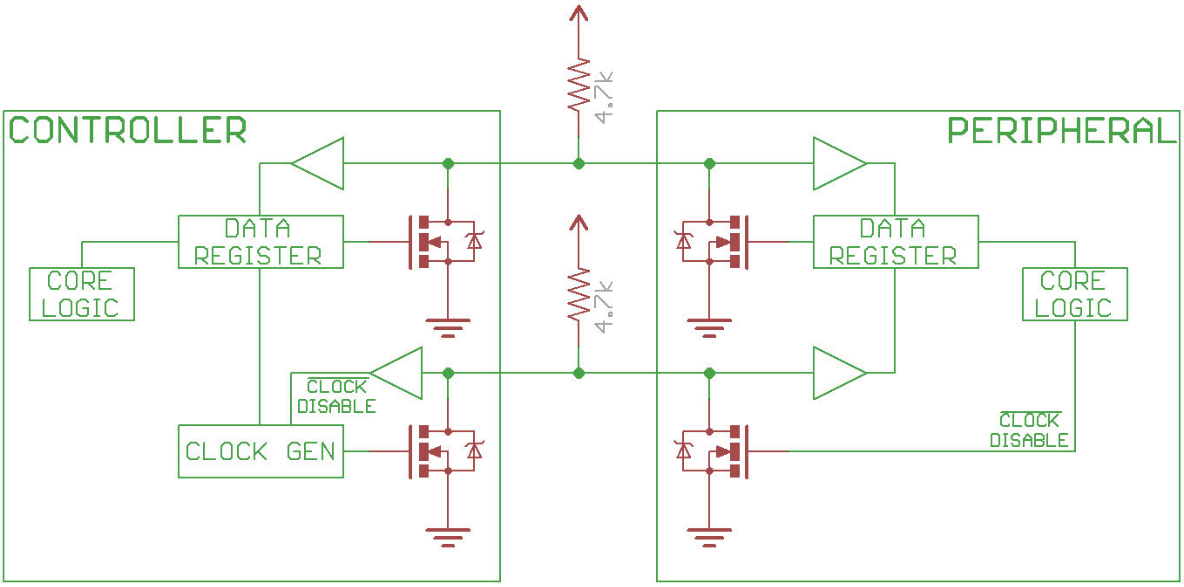

Figure: I2C Protocol

9. VL53L4CD - Time-of-Flight high accuracy proximity sensor

The VL53L4CD is a state of the art, Time-of-Flight (ToF), laser-ranging sensor enhancing the ST FlightSense product family. It integrates a SPAD array, physical infrared filters, and an integrated firmware to achieve the best ranging performance in various ambient lighting conditions with a range of cover glass materials.

ToF is the measurement of the time taken by an object, particle, or wave to travel a distance. ToF sensors use a tiny laser to fire out infrared light where the light produced will bounce off any object and return to the sensor. Based on the time difference between the emission of the light and its return to the sensor after being reflected by an object, the sensor is able to measure the distance between the object and the sensor.

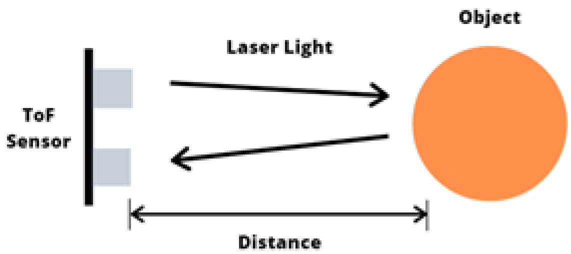

Figure: Lidar - ToF

10. Credits / Additional resources:

https://www.arrow.com/en/research-and-events/articles/engineering-basics-what-is-a-microcontroller

https://www.arduino.cc/

https://www.arduino.cc/reference/en/

https://www.arduino.cc/en/Guide/Environment

https://aek.arduino.cc/

https://www.electrical4u.com/potentiometer/

https://learn.sparkfun.com/tutorials/voltage-dividers/all

https://learn.sparkfun.com/tutorials/analog-to-digital-conversion

https://learn.sparkfun.com/tutorials/i2c/all

https://www.st.com/en/imaging-and-photonics-solutions/vl53l4cd.html

https://www.seeedstudio.com/blog/2020/01/08/what-is-a-time-of-flight-sensor-and-how-does-a-tof-sensor-work/

https://docs.arduino.cc/learn/starting-guide/whats-arduino

https://www.electronicsforu.com/news/new-products/programmable-led-driver-for-automotive-applications