

IoT based Real-Time Water Quality Monitoring System using smart Sensors

Abstract – This paper describes the design and development of a water quality monitoring system based on Internet of Things (IoT) using smart sensor networks. The objective is to notify the user in real time about water quality parameters. According to the World Health Organization (WHO), around 76 million people in our country do not have access to safe drinking water, and 21% of diseases are water-related. To reduce water-related diseases, we need to measure water parameters such as pH, conductivity, turbidity, temperature, etc. These parameters are used to detect water contaminants. The sensors are designed and connected to a microcontroller which has an inbuilt ADC circuit to convert the signal, also processes and analyzes the data. In this design, the Wi-Fi protocol is used to send the data to the cloud. The projected water quality observation interface shows the sensor data with a quality observation in an IoT setting. WQM selects parameters of water like temperature, pH level, turbidity by multiple different device nodes. This methodology sends the information to the web server. The updated data in the server may be retrieved or accessed from any place in the world in real time. Also, the system predicts the status of water based on a machine learning algorithm. Classification using a decision tree algorithm is used to predict the classes of water.

Key Words: pH sensor, turbidity sensor, temperature sensor, Wi-Fi protocol, internet of things (IoT), machine learning, Decision tree algorithm

1. INTRODUCTION

In the 21st century, providing pure drinking water is becoming a major challenge worldwide. International governing bodies such as the United Nations (UN) and the World Health Organization (WHO) also recognize human right to sufficient, continuous, safe, and acceptable, physically accessible, and affordable water for personal and domestic use. According to research by WHO, 844 million people lack even a basic drinking water service; impure drinking can cause life-threatening diseases such as diarrhea, cholera, dysentery, typhoid, and polio. The research alarmingly estimates that every year, diarrhea alone is causing around 5 lakh deaths. Hygienic water is one of the most important resources required to sustain life, and the quality of drinking water plays a very important role in the well-being and health of human beings [1]. Water supply to taps at urban homes and water sources available in more rural areas, is however, not necessarily safe for consumption. It is

thus paramount to monitor the quality of water which will be used for consumption. The monitoring is defined as the collection of information at set locations and at regular intervals in order to provide data which may be used to define current conditions. Traditional water quality monitoring methods involve sampling and laboratory techniques. These methods are however time-consuming (leading to delayed detection of and response to contaminants) and not very cost-effective. There is thus a need for more extensive and efficient monitoring methods.

Quality monitoring can be achieved through physicochemical measurements as well as microbial measurements. Physicochemical parameters include electrical conductivity, pH, oxidation-reduction potential (ORP), turbidity, temperature, chlorine content, and flow. These parameters can be analyzed quickly and at less cost than the microbial parameters. Monitoring with sensor technology [11] is still not very effective, as they do not always meet the practical needs of specific utilities; although cheaper than traditional equipment, cost, reliability, and maintenance issues still exist; and data handling and management can also be improved. In this paper, the development of a low-cost, wireless, multi-sensor network for measuring the physicochemical water parameters; enabling real-time monitoring, is presented. The system implements temperature, turbidity, and pH sensors from first principles. All the data from the sensors are processed and analyzed, and transmitted to the cloud and stored data in the cloud is downloaded and analyzed using classification-based decision tree algorithm.

2. LITERATURE REVIEW

The literature review involving the implementation of water quality monitoring systems using wireless sensor network (WSN) technology can be found in literature. In [8], a distributed system for measuring water quality is designed and implemented. Temperature, conductivity, pH, and turbidity sensors are connected to a field point, wherefrom and transferring the measured value via GSM technology to the work station. In [11], [8] a Zigbee-based WSN water quality monitoring and measurement system is presented. The system enables remote probing and real-time monitoring of the water quality parameters and also enables observation of current and historical water quality status. Data is sent using a GSM (Global System for Mobile Communications) network to a land-based station. The focus of this study is

however on the processing of the sensor data using Kohonen maps (auto-associative neural networks). In [9] WSN-based water environment system which consist of three part :Data monitoring nodes, video base station and remote monitoring center of key areas and water parameters such as temperature, turbidity, pH, dissolved oxygen and conductivity is presented the measured parameter processed by ARM-DSP and for communication purpose CDMA and Zigbee technology is used. Data is sent from the data monitoring nodes and data video base station to a remote monitoring center using Zigbee and CDMA (code division multiple access) technology. The water environment monitoring system implemented in [10] analyses and processes water quality parameters (pH, conductivity, dissolved oxygen and temperature), and also sounds an alarm when there is a water contamination is above the standard value mentioned from the WHO, or change in water quality. The whole water environment monitoring system presents useful characteristics as large network capacity. The parameters are measured with off-the shelf sensors and data is sent to a base station via GPRS (general packet radio service).

A river basin scale WSN for agriculture and water monitoring, called Soil Weather is implemented in [12] in this high quality sensors are used for water and soil to calibrate the sensor to monitor data quality and automatic cleaning sensors in water by compressed air. The network uses GSM and GPRS technology for transmission of sensor data. A turbidity system is proposed in [13], operation is based on the principle that the intensity of the light scattered by the suspended matter is to its concentration.. Turbidity can be interpreted as a measure of the relative clarity of water and often indicates the presence of dispersed, suspended solids; particles not in true solution such as silt, clay, algae based on passing the UV rays. Which is low powered [14], small-sized, easy-to-use, light weight and inexpensive. In [15], the DEPLOY project is introduced to monitor the spatial and temporal distribution of water quality and environmental parameters of a river catchment. It is intended to demonstrate that an autonomous network of sensors can be deployed over a wide area and the system measures parameters such as pH, temperature, depth, conductivity, and turbidity and dissolved oxygen. A microcontroller-based WSN system is proposed in [16] to measure pH, chlorine concentration and temperature in a pool. Data is transmitted using GSM and in sleep mode the sensor nodes are shown to consume 27 μ A. Besides, the input voltage is 3.0V, so TB88-30 is idle and no power loss. When the input voltage descends, the current consumption is getting to increase. In [17] a WSN system is used to measure the water quality of fresh water and uses solar daylight harvesting for optimized power management. The data collected from the various sensor nodes are sent to a sub-base node and from there to a monitoring station using a GSM network. The system powered by PIC16F886, RF ZIGBEE 802.15.4 ISM 2.4G Hz module for each node, integrated with GSM/GPRS modem, to

solve power issue Li-PO battery is used. A low-cost, real-time, in-pipe sensor node with a sensor array for measuring flow, pH, conductivity, ORP and turbidity, is designed and developed in [18]. Contamination event detection algorithms are also developed to enable sensor nodes to make decisions and trigger alarms when contaminants are detected.

3. METHODOLOGY AND IMPLIMENTATION

OF SYSTEM

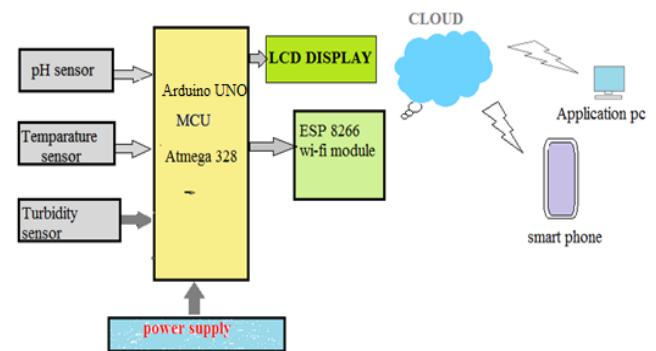


Figure 1: Block diagram of water quality monitoring system

This paper presents the theory on real time monitoring of water quality in IoT environment. The overall block diagram of the proposed method is explained. Each and every block of the system is explained in detail.

In this proposed block diagram consist of several sensors (temperature, pH, turbidity) is connected to core controller. The core controller are accessing the sensor values and processing them to transfer the data through internet. Arduino is used as a core controller. The sensor data can be viewed on the internet Wi-Fi system. The pure water should be within the specified WHO standard range shown in below table.

Table1: water quality physical parameter standard value table

Sl.No	Parameter	Range	Unit
1	Temperature	15-30	°C
2	pH	6.5-8.5	pH
3	Turbidity	0-5	NTU

a) pH sensor: The pH of a solution is the measure of the acidity or alkalinity of that solution. The pH scale is a logarithmic scale whose range is from 0-14 with a neutral point being 7. Values above 7 indicate a basic or alkaline solution and values below 7 would indicate an acidic

solution. It operates on 3.3V power supply and it is easy to interface with arduino. The normal range of pH is 6 to 8.5

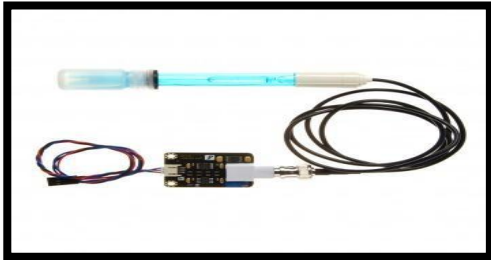


Figure 2: pH sensor

b) Turbidity sensor: Turbidity is a measure of the cloudiness of water. Turbidity has indicated the degree at which the water loses its transparency. It is considered as a good measure of the quality of water. Turbidity blocks out the light needed by submerged aquatic vegetation. It also can raise surface water temperatures above normal because suspended particles near the surface facilitate the absorption of heat from sunlight.



Figure 3: turbidity sensor

c) Temperature sensor: Water Temperature indicates how water is hot or cold. The range of DS18B20 temperature sensor is -55 to +125 °C. This sensor is digital type which gives accurate reading.



Figure 4: Temperature sensor

Arduino Uno: Arduino is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards

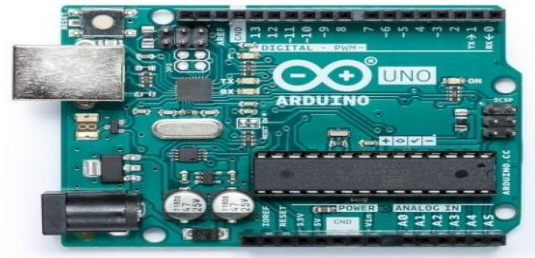


Figure 5: arduino uno platform

Wi-Fi module: The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware. The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.



Figure 6: ESP8266 Wi-Fi module

Implementation: The whole design of the system is based mainly on IOT which is newly introduced concept in the world of development. There is basically two parts included, the first one is hardware & second one is software. The hardware part has sensors which help to measure the real time values, another one is arduino atmega328 converts the analog values to digital one, & LCD shows the displays output from sensors, Wi-Fi module gives the connection between hardware and software. In software we developed a program based on embedded c language. Classification based decision tree algorithm is used to predict the water quality classes using matlab.

Once the all hardware is interfaced to the core we need to program for sensor and communication devices. power the module to measure water parameters from sensor, each sensor value is measured at different interval time and displayed on LCD alternatively accumulate the measured value and send to cloud using ESP8266 wi-fi module. Here Thingspeak is used as a cloud server. User can view the real time measured value by login the account, where we can monitor the real time water parameters in graphical view.

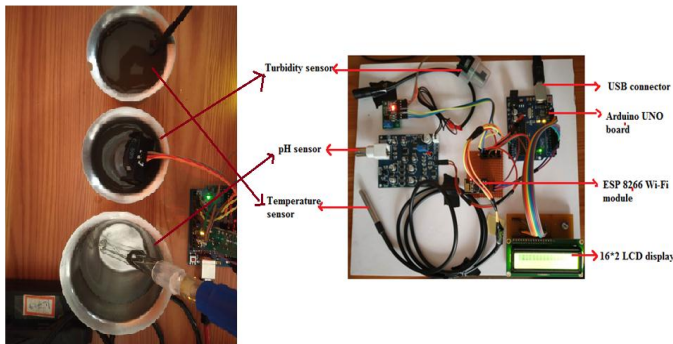


Figure7: System design model of water quality monitoring system using IOT

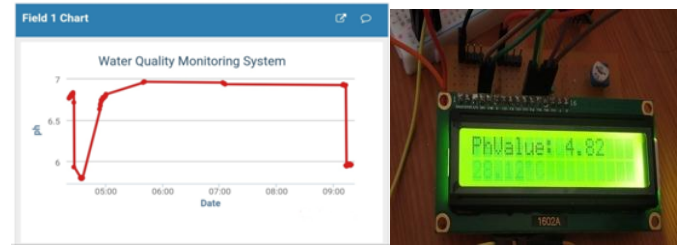


Figure 9: Measurement of pH value

The voltage output from the pH electrode which ranges from -430 mV to 430 mV is represented by measured voltage. The resolution is 60mV/pH. Figure 9 shows the different water sample results but in LCD display 4.82 is the lemon pH.

Similarly the temperature is also a one of parameter of water quality .which is to be sensed by temperature sensor and obtained the live temperature data on thingpeak window in the form of graph shown in below figure. Where X axis represents the time slot and Y axis represents the temperature value of water sample. The expected output from the op-amp U1 should read from 0 to 2.85V as specified in the design. The resultant output displayed in integer on LCD.

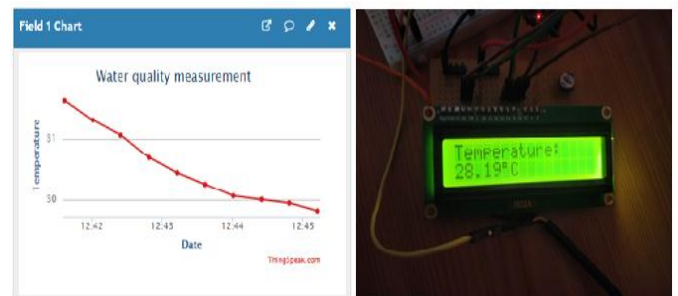


Figure 10 : Measurement of water temperature using temperature sensor

Likewise the turbidity is also a one of parameter of water quality .which is to be sensed by turbidity sensor and obtained the live result data on thingspeak window in the form of graph shown in below figure. Where X axis represents the time slot and Y axis represents the turbidity in NTU and which is in the range of 0-3000 NTU of water sample. Turbidity causes cloudiness or decrease in transparency of water the direction of transmitted light path will undergo changes when the light hits the particles in the water column and microcontroller gets the digital signal from inbuilt ADC circuit and display the results along with all above mentioned parameters shown in figure 11.



Figure 8.1 figure: 8.2 figure 8.3

The figure 9 represents the pH value of the water, the X axis is represents the time and Y axis represents the pH value. We took three different samples for demo purpose. One is pure water, second one is drainage water and third one is added acidic water each samples parameters are measured continuously and obtained result is real time on both lcd display and on web. In figure 9 the pH value is range from in between the 7 and 5.5 of three different samples.

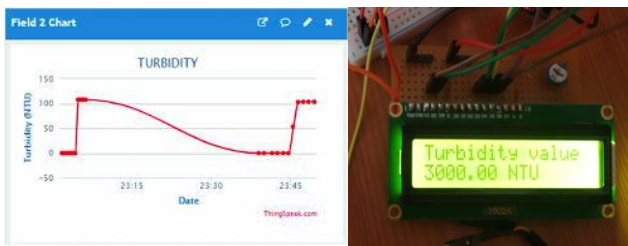


Figure 11: Measurement of water turbidity using turbidity sensor

Once the sensed data from sensor by all three sensor is sent to cloud, that data sample means last obtained data is downloaded using MATLAB code and analyzes or predict quality of water. Here we used decision tree classification algorithm to predict the water parameter. The decision tree algorithm is preferred and GUI window is created to show the predicted result on MATLAB.

The GUI is created as the measured data can be in used to predict in either on IoT or manual mode. In manual mode we entered the data manually where we get from on either LCD window which is connected to circuit or where we get from cloud. Based on those data algorithm is predicts water pH value is either acidic or alkaline, and temperature is COOL, HOT, and NORMAL, and Turbidity level is BAD, MODERATE, POOR. Shown in figure12.

The IoT mode operates based on live stream data is downloaded automatically from the cloud and predicts the result and measure the accuracy of the water sample shown in following figure. In IoT mode operation, initially we train the model based on data sets and then read the data from cloud where last measured value is fetched from cloud as Excel file. That data is used to predict the water quality using trained data set and measures the acidity level, Turbidity level and Temperature of water. The following tree is formed based on decision tree algorithm. The figure 12 shows the proposed model decision tree diagram.

Here we considered three different parent nodes each node having three child nodes. The child node is formed based on the information Gain. Which decides feature to split on at each step in building the tree? The split with the highest information gain will be taken as the first split and the process will be continue until all children nodes are pure, or until the information gain is 0.

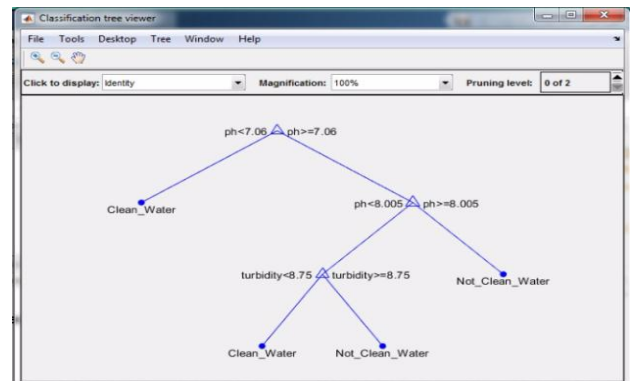


Figure 12: Decision tree of proposed model

The Graphic user interface (GUI) window is prepared in MATLAB for viewing the water parameters based on run the model. Which is also known as Apps, provide point and click control of our software application.

Its an interactive environment that integrates the two primary tasks of App building: laying out the visual components and programming the Apps behavior. It allows us to quickly move between visual design in the canvas and developing code in MATLAB editor.

The following GUI window having two blocks of operation. first one is IoT mode and Manual mode, initially we train the model just click on the "train the model" button then read the data from cloud after that click on "predict the water quality" then following empty block will show a quality of water

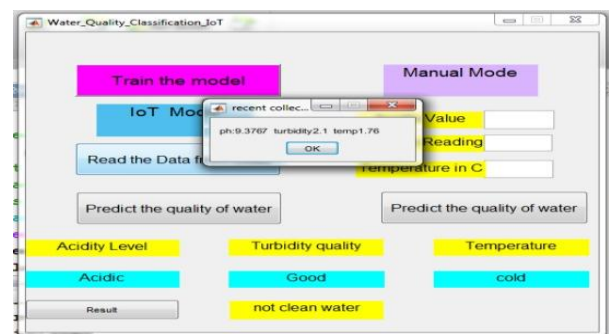


Figure 13: GUI window of water quality

5. CONCLUSION AND FUTURE SCOPE

Based on study of existing water quality monitoring system and scenario of water we can define that proposed system is more suitable to monitor and analysis of water quality parameters in real time. The smart water quality meter is automatic and does not require human interface, thereby reducing the errors, a sensor node with a temperature, pH, and turbidity sensors was designed and constructed on a Vero-board. The proposed system having wireless sensor networking using Wi-Fi module, which makes sensor

BREAD BOARD TRAINER



DTK-06 is a single board Breadboard Trainer Kit which is widely used to train engineers to develop/ Study hardware for any Digital/Analog integrated circuit in laboratory. This board having logic inputs, output indicator LED's, Push Switch Key, Sine/Square wave Generator, Potentiometer, Panel Meter, General purpose Breadboard, AC/DC Supplies for experimenting the digital/analog techniques. In this board all interfacing between onboard source & breadboard are done by patch chords.

Specifications

● Indicators

- 08 TTL/CMOS Logic Level Inputs with Dual Color LED indication for logic low and logic high.
- 10 Dual color LED's indicator for output for indicating low, High and tristate.

● Push switch key

- On board 2 nos. of push switch tact keys provided output on 2mm banana connector.

● Potentiometers

- Variable potentiometers of 1K Ω , 10K Ω , 100K Ω & 1M Ω provided on board

● Bread Board Area

- Eight Distribution Strip of 100 tie points each totaling 800 tie points
- Four Terminal Strip of 630 tie points totaling 2520 tie points

● Function Generator

- Provides Sine, Square and Triangular output waveforms.
- Frequency variable from 1Hz to 100 KHz in FIVE steps.

● On board discrete component

- Selectable 3W Resistor Bank ranging from 1 Ω to 10k Ω (11 Nos.)
- Selectable Electrolytic Capacitor Bank ranging from 1 μ F to 1000 μ F (11 Nos.)
- Selectable Ceramic Capacitor Bank ranging from 10pF to 470KpF (11 Nos.)
- One 8ohm Speaker output provided on 2mm banana sockets

● Panel Meter

- Analog DC Ammeter of 0 to 20 mA.

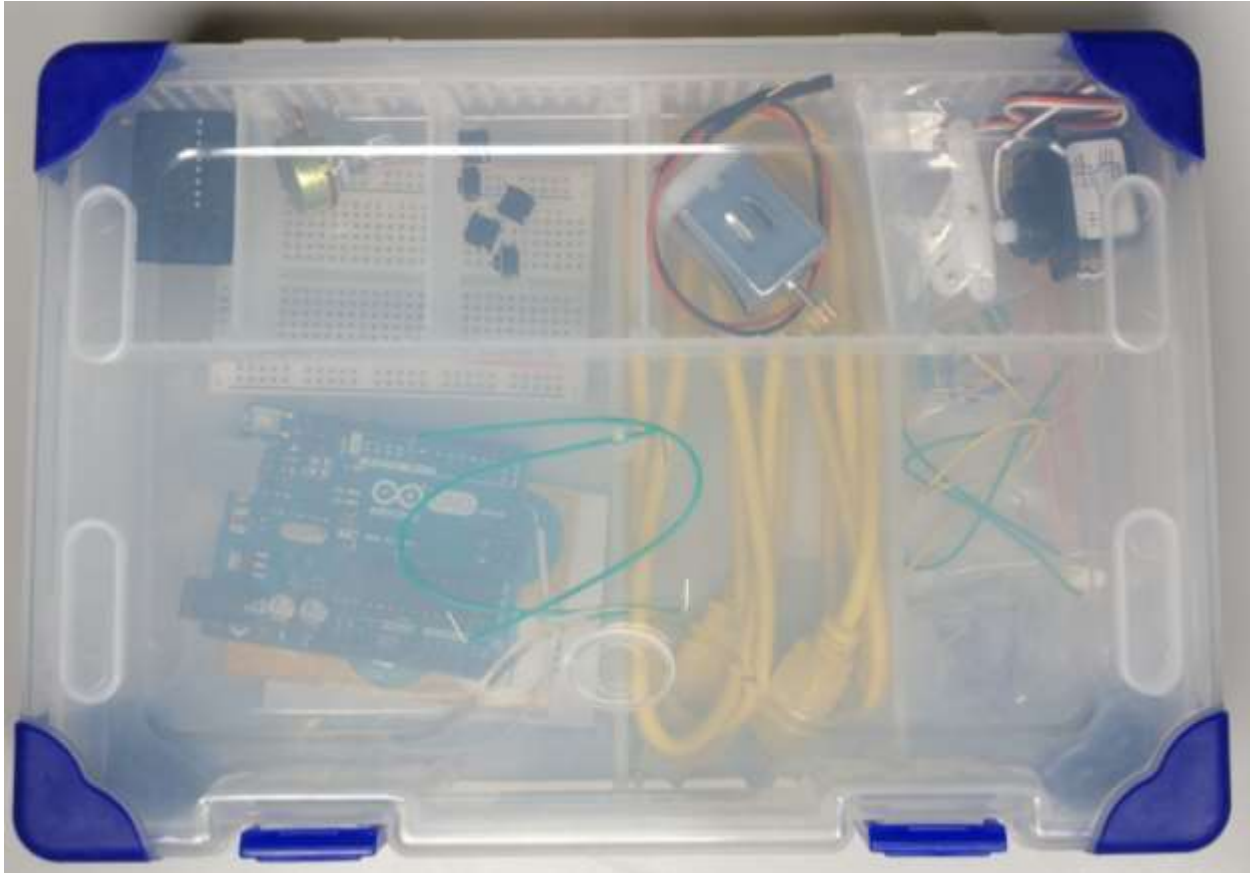
● Power Supplies

- Selectable Fixed DC Power supply of +5V, +6V, +9V, +12V, +15V.
- AC Power Supply of 15V-0-15V.

● Interconnections

- All interconnections are made using 2mm banana Patch cords.
- All ICS are mounted on IC Sockets.
- Bare board Tested Glass Epoxy SMOBC PCB is used.
- Attractive Metal/ Wooden enclosures.
- Set of 2mm Patch cords for interconnections.
- User's Manual.

Thank you for borrowing Toronto Public Library's Arduino Kit. Please return this kit to the Digital Innovation Hub from which it was borrowed.



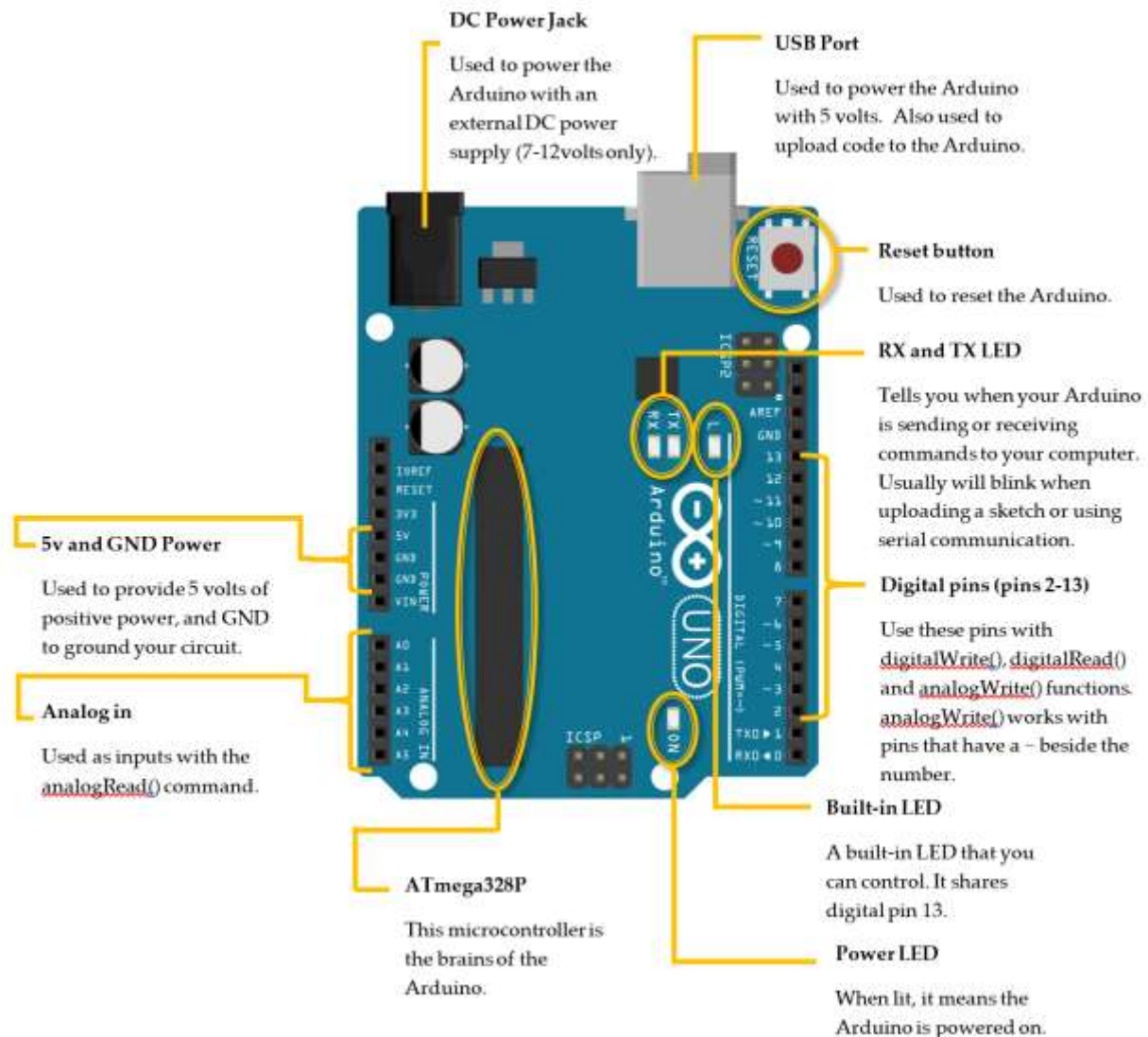
Borrowing Arduino Kits

- Arduino Kits are available to Toronto Public Library customers with a valid Teen (13-17), Adult – Under 25 (18 – 24), or Adult (25+) library card.
- Holds cannot be placed on the Arduino Kits.
- You can only borrow one Arduino Kit at a time. Each kit can be borrowed for 21 days and cannot be renewed.

Fines Per Day and Maximum Fines for Arduino Kits		
CARD TYPE	FINE AMOUNT PER DAY	MAXIMUM YOU WILL BE CHARGED FOR EACH LOAN PERIOD
Adult	\$0.35	\$14.00
Adult Under 25 (18-24)	\$0.15	\$6.00
Teen (13-17)	\$0.15	\$6.00

- If you lose an Arduino Kit, you will be charged the purchase price of the Arduino (\$50). The library does not accept a replacement Arduino or an item of equal value.
- If the Arduino Kit is overdue by more than 40 days, the library considers it lost. If you find the kit within 6 months of paying the replacement cost you can get a refund, minus any overdue fines so please keep your receipt.
- Please report damaged equipment or missing parts to the Digital Innovation Hub staff from which it was borrowed. Damaged Arduino boards and kits are subject to replacement purchase fees.

Arduino Uno Tour



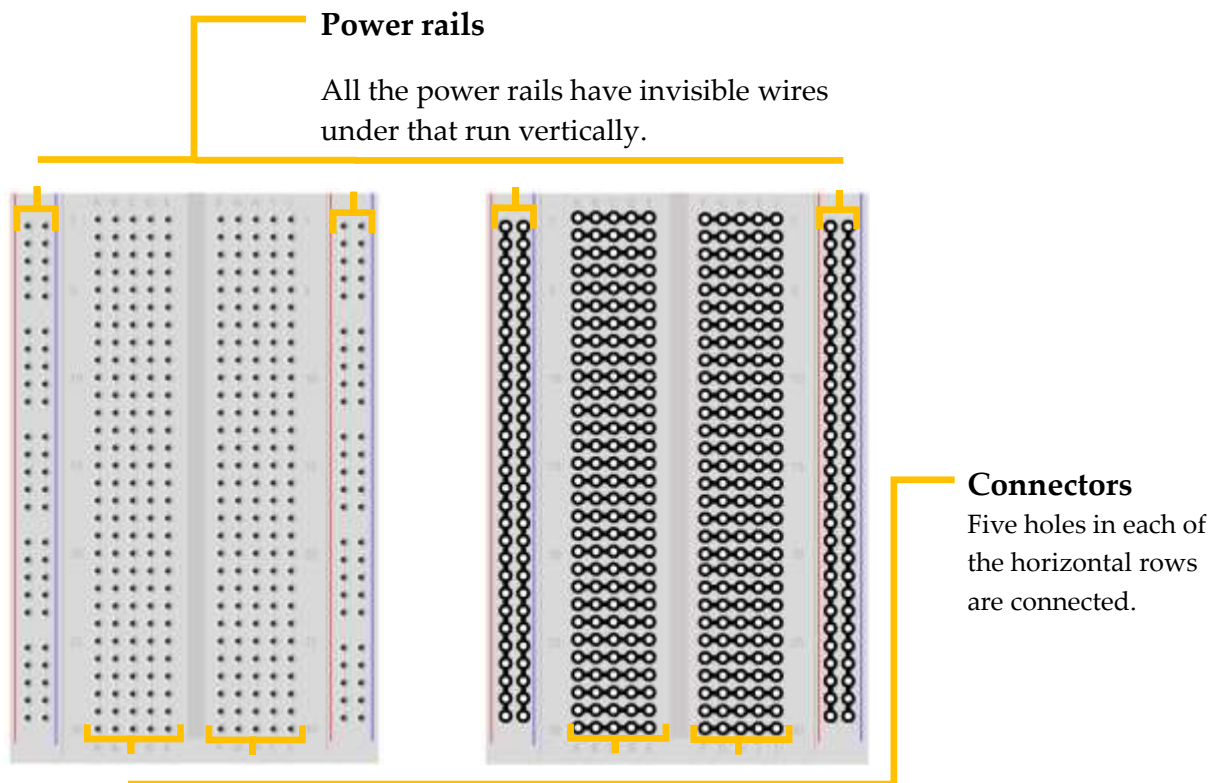
Spend a few moments looking at the diagram below and compare it to the Arduino included in your kit.

The Arduino has been labeled to help you learn all the different connectors and parts.

Getting to Know your Breadboard

In order for us to connect our tiny components together, we need our breadboard. A breadboard is great for prototyping since it does not create a permanent connection between components like soldering does. Everything is held together by friction when you insert them into those tiny holes inside your breadboard.

Remember: If you have any questions, or need some extra help, feel free to visit a Digital Innovation Hub at the Toronto Public Library for classes or assistance.



What a breadboard looks like if we could see the wires under the breadboard

Introduction

An Arduino is a microcontroller; a small, simple computer. It is designed specifically for beginners who are new to coding and electronics. You can learn more about the Arduino at <https://www.arduino.cc/en/Guide/Introduction>.

There are thousands of projects you can build with an Arduino.

Parts in This Kit

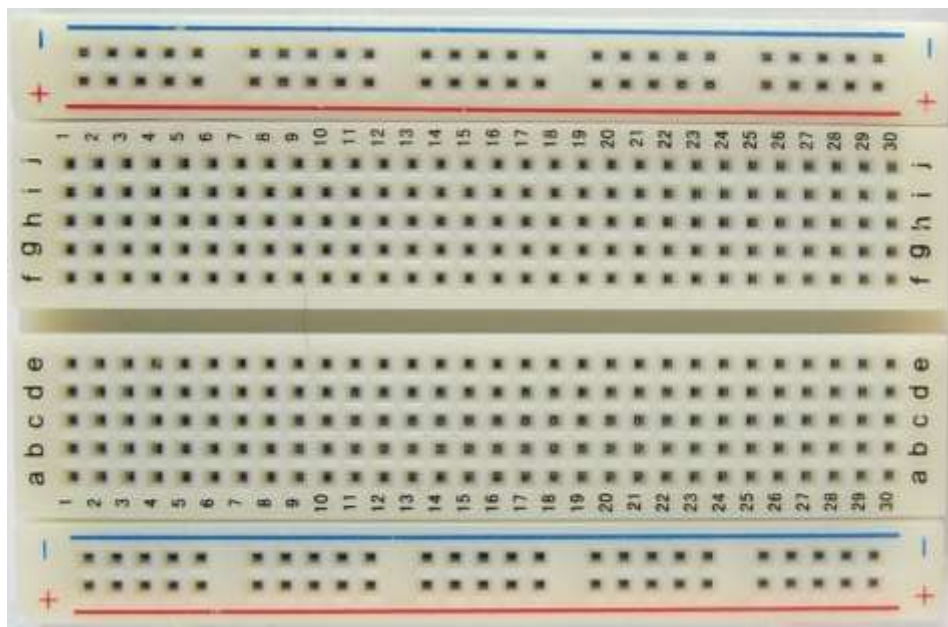
Your assembled kit includes all the parts you'll need for the Arduino projects outlined in this manual. When you're done with your projects, please return the parts to their proper slots, as indicated in this diagram, for the next person to enjoy.



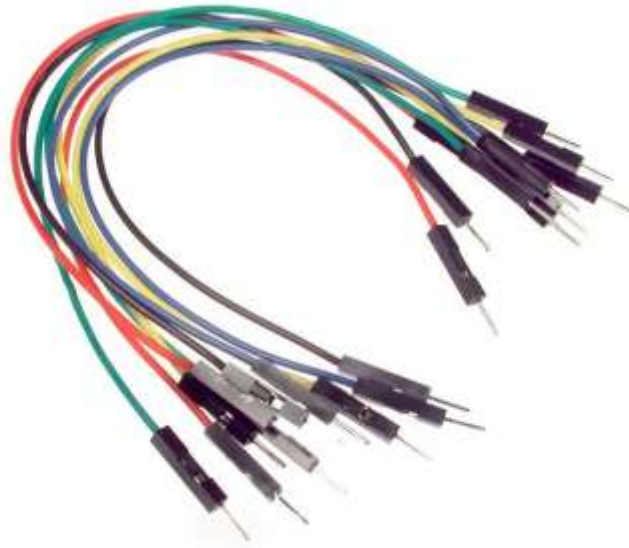
There are different types of Arduinos. This kit uses a blue **Arduino Uno** board. The different parts on the Arduino are labelled in white.



The **USB cable** is used to connect the Arduino to your computer.



The **breadboard** lets you build circuits. It has a series of holes where you can insert wires to create circuits. The magic of a breadboard is that it's reusable, and you don't need to solder (permanently joining components together to form a circuit by melting metals).



Jumper wires are used to create electric circuits and can be inserted into the breadboard.



A **H-Bridge IC Chip (model L293NE)** is an electronic integrated circuit chip that allows a voltage to be applied across a load (like a motor) in opposite directions. These are used in robotics so that you can control the direction of two motors independently.



DC Motor uses electricity to convert it to rotational mechanical energy. Connect the positive and negative to power and it will spin. Reverse the positive and negative pin to reverse the power flow and it will spin the opposite direction.

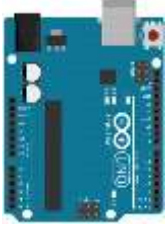
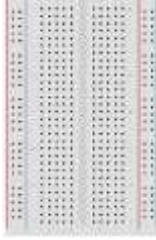








Potentiometers are a manually adjustable variable resistor with 3 terminals. Two terminals are connected to both ends of a resistive element, and the third middle terminal connects to a sliding contact, called a wiper, moving over the resistive element. The position of the wiper determines the output voltage of the potentiometer.



Servo Motor can be commanded to rotate to a specific angle. These motors cannot rotate continuously and only have a movement of 180 degrees.

Part Inventory for the Advanced Kit

 <p>1x Arduino</p> <p><i>The Arduino is the microcontroller and brains of our project. It stores programs and processes inputs and outputs.</i></p>	 <p>1x Breadboard</p> <p><i>The breadboard is used to temporarily connect multiple components and wires together during prototyping.</i></p>	 <p>16x Jumper Wires</p> <p><i>Jumper wires connect the components completing a circuit. Note: The colour of the wires do not matter when building the projects.</i></p>	 <p>1x Pushbutton</p> <p><i>When you push the button, it completes the circuit.</i></p>
 <p>1x H-Bridge IC chip</p> <p><i>Used to control the direction of two motors.</i></p>	 <p>1x Potentiometer</p> <p><i>Can be used to detect the position of a knob when connected to the Arduino analog input pins.</i></p>	 <p>1x DC Motor</p> <p><i>It is your standard DC motor. It spins when you apply power.</i></p>	 <p>1x Servo Motor</p> <p><i>Servo motor can be commanded to rotate to a specific angle (0-~180 deg).</i></p>

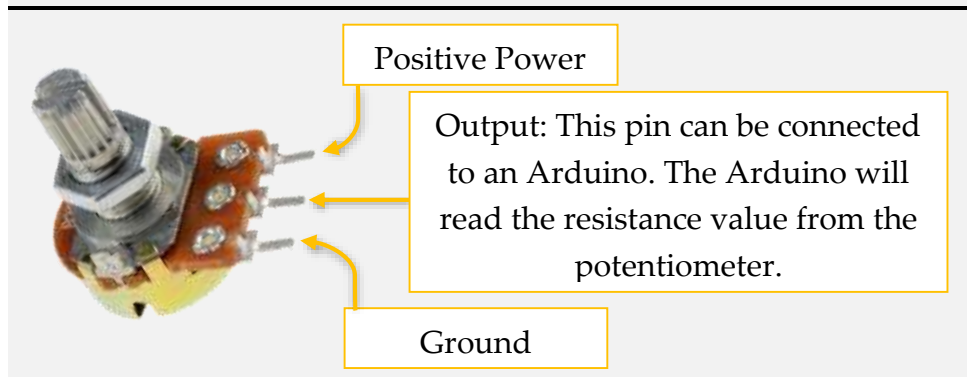
Controlling a Servo with a Potentiometer

In this project, we will be using a potentiometer to control the movement of our servo motor. When we twist our potentiometer knob, the servo will also rotate approximately the same amount. Remember, it doesn't matter what color the wires are in the diagram compared to the ones you use.



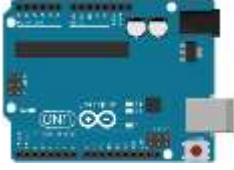




Potentiometers are like a dynamic resistor, as you twist, the resistance value changes (this will make the voltage change from 0-5v). These changes in resistance is what our Arduino will read and convert into a number between 0-1023.

What are the pins on the included Potentiometer?

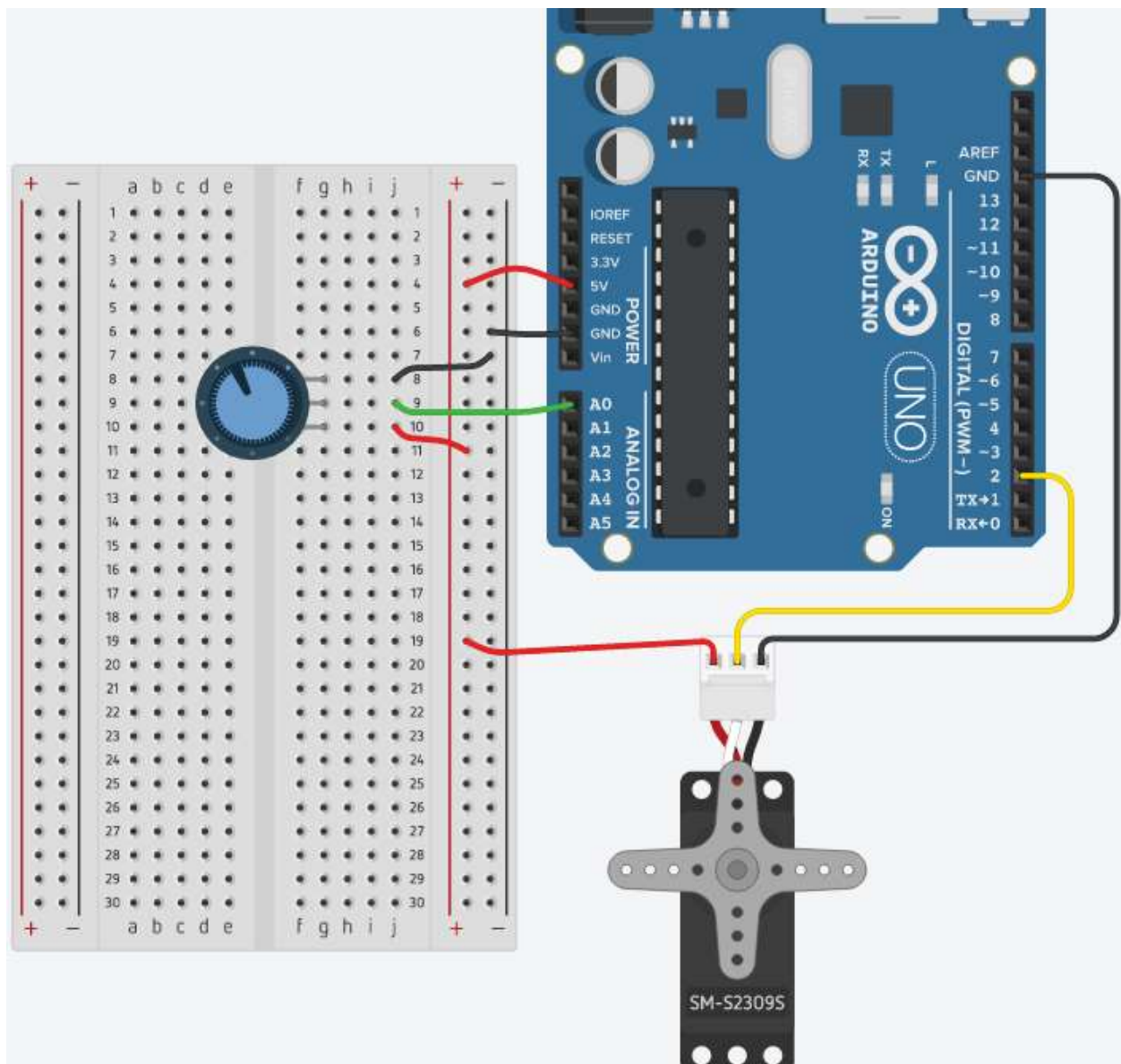


A servo motor, is a special type of motor that can be used to rotate precisely. There is a sensor inside the motor that keeps track of how much it rotates. This allows us to tell the motor to rotate by degrees and it will stop at the correct location.

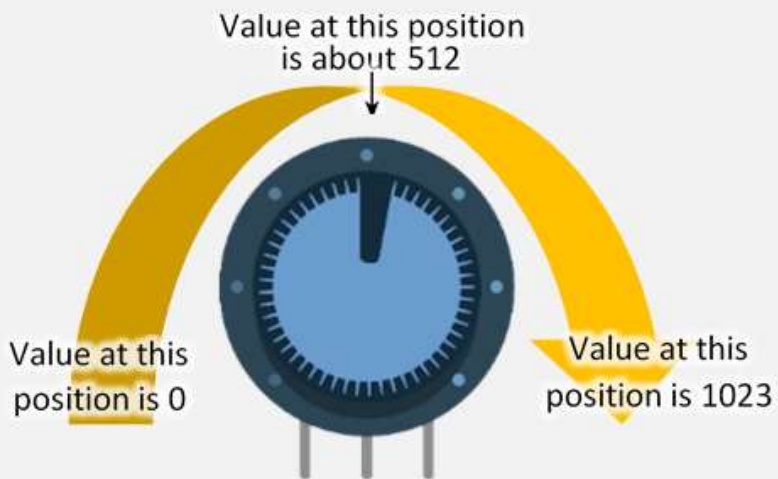
Required Components

				
1x Arduino	1x Breadboard	3x Jumper wires	1x potentiometer	1x Servo Motor

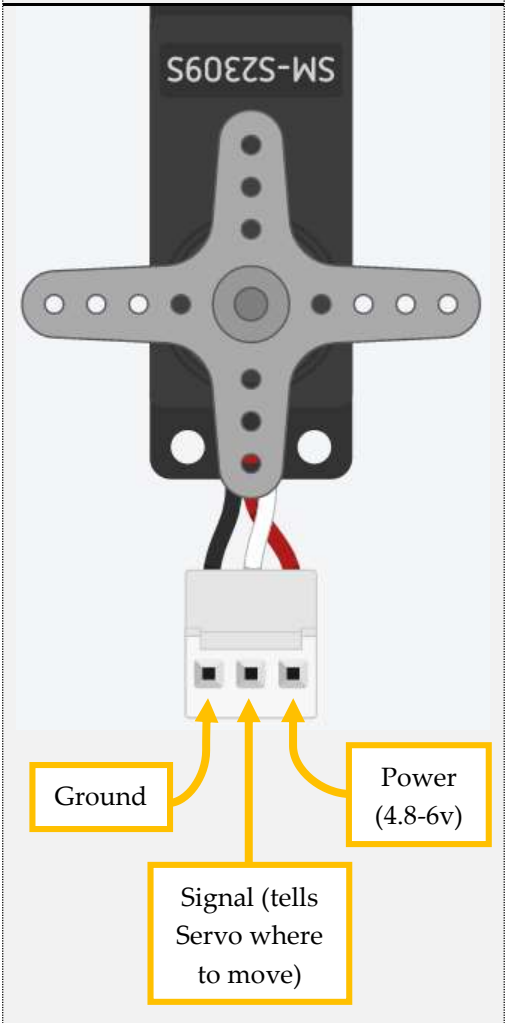
Wiring



Potentiometer Knob Input is from 0-1023



What are the pins on our Servo Motor?



Code

Line 1.
Line 2.
Line 3.
Line 4.
Line 5.
Line 6.
Line 7.
Line 8.
Line 9.
Line 10.
Line 11.
Line 12.
Line 13.
Line 14.
Line 15.
Line 16.
Line 17.
Line 18.
Line 19.
Line 20.
Line 21.
Line 22.
Line 23.
Line 24.
Line 25.
Line 26.

```
#include <Servo.h>
```

Include the ability to use the Servo motor library (enables us to use the servo commands).

```
const int servoPin = 2;
```

```
const int userInputPin = A0;
```

Create two constant integer variables. Variable *servoPin* will store the number 2, and *userInputPin* will store the number A0.

```
Servo myservo;
```

Create a new servo called *myservo*. We give it a name since we could control more than one servo at a time.

```
int pos = 0;
```

Create an integer variable with the value of 0. This will be used to store the position the servo should move to.

```
void setup() {
```

```
    myservo.attach(servoPin);
```

Tell the Arduino to use (attach) the servoPin (pin #2) to the servo called "*myservo*".

```
    pinMode(userInputPin, INPUT);
```

Set the potentiometer that is connected to userInputPin (pin #A0) as an INPUT.

```
}
```

```
void loop() {
```

Analog read the inputPin (A0), and store the value in our variable called *pos*.

```
    pos = analogRead(userInputPin);
```

```
    pos = map(pos, 0, 1023, 0, 180);
```

Map the raw sensor data (our sensor gives us a value between 0-1023) to a value between (0-255) and store the converted value back to the variable *pos*.

```
    myservo.write(pos);
```

Move the servo motor (called *myservo*) with the calculation we did previously stored in the variable *servoPos*.

```
    delay(100);
```

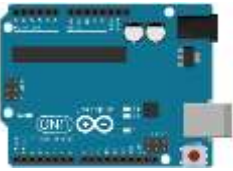
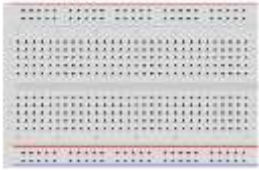



Delay (pause) for 100 microseconds. This will allow the servo motor to go to its position before being told to move again.

```
}
```

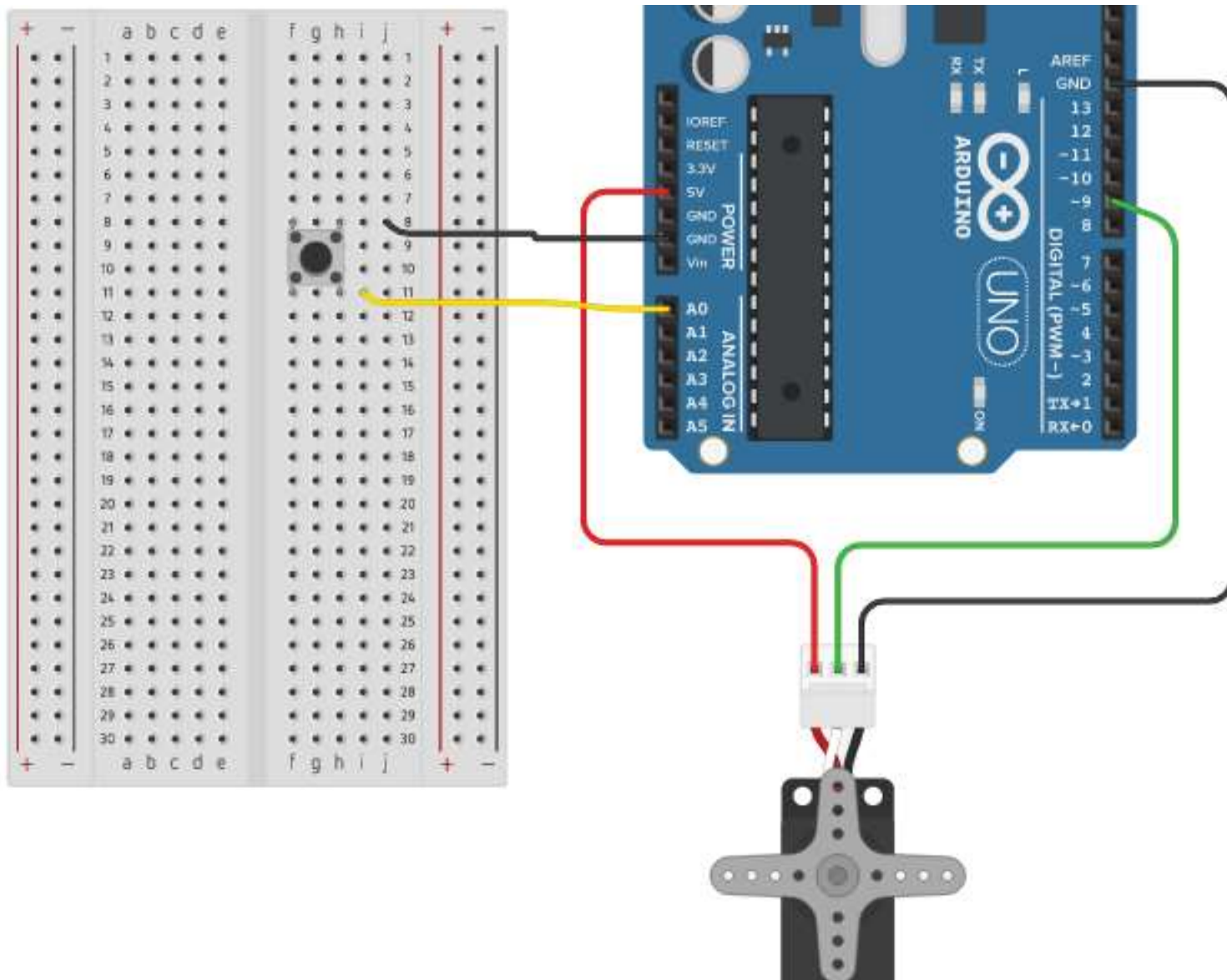

Creating a 30 Second Countdown Timer

In this project, we will be using a servo motor as a countdown timer. The Servo motor will rotate 3 degrees of movement for every second for a total of 30 seconds. We have a button that is connected to the Arduino so we can restart the 30 second timer when needed.

Required Components

				
1x Arduino	1x Breadboard	5x Jumper Wires	1x Pushbutton	1x Servo Motor

Wiring



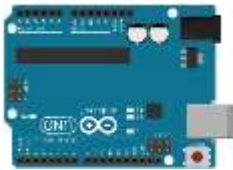
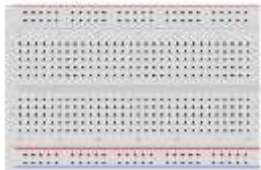



H-Bridge Controlled by an Arduino

In this project, we will be using an H-Bridge (Model L293D) connected with our Arduino to control the motor so it can rotate clockwise or counter clockwise.

Pin #9 from our Arduino must output HIGH (output 5v) to enable the h-bridge to work. Pins #11 and pin #10 is used to control which direction the motor will turn. If you have pin #11 high (output 5v) the motor will spin one direction, while if you output high on pin #9 it will spin the other direction. The motor will not turn if both pin #10 and pin #11 is high (outputting 5volts) since you are telling the h-bridge to go clockwise and counter clockwise at the same time.

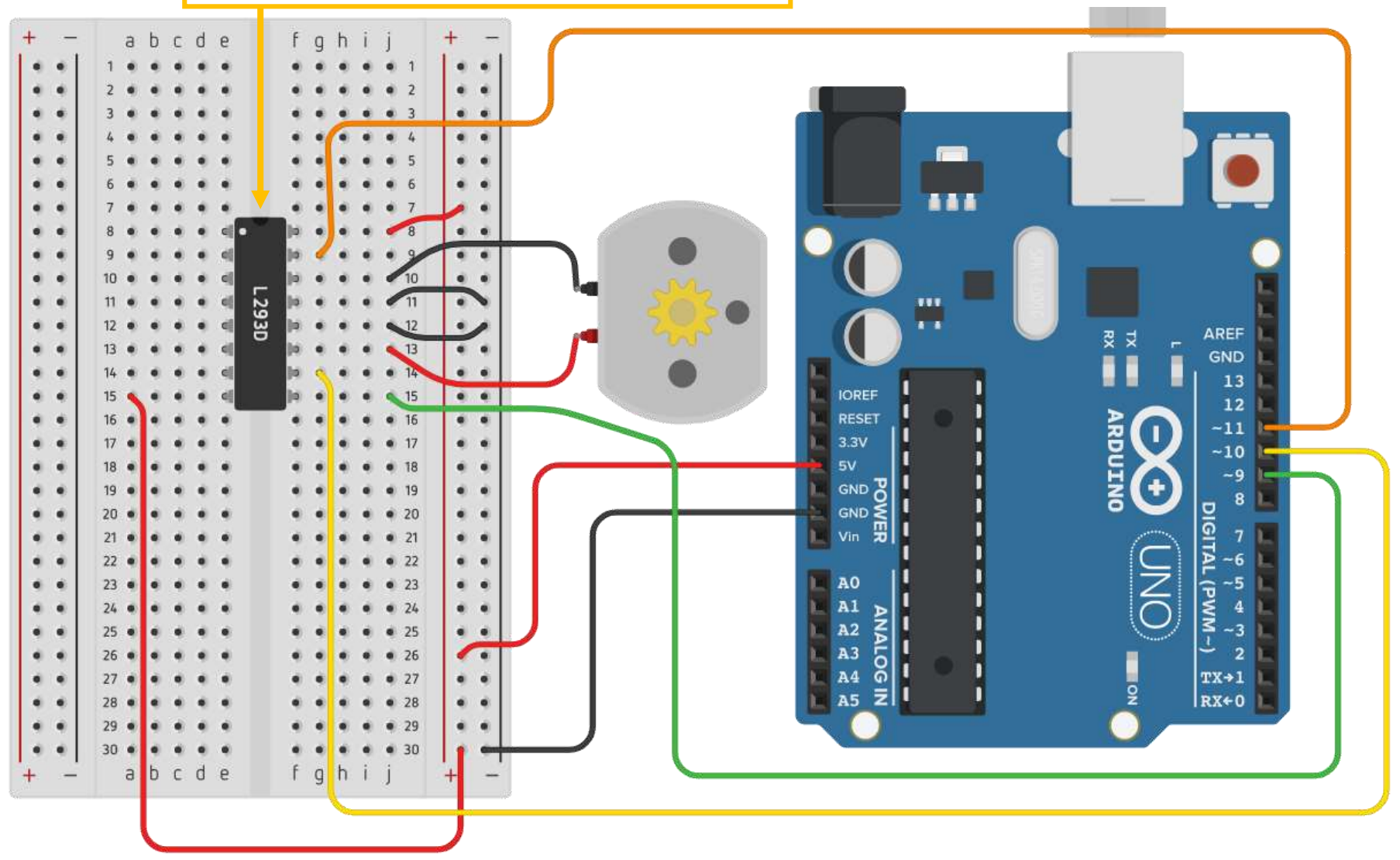
This project will rotate the motor clockwise for 6 seconds, stop spinning the motor for 2 seconds, rotate the motor counterclockwise for 6 seconds, stop the motor for 2 seconds and then repeat forever.

Required Components

				
1x Arduino	1x Breadboard	14x Jumper Wires	1x H-Bridge	1x DC Motor

Wiring

This indented semicircle is used to indicate the top of the IC H-bridge chip. Make sure you put in the h-bridge correctly.



Code

```

Line 1. void setup() {
Line 2.     pinMode(9, OUTPUT);
Line 3.     pinMode(10, OUTPUT);
Line 4.     pinMode(11, OUTPUT);
Line 5.
Line 6.     digitalWrite(9, HIGH);
Line 7. }
Line 8.
Line 9. void loop() {
Line 10.     digitalWrite(10, HIGH);
Line 11.     digitalWrite(11, LOW);
Line 12.     delay(6000);
Line 13.
Line 14.     digitalWrite(10, LOW);
Line 15.     digitalWrite(11, LOW);
Line 16.     delay(2000);
Line 17.
Line 18.     digitalWrite(10, LOW);
Line 19.     digitalWrite(11, HIGH);
Line 20.     delay(6000);
Line 21.
Line 22.     digitalWrite(10, LOW);
Line 23.     digitalWrite(11, LOW);
Line 24.     delay(2000);
Line 25.
Line 26. }

```

Set pin #9, pin #10 and pin #11 as an output

When the Arduino turns on or restart, set pin #9 as HIGH (on). This pin will be used to tell the h-bridge to be enabled. We did it in the setup since we never need to turn it off later

Set pin #10 to HIGH (on) and pin #11 to LOW (off) and then wait for 6 seconds. This will spin the motor one direction

Set pin #10 to LOW (off) and pin #11 to LOW (off) and then wait for 2 seconds. This will stop the motor from spinning.

Set pin #10 to LOW (off) and pin #11 to HIGH (on) and then wait for 6 seconds. This will spin the motor in the opposite direction.

Set pin #10 to LOW (off) and pin #11 to LOW (off) and then wait for 2 seconds. This will stop the motor from spinning.

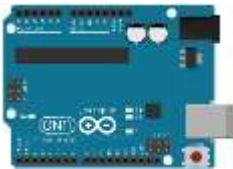
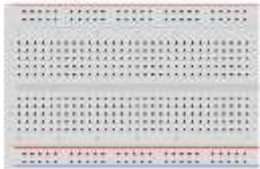




H-Bridge Using Potentiometer

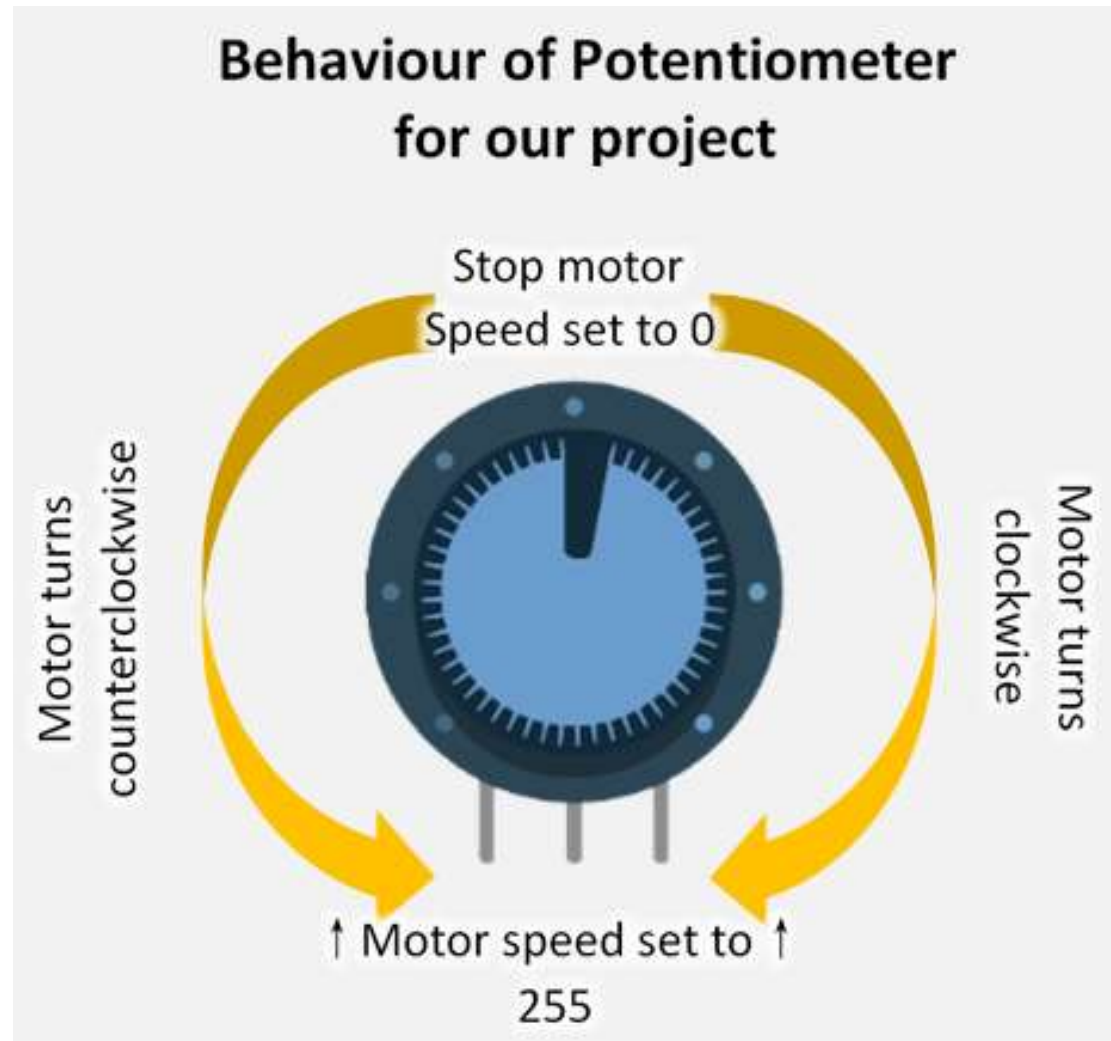
In this project, we will use an H-Bridge (Model L293D) with an Arduino to control the direction of the rotation of the motor based on the position of the potentiometer.

Pin #9 from our Arduino will be used to control the speed of the motor attached (output 5v). Pins #11 and pin #10 are used to control which direction the motor will turn. If you have pin #11 high (output 5v) the motor will spin one direction, while if you output high on pin #10 it will spin the other direction. The motor will not turn if both pin #10 and pin #11 are high (outputting 5v) since you are telling the h-bridge to go clockwise and counter clockwise at the same time.

The potentiometer will be used at the 12 o'clock position to stop all motor movement. If you turn the potentiometer counterclockwise, the motor will spin in that direction as well. If you turn the potentiometer clockwise the motor will spin in the direction as well. As you rotate the potentiometer more and more clockwise or counterclockwise, the motor will increase in speed until you reach the limit of how far it can move in either direction.

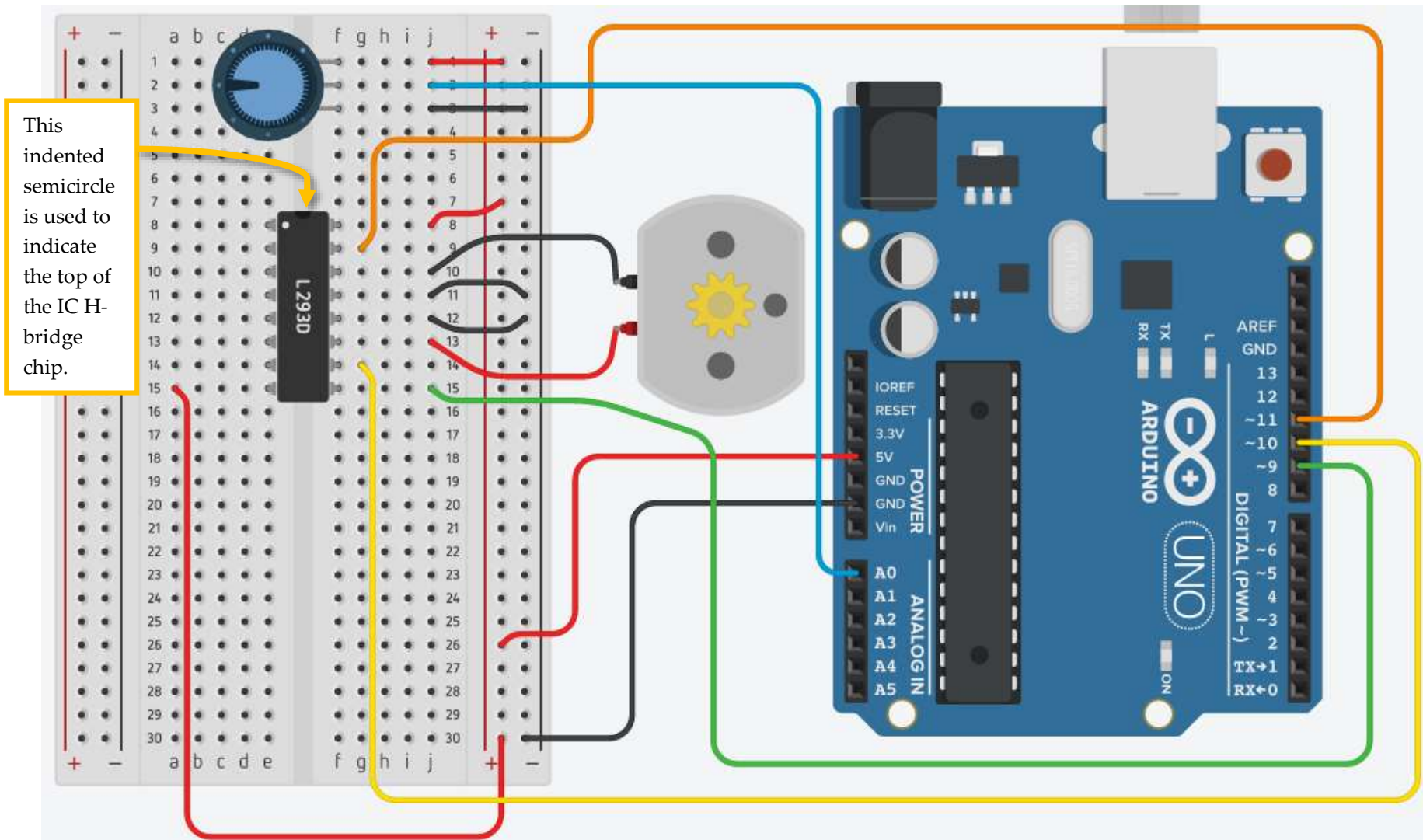
Required Components

					
1x Arduino	1x Breadboard	12x Jumper Wires	1x Potentiometer	1x H-Bridge	1x DC Motor



The above diagram above is used to illustrate the movement of the motor when you rotate the potentiometer for this project.

Wiring



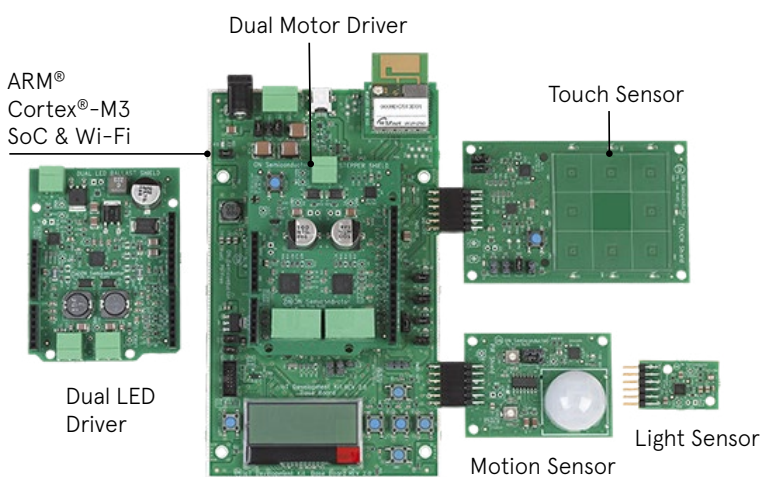
IoT Development Kit (IDK)

CONFIGURABLE RAPID PROTOTYPING PLATFORM FOR INDUSTRIAL IoT, SMART CITY/BUILDING, AND mHealth APPLICATIONS

The IDK is a fully configurable platform that enables engineers to design and offer differentiated IoT products and systems for a broad range of end applications, including smart home/building, smart city, industrial automation and mHealth.

By attaching different daughter cards to the IDK baseboard, a wealth of connectivity (SIGFOX, Bluetooth® Low Energy, Ethernet and 802.15.4 based radios enabling ZigBee, etc.), sensor (Smart Passive Sensor, Motion, Ambient Light, Proximity, Heart Rate, etc.) and actuator (Dual Stepper, BLDC and Power Stage or LED and Ballast) options can be added to the system. This means that compromises do not have to be made and the most suitable technology for a specific application can be chosen.

The IDK gives engineers a development resource that combines advanced IC technology with a sophisticated software framework in order to significantly aid 'device-to-cloud' IoT deployment, accelerating prototyping and shortening time-to-market cycles.



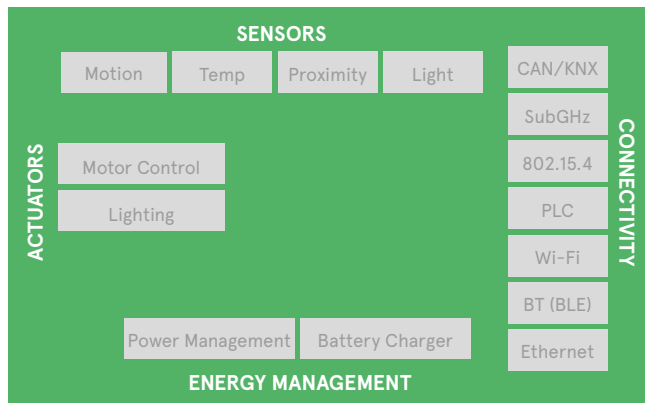
FEATURES & BENEFITS

- Comprehensive portfolio of sensors, connectivity and actuator devices
- Individual API for each and every device
- Complex C Code examples adapted to multiple applications
- Integrated development environment
- Full documentation of system hardware and software design
- Cloud software
- Ready to use for fast turnaround from concept to production

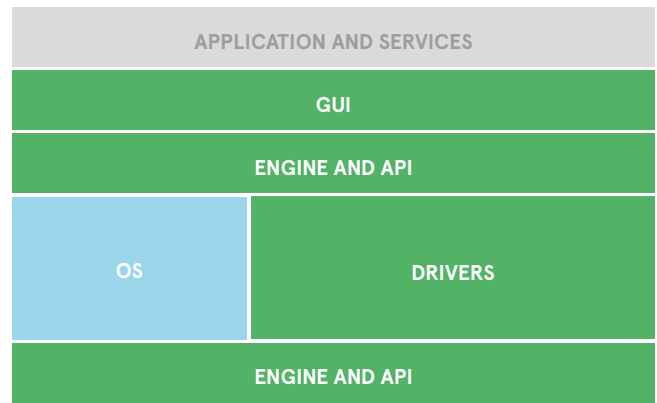
APPLICATIONS

- Motor Control
- Climate & Sense Control
- Lighting & Sense Control
- Process Control
- Motion Control
- Position Control
- Patient Monitoring
- Healthcare

HARDWARE



SOFTWARE FRAMEWORK



MODULES CAPABILITIES

Baseboard	BB-GEVK
Wireless Connectivity: SIGFOX EU	EU-SIGFOX-GEVB
Wireless Connectivity: SIGFOX US	US-SIGFOX-GEVB
Wired Connectivity: Power over Ethernet	POE-GEVB
Wired Connectivity: CAN	CAN-GEVB
Wireless Connectivity: Bluetooth Low Energy	BLE-IOT-GEVB
Sensor: PIR motion	PIR-GEVB
Sensor: Ambient light	ALS-GEVB
Sensor: Touch/proximity/level	TS-GEVB
Sensor: Battery Free Sensor	SPS-READER-GEVK
Sensor: Multi Sensor Board	MULTI-SENSE-GEVB
Actuator: Dual stepper motor	D-STPR-GEVK
Actuator: Dual LED + ballast	D-LED-B-GEVK
Actuator: BLDC motor control + power stage	BLDC-GEVK

SOFTWARE DEVELOPMENT TOOL

Development Environment (IDE)	Eclipse (Mars) + CDT + GNU Arm® for Eclipse
Supported Operating Systems	Windows®
Baseboard Processor	Arm Cortex®-M3 processor
Shield Boards	Connectivity, Sensors, Actuators
Operating System / RTOS	Arm mBed™
Toolchain	GNU tools for ARM
Debugger	gdb
Libraries	mBed, ON Semiconductor shield board libraries
IoT/Cloud Platform Support	Carriots, compatible with most commercial cloud platforms
IoT Protocols	REST, MQTT, HTTP
Connectivity Protocols	SIGFOX, Thread, EnOcean, BLE, MBUS, PoE, Wi-Fi, CAN, Zigbee®

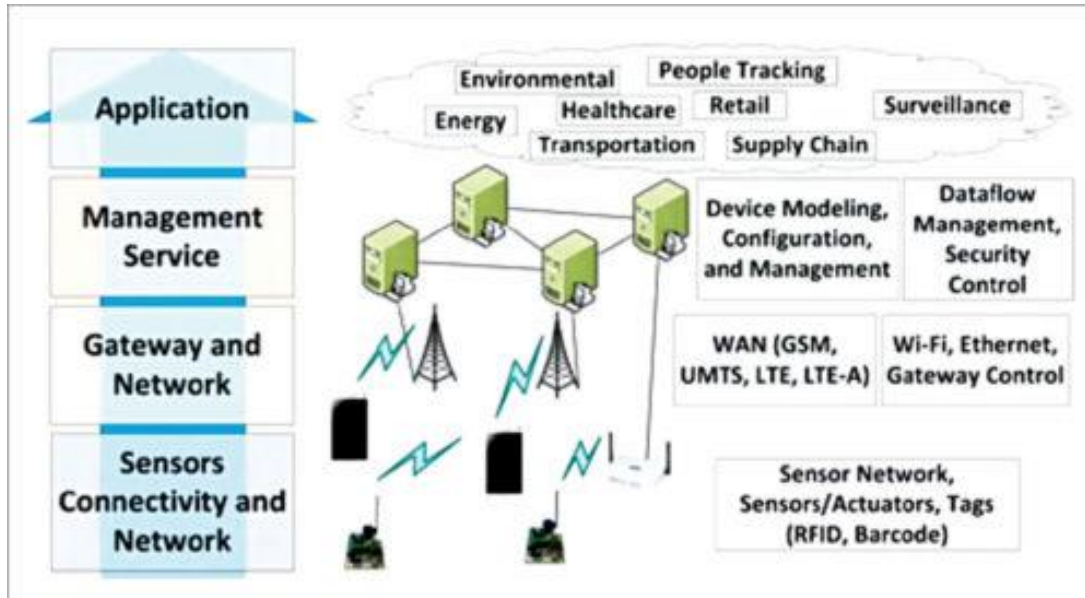


Fig. 2 IoT Architecture Layers

The main purpose of sensors is to collect data from the surrounding environment. Sensors, or ‘things’ of the IoT system, form the front end. These are connected directly or indirectly to IoT networks after signal conversion and processing. But all sensors are not the same and different IoT applications require different types of sensors. For instance, digital sensors are straightforward and easy to interface with a microcontroller using Serial Peripheral Interface (SPI) bus. But for analogue sensors, either analogue-to-digital converter (ADC) or Sigma-Delta modulator is used to convert the data into SPI output.

2.1 Types of sensors

All the parameters i.e. the Sensors (which give inputs to the Computers), the Computers (the brains of the system) and the mechanics (the outputs of the system like engines and motors) are equally important in building a successful automated system. Sensor as an input device which provides an output (signal) with respect to a specific physical quantity (input). Sensor means that it is part of a bigger system which provides input to a main control system (like a Processor or a Microcontroller).

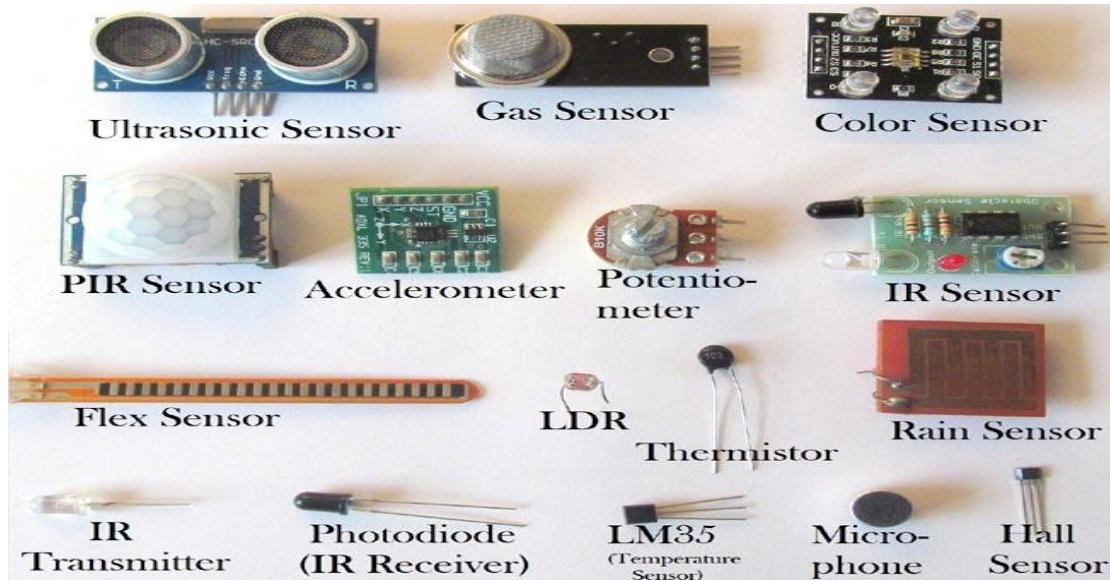


Fig 3.Examples of Sensors

1.IR LED

It is also called as IR Transmitter. It is used to **emit Infrared rays**. The range of these frequencies are greater than the microwave frequencies (i.e. >300GHz to few hundreds of THz). The rays generated by an infrared LED can be sensed by Photodiode explained below. **The pair of IR LED and photodiode is called IR Sensor.**

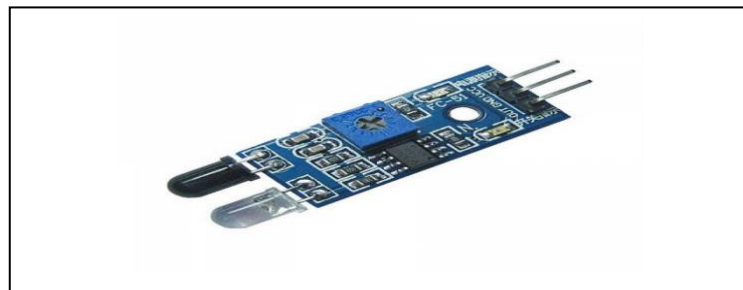


Fig 4. LED sensor

Some of the applications of Proximity Sensors are Mobile Phones, Cars (Parking Sensors), industries (object alignment), **Ground Proximity in Aircrafts, etc. Proximity Sensor in Reverse Parking is implemented in this Project: Reverse Parking Sensor Circuit.**

4.LDR (Light Dependent Resistor)

As the name itself specifies that the resistor that depends upon the light intensity. It works on the principle of photoconductivity which means the conduction due to the light. It is generally made up of Cadmium sulfide. **When light falls on the LDR, its resistance decreases and acts similar to a conductor and when no light falls on it, its resistance is almost in the range of $M\Omega$ or ideally it acts as an open circuit.** One note should be considered with LDR is that it won't respond if the light is not exactly focused on its surface.

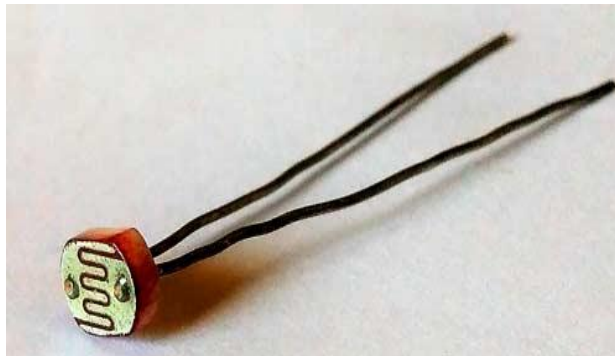


Fig. 7 LDR

With a proper circuitry using a transistor it can be used to detect the availability of light. A voltage divider biased transistor with R2 (resistor between base and emitter) replaced with an LDR can work as a light detector.

5.Thermistor (Temperature Sensor)

A thermistor can be used to *detect the variation in temperature*. It has a negative temperature coefficient that means when the temperature increases the resistance decreases. So, the thermistor's resistance can be varied with the rise in temperature which causes more current

here to measure the load. On a flexible board, a wire is arranged in a zig-zag manner as shown in the figure below. So, when the pressure is applied to that particular board, it bends in a direction causing the change in overall length and cross-sectional area of the wire. This leads to change in resistance of the wire. The resistance thus obtained is very minute (few ohms) which can be determined with the help of the Wheatstone bridge. The strain gauge is placed in one of the four arms in a bridge with the remaining values unchanged. Therefore, when the pressure is applied to it as the resistance changes the current passing through the bridge varies and pressure can be calculated.

Strain gauges are majorly used to calculate the amount of pressure that an airplane wing can withstand and it is also used to measure the number of vehicles allowable on a particular road etc.

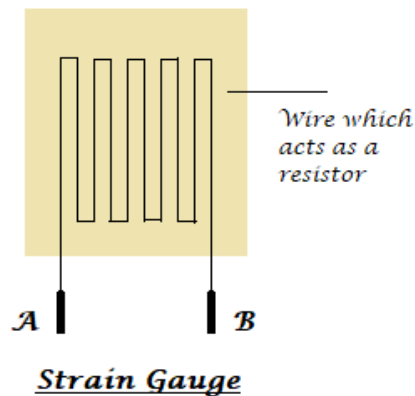


Fig .10 Strain Guage

8.Load Cell (Weight Sensor)

Load cells are similar to strain gauges which measure the physical quantity like force and give the output in form of electrical signals. When some tension is applied on the load cell its structure varies causing the change in resistance and finally, its value can be calibrated using a Wheatstone bridge. Here is the project on how to measure weight using Load cell.

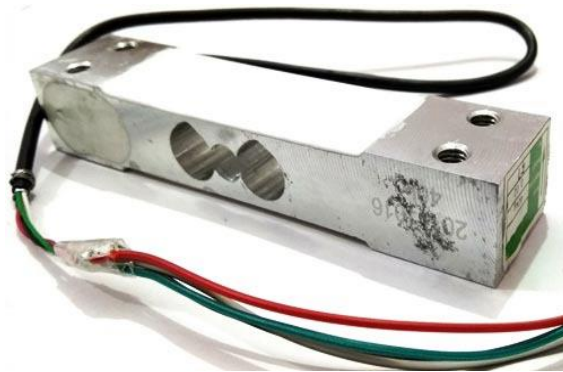


Fig 11. Load Cell

9. Potentiometer

A potentiometer is used to detect the position. It generally has various ranges of resistors connected to different poles of the switch. A potentiometer can be either rotary or linear type. In rotary type, a wiper is connected to a long shaft which can be rotated. When the shaft has rotated the position of the wiper alters such that the resultant resistance varies causing the change in the output voltage. Thus the output can be calibrated to detect the change its position.



Fig 12. Potentiometer

10. Encoder

To detect the change in the position an encoder can also be used. It has a circular rotatable disk-like structure with specific openings in between such that when the IR rays or light rays

pass through it only a few light rays get detected. Further, these rays are encoded into a digital data (in terms of binary) which represents the specific position.

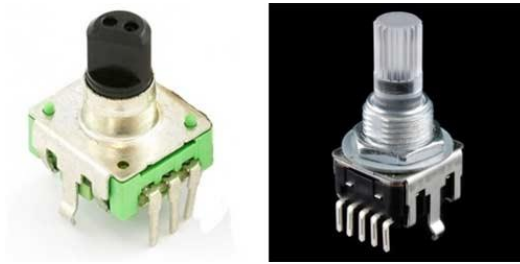


Fig 13.Encoder

11 Hall Sensor

The name itself states that it is the sensor which works on the Hall Effect. It can be defined as when a magnetic field is brought close to the current carrying conductor (perpendicular to the direction of the electric field) then a potential difference is developed across the given conductor. Using this property a *Hall sensor is used to detect the magnetic field* and gives output in terms of voltage. Care should be taken that the Hall sensor can detect only one pole of the magnet.



Fig 14.Hall sensor

The hall sensor is used in few smartphones which are helpful in turning off the screen when the flap cover (which has a magnet in it) is closed onto the screen. Here is one practical application of Hall Effect sensor in Door Alarm.

12. Flex Sensor

A FLEX sensor is a transducer which *changes its resistance when its shape is changed or when it is bent*. A FLEX sensor is 2.2 inches long or of finger length. Simply speaking the sensor terminal resistance increases when it's bent. This change in resistance can do no good unless we can read them. The controller at hand can only read the changes in voltage and nothing less, for this, we are going to use voltage divider circuit, with that we can derive the resistance change as a voltage change.

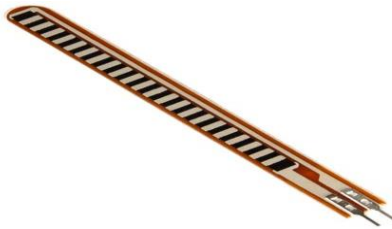


Fig 15. Flex sensor

13. Microphone (Sound Sensor)

Microphone can be seen on all the smartphones or mobiles. It can detect the audio signal and convert them into small voltage (mV) electrical signals. A microphone can be of many types like condenser microphone, crystal microphone, carbon microphone etc. each type of microphone work on the properties like capacitance, piezoelectric effect, resistance respectively. Let us see the operation of a crystal microphone which works on the piezoelectric effect. A bimorph crystal is used which under pressure or vibrations produces proportional alternating voltage. A diaphragm is connected to the crystal through a drive pin such that when the sound signal hits the diaphragm it moves to and fro, this movement changes the position of the drive pin which causes vibrations in the crystal thus an alternating voltage is generated with respect to the applied sound signal. The obtained voltage is fed to an amplifier in order to increase the overall strength of the signal.



Fig 16. Microphone

14. Ultrasonic sensor

Ultrasonic means nothing but the range of the frequencies. Its range is greater than audible range (>20 kHz) so even it is switched on we can't sense these sound signals. Only specific speakers and receivers can sense those ultrasonic waves. *This ultrasonic sensor is used to calculate the distance between the ultrasonic transmitter and the target and also used to measure the velocity of the target.*

Ultrasonic sensor HC-SR04 can be used to measure distance in the range of 2cm-400cm with an accuracy of 3mm. Let's see how this module works. The HCSR04 module generates a sound vibration in ultrasonic range when we make the 'Trigger' pin high for about 10us which will send an 8 cycle sonic burst at the speed of sound and after striking the object, it will be received by the Echo pin. Depending on the time taken by sound vibration to get back, it provides the appropriate pulse output. We can calculate the distance of the object based on the time taken by the ultrasonic wave to return back to the sensor.



Fig 17. Ultrasonic sensor

There are many applications with the ultrasonic sensor. We can make use of it avoid obstacles for the automated cars, moving robots etc. The same principle will be used in the RADAR for

detecting the intruder missiles and airplanes. A mosquito can sense the ultrasonic sounds. So, ultrasonic waves can be used as mosquito repellent.

15.Touch Sensor

In this generation, we can say that almost all are using smartphones which have widescreen that too a screen which can sense our touch. So, let's see how this touchscreen works. Basically, there are two types of touch sensors *resistive based and a capacitive based touch screens*. Let's know about working of these sensors briefly.

The *resistive touch screen* has a resistive sheet at the base and a conductive sheet under the screen both of these are separated by an air gap with a small voltage applied to the sheets. When we press or touch the screen the conductive sheet touches the resistive sheet at that point causing current flow at that particular point, the software senses the location and relevant action is performed.

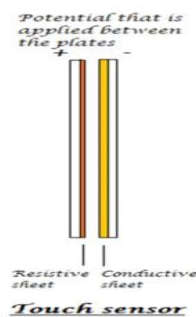


Fig 18.Touch sensor

16.PIR sensor

PIR sensor stands for **Passive Infrared sensor**. These are *used to detect the motion* of humans, animals or things. We know that infrared rays have a property of reflection. When an infrared ray hits an object, depending upon the temperature of the target the infrared ray properties changes, this received signal determines the motion of the objects or the living beings. Even if the shape of the object alters, the properties of the reflected infrared rays can differentiate the objects precisely. Here is the complete working or PIR sensor.



Fig 19.PIR Sensor

17.Accelerometer (Tilt Sensor)

An accelerometer sensor can sense the tilt or movement of it in a particular direction. It works based on the acceleration force caused due to the earth's gravity. The tiny internal parts of it are such sensitive that those will react to a small external change in position. It has a piezoelectric crystal when tilted causes disturbance in the crystal and generates potential which determines the exact position with respect to X, Y and Z axis.

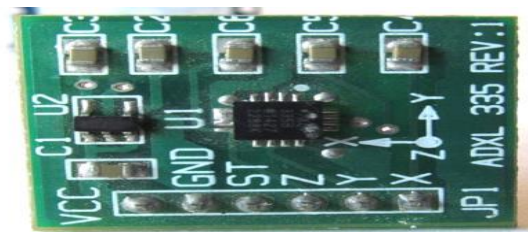


Fig 20.Accelerometer

These are commonly seen in mobiles and laptops in order to avoid breakage of processors leads. When the device falls the accelerometer detects the falling condition and does respective action based on the software.

18.Gas sensor

In industrial applications gas sensors plays a major role in **detecting the gas leakage**. If no such device is installed in such areas it ultimately leads to an unbelievable disaster. These gas sensors are classified into various types based on the type of gas that to be detected. Let's see

how this sensor works. Underneath a metal sheet there exists a sensing element which is connected to the terminals where a current is applied to it. When the gas particles hit the sensing element, it leads to a chemical reaction such that the resistance of the elements varies and current through it also alters which finally can detect the gas.



Fig 21. Gas Sensor

So finally, we can conclude that sensors are not only used to make our work simple to measure the physical quantities, making the devices automated but also used to help living beings with disasters.

19. Resistive Sensors

Resistive sensors, such as the potentiometer, have three terminals: power input, grounding terminal, and variable voltage output. These mechanical devices have varied resistance that can be changed through movable contact with its fixed resistor. Output from the sensor varies depending on whether the movable contact is near the resistor's supply end or ground end. Thermistors are also variable resistors, although the resistance of the sensor varies with temperature

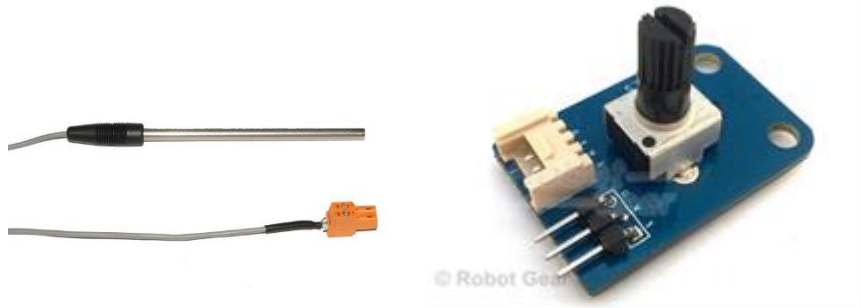


Fig 22 Resistive Sensors

20.Voltage generating sensors

Voltage-generating sensors, such as piezo electrics, generate electricity by pressure with types of crystals like quartz. As the crystal flexes or vibrates, AC voltage is produced. Knock sensors utilize this technology by sending a signal to an automobile's on-board computer that engine knock is happening. The signal is generated through crystal vibration within the sensor, which is caused by cylinder block vibration. The computer, in turn, reduces the ignition timing to stop the engine knock.

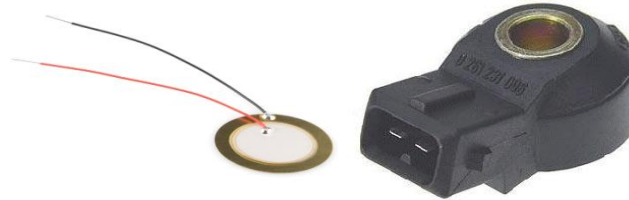


Fig 23.Voltage Generating Sensors

21.Switch Sensors

Switch sensors are composed of a set of contacts that open when close to a magnet. A reed switch is a common example of a switch sensor and is most commonly used as a speed or position sensor. As a speed sensor, a magnet is attached to the speedometer cable and spins along with it. Each time one of the magnet's poles passes the reed switch, it opens and then closes. How fast the magnet passes allows the sensor to read the vehicle's speed.



Fig 24.Switch Sensors

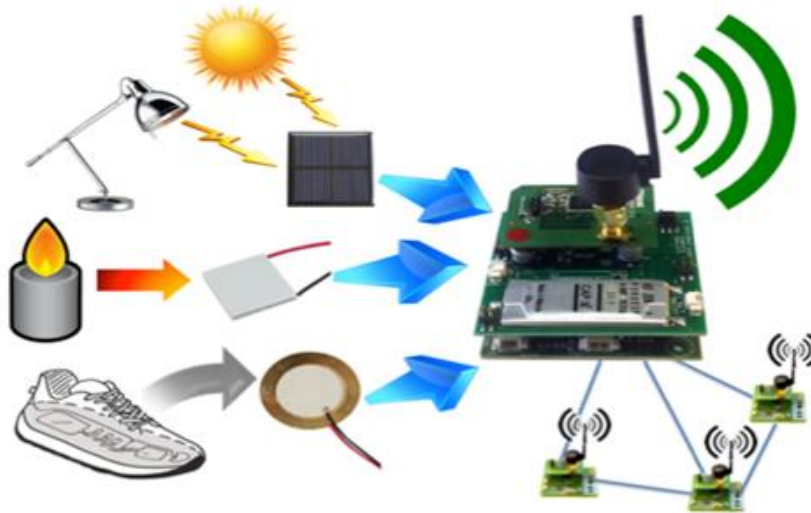
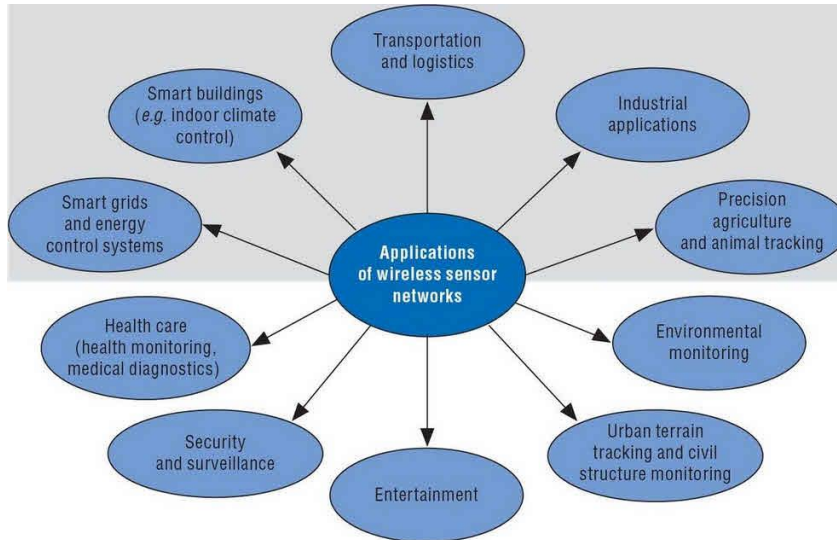


Fig. 25 Wireless Sensor Network

The sensor node is a multi-functional, energy-efficient wireless device. The applications of nodes in industrial are widespread. A collection of sensor nodes collects the data from the surroundings to achieve specific application objectives. The communication between nodes can be done with each other using transceivers. In a wireless sensor network, the number of nodes can be in the order of hundreds/ even thousands. In contrast with sensor networks, Ad Hoc networks will have fewer nodes without any structure.

The most common wireless sensor network architecture follows the OSI architecture Model. The architecture of the WSN includes five layers and three cross layers. Mostly in sensor networks, we require five layers, namely application, transport, network, data link & physical layer. The three cross planes are namely power management, mobility management, and task management. These layers of the WSN are used to accomplish the network and make the sensors work together in order to raise the complete efficiency of the network. The architecture used in WSN is sensor network architecture. This kind of architecture is applicable in different places such as hospitals, schools, roads, buildings as well as it is used in different applications such as security management, disaster management & crisis management, etc. There are two types of architectures used in wireless sensor networks which include the following. There are 2 types of wireless sensor architectures: Layered Network Architecture, and Clustered Architecture.



- Military Applications
- Health Applications
- Environmental Applications
- Home Applications
- Commercial Applications
- Area monitoring
- Health care monitoring
- Environmental/Earth sensing
- Air pollution monitoring
- Forest fire detection
- Landslide detection
- Water quality monitoring
- Industrial monitoring

4. REQUIREMENTS OF IOT SENSOR

Smart devices, powered by the hyper-connected Internet of Things (IoT), are becoming ever more prevalent and pervasive in our lives, and the trend will only continue. Every industry is seeking ways to use device-enabled insights to improve the lives of their customers, and the

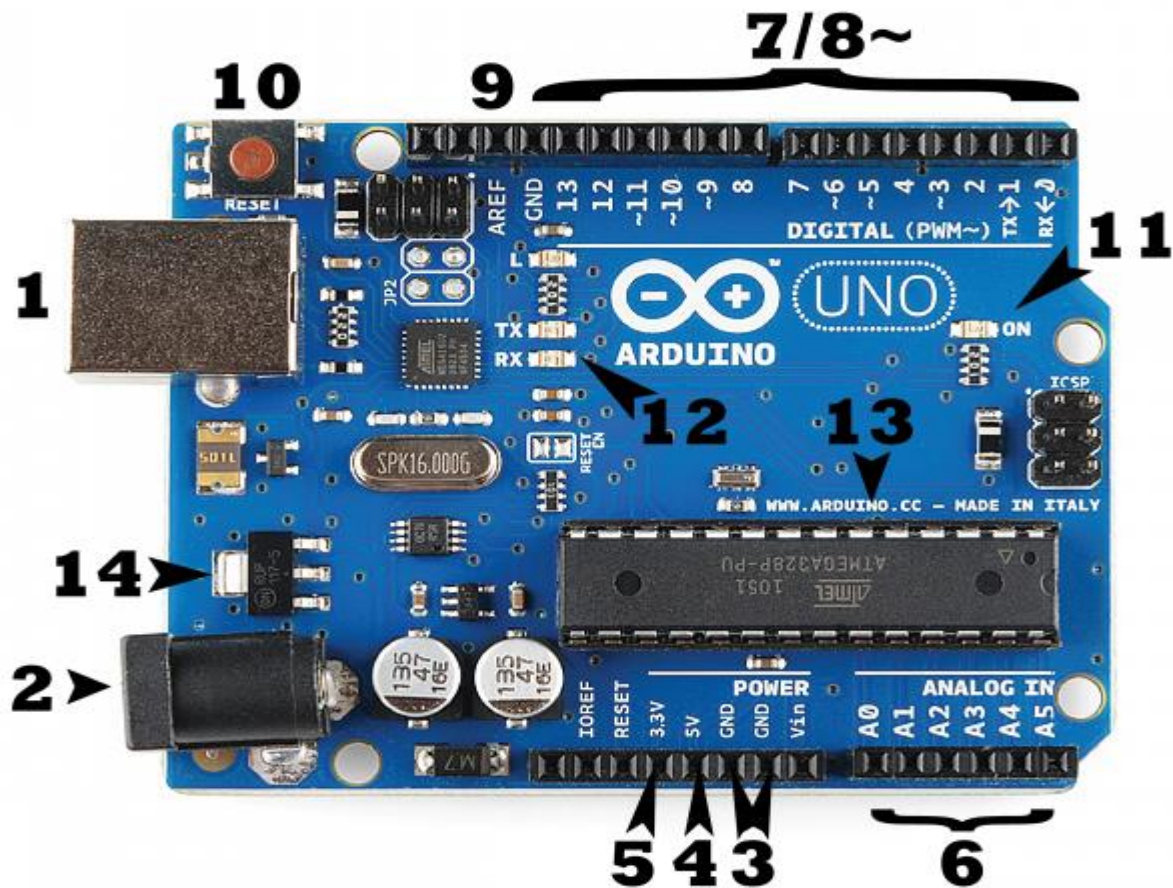


Fig. 1 Arduino Board

Power (USB / Barrel Jack):

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack. In the picture above the USB connection is labeled (1) and the barrel jack is labeled (2). The USB connection is also how you will load code onto your Arduino board.

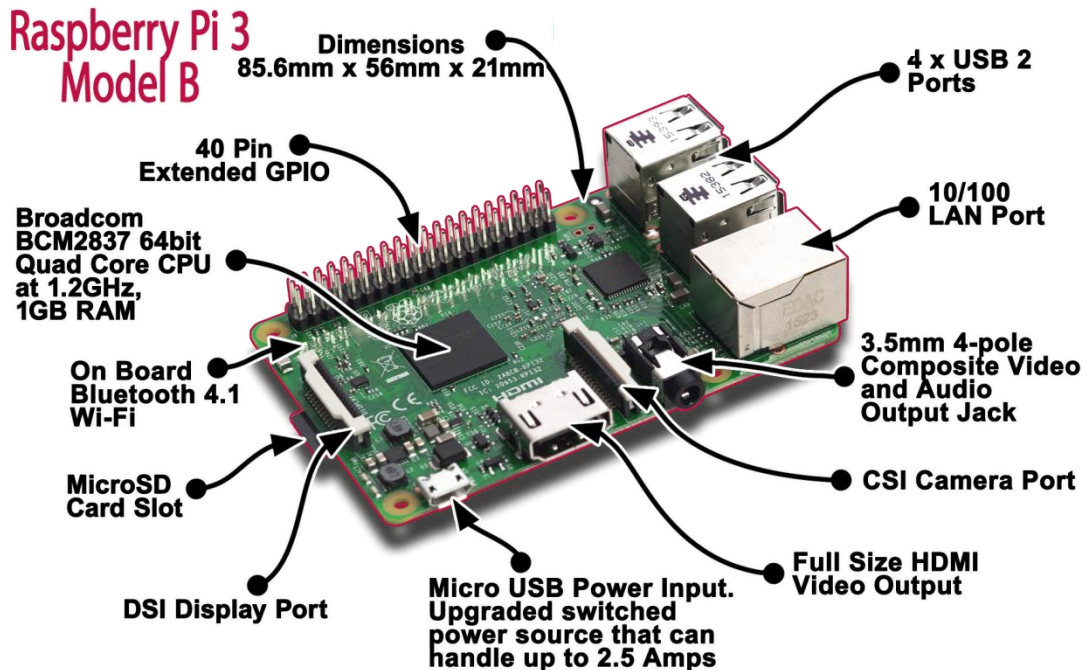


Fig. 4 Raspberry Pi Elements

As for the specifications, the Raspberry Pi is a credit card-sized computer powered by the Broadcom BCM2835 system-on-a-chip (SoC). This SoC includes a 32-bit ARM1176JZFS processor, clocked at 700MHz, and a Videocore IV GPU. It also has 256MB of RAM in a POP package above the SoC. The Raspberry Pi is powered by a 5V micro USB AC charger or at least 4 AA batteries (with a bit of hacking). While the ARM CPU delivers real-world performance similar to that of a 300MHz Pentium 2, the Broadcom GPU is a very capable graphics core capable of hardware decoding several high definition video formats. The Raspberry Pi model available for purchase at the time of writing — the Model B — features HDMI and composite video outputs, two USB 2.0 ports, a 10/100 Ethernet port, SD card slot, GPIO (General Purpose I/O Expansion Board) connector, and analog audio output (3.5mm headphone jack). The less expensive Model A strips out the Ethernet port and one of the USB ports but otherwise has the same hardware. Raspberry Pi Basics: installing Raspbian and getting it up and running.

Please remember that after booting the Pi, there might be situations when the user credentials like the "username" and password will be asked. Raspberry Pi comes with a default user name and password and so always use it whenever it is being asked. The credentials are:

login: pi

password: raspberry

When the Pi has been booted for the first time, a configuration screen called the "Setup Options" should appear and it will look like the image below.

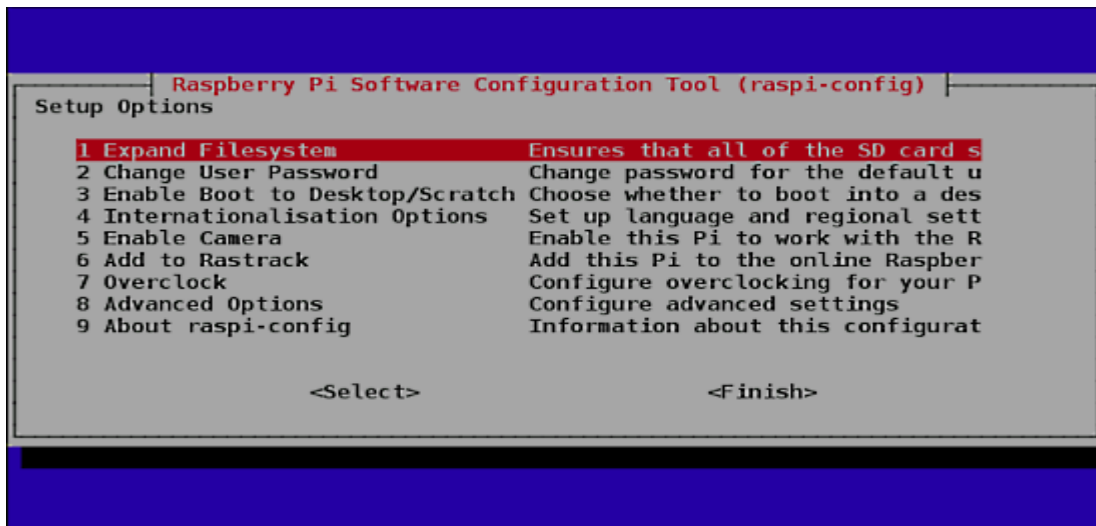


Fig. 6 Raspberry Configuration

If you have missed the "Setup Options" screen, its not a problem, you can always get it by typing the following command in the terminal.

sudo raspi-config

Once you execute this command the "Setup Options" screen will come up as shown in the image above.

Now that the Setup Options window is up, we will have to set a few things. After completing each of the steps below, if it asks to reboot the Pi, please do so. After the reboot, if you don't get the "Setup Options" screen, then follow the command given above to get the screen/window.

- The first thing to do:

select the first option in the list of the **setup options window**, that is select the "**Expand Filesystem**" option and hit the enter key. We do this to make use of all the space present on the SD card as a full partition. All this does is, expand the OS to fit the whole space on the SD card which can then be used as the storage memory for the Pi

- The second thing to do:

Select the third option in the list of the setup options window, that is select the "**Enable Boot To Desktop/Scratch**" option and hit the enter key. It will take you to another window called the "**choose boot option**" window that looks like the image below.

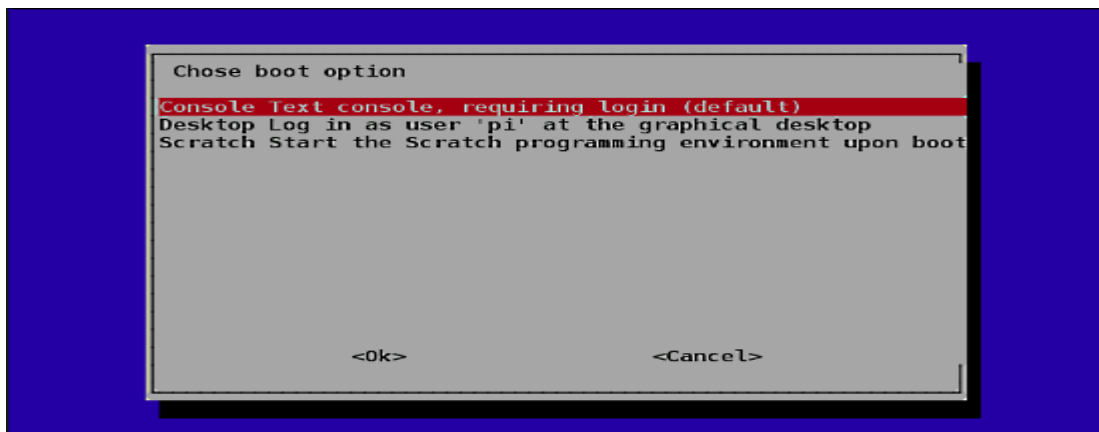


Fig. 7 Boot Options

In the "**choose boot option window**", select the second option, that is, "**Desktop Log in as user 'pi' at the graphical desktop**" and hit the enter button. Once done you will be taken back to the "**Setup Options**" page, if not select the "OK" button at the bottom of this window and you will be taken back to the previous window. We do this because we want to boot into the desktop environment which we are familiar with. If we don't do this step then the Raspberry Pi boots into a terminal each time with no GUI options. Once, both the steps are done, select the "**finish**" button at the bottom of the page and it should reboot automatically. If it doesn't, then use the following command in the terminal to reboot.

sudo reboot

Updating the firmware

After the reboot from the previous step, if everything went right, then you will end up on the desktop which looks like the image below.

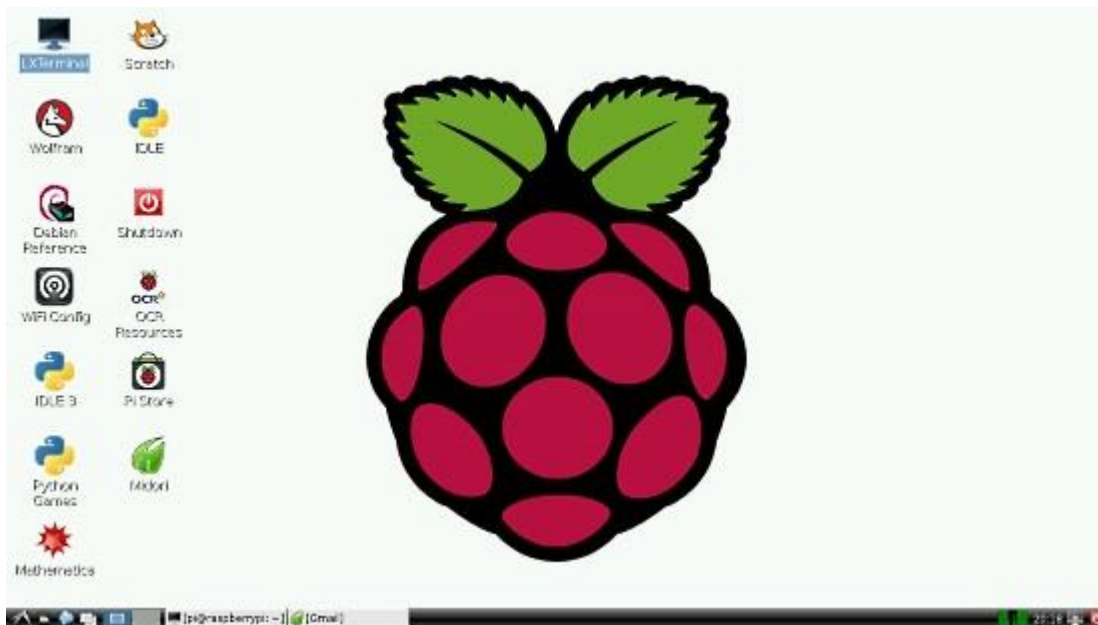


Fig. 8 Raspberry Desktop

Once you are on the desktop, open a terminal and enter the following command to update the firmware of the Pi.

sudo rpi-update

Updating the firmware is necessary because certain models of the Pi might not have all the required dependencies to run smoothly or it may have some bug. The latest firmware might have the fix to those bugs, thus its very important to update it in the beginning itself.

5 Conclusion

So, we have covered the steps to get the Pi up and running. This method works on all the different models of Raspberry Pi (model A, B, B+ and also RPi 2) as Raspbain was made to be supported on all models. However, while installing other software or libraries, the procedure

might change a bit while installing depending on the model of the Pi or the version of Raspbian itself. The concept of Raspberry is to keep trying till you get the result or build that you want. This might involve a lot of trial and error but spending the time will be worth it. The actual usage doesn't end here. This is just the beginning. It is up to you to go ahead to build something amazing out of it.

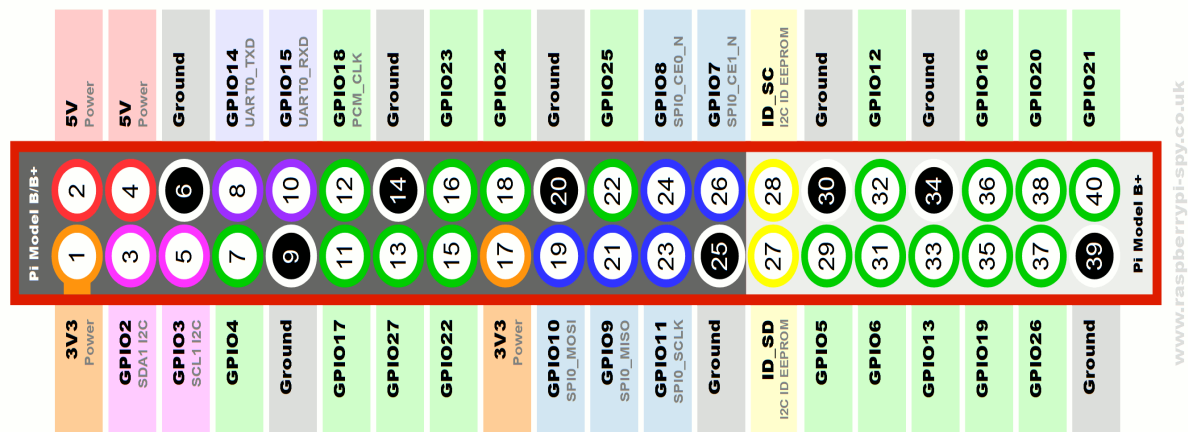


Fig. 9 GPIO Pins

GPIO:

Act as both digital output and digital input.

Output: turn a GPIO pin high or low.

Input: detect a GPIO pin high or low

Installing GPIO library:

Open terminal

Enter the command “`sudo apt-get install python-dev`” to install python development

Enter the command “`sudo apt-get install python-rpi.gpio`” to install GPIO library.

Basic python coding:



Fig. 10 Using RealVNC to access the Raspberry Pi's graphical desktop

Enable VNC

You will need to interact with your Pi in order to turn on the VNC server. To do this, you have several options:

- Connect a keyboard, mouse, and monitor. Click the **Terminal** icon on the top left of the desktop to open a terminal window.

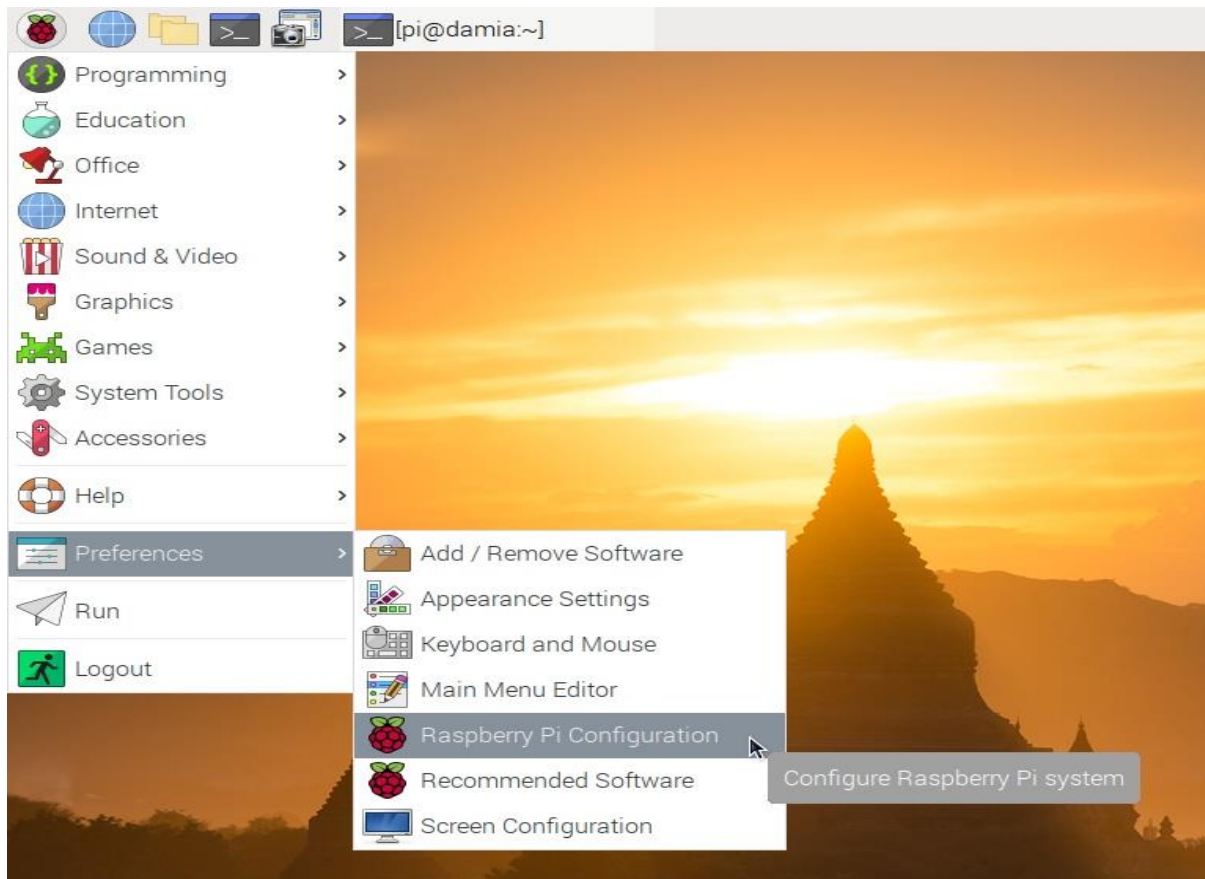


Fig. 12 Raspberry Pi Configuration

Configure your Raspberry Pi system using the Raspberry Pi Configuration tool. The tool will have several tabs, click on the tab “Interfaces”, to see the available options there. Note that VNC is disabled by default: Raspberry Pi Configuration tool, showing VNC as disabled (“Enable remote access to this Pi using RealVNC”) Click on the enable radio button, and then on ok button. VNC is now enabled in the Interfaces tab of the Raspberry Pi Configuration tool. Watch the task bar at the top of the screen. A new icon with a V2 symbol will appear. Raspberry Pi taskbar top right corner, before enabling VNC

CONNECT THE LED TO THE RASPBERRY PI

Components:

- Raspberry Pi
- One LED
- One 330 Ohm resistor
- Jumper wires
- Breadboard

Connect the components as shown in the wiring diagram below.

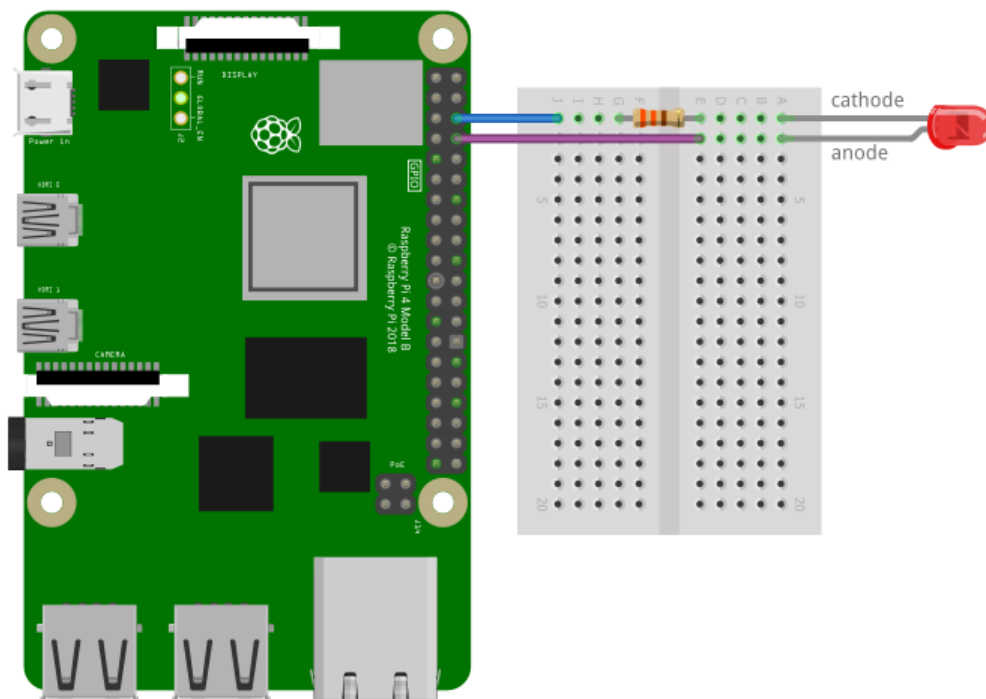


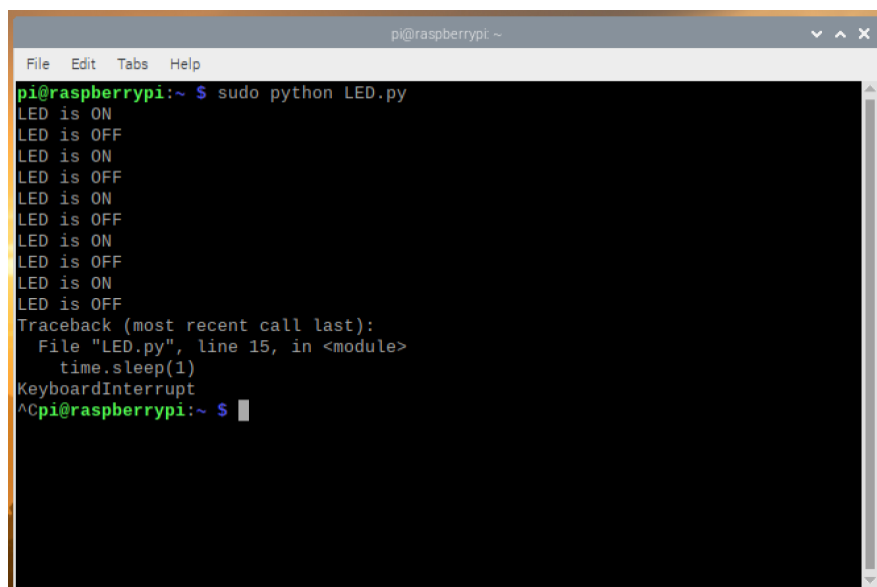
Fig. 16 Raspberry pi with LED

The 330 Ohm resistor is a current limiting resistor. Current limiting resistors should always be used when connecting LEDs to the GPIO pins. If an LED is connected to a GPIO pin without a resistor, the LED will draw too much current, which can damage the Raspberry Pi or burn out the LED. [Here](#) is a nice calculator that will give you the value of a current

sudo python LED.py

You should see the LED blinking on and off once every second.

You should also see a message in the terminal with “LED is ON“ when the LED is turned on, and “LED is OFF” when the LED is turned off.



```

pi@raspberrypi:~ $ sudo python LED.py
LED is ON
LED is OFF
LED is ON
LED is OFF
LED is ON
LED is OFF
LED is ON
LED is OFF
LED is ON
LED is OFF
LED is ON
LED is OFF
Traceback (most recent call last):
  File "LED.py", line 15, in <module>
    time.sleep(1)
KeyboardInterrupt
^Cpi@raspberrypi:~ $

```

Fig. 17 output

4.3 Using a PIR sensor

Humans and other animals emit radiation all the time. This is nothing to be concerned about, though, as the type of radiation we emit is infrared radiation (IR), which is pretty harmless at the levels at which it is emitted by humans. In fact, all objects at temperatures above absolute zero (-273.15C) emit infrared radiation. A PIR sensor detects changes in the amount of infrared radiation it receives. When there is a significant change in the amount of infrared radiation it detects, then a pulse is triggered. This means that a PIR sensor can detect when a human (or any animal) moves in front of it. The pulse emitted when a PIR detects motion needs to be amplified, and so it needs to be powered. There are three pins on the PIR: they should be

labelled Vcc, Gnd, and Out. These labels are sometimes concealed beneath the Fresnel lens (the white cap), which you can temporarily remove to see the pin labels.

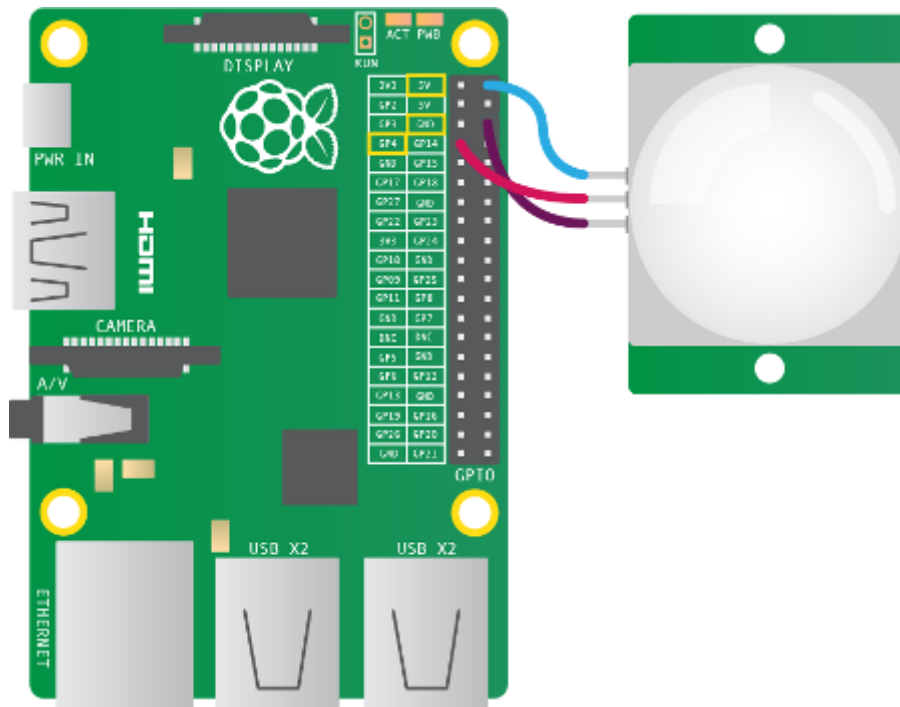


Fig. 18 Raspberry with PIR

1. As shown above, the Vcc pin needs to be attached to a 5V pin on the Raspberry Pi.
2. The Gnd pin on the PIR sensor can be attached to any ground pin on the Raspberry Pi.
3. Lastly, the Out pin needs to be connected to any of the GPIO pins.

Tuning a PIR

Most PIR sensors have two potentiometers on them. These can control the sensitivity of the sensors, and also the period of time for which the PIR will signal when motion is detected.

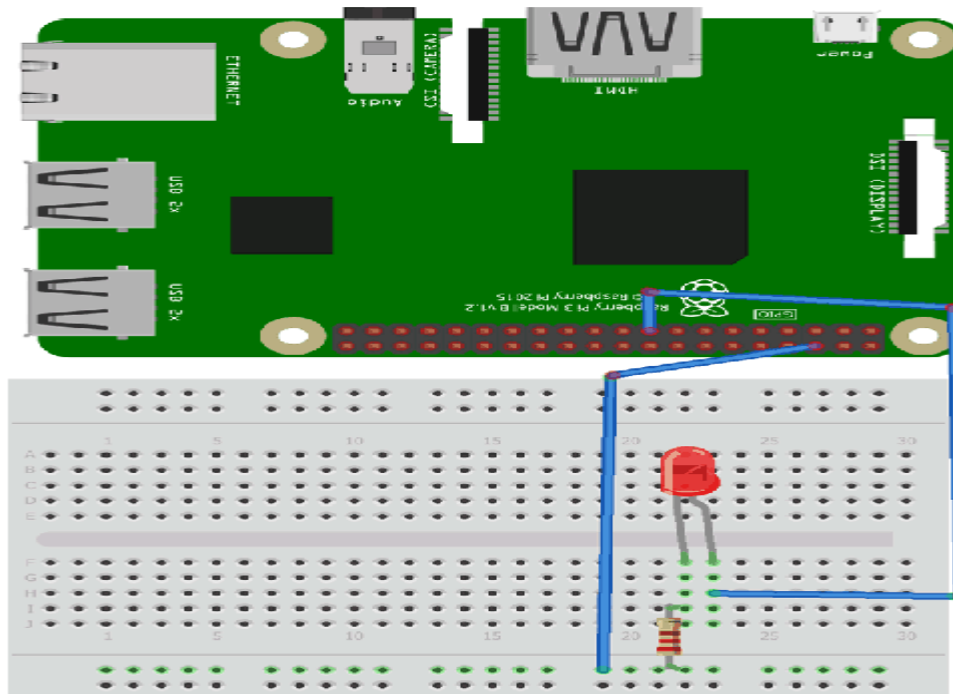


Fig 19 LED with Raspberry pi

Step 2: Installing WiringPi Library

WiringPi is a PIN-based GPIO access library written in C for the BCM2835, BCM2836, and BCM2837 SoC devices used in all Raspberry Pi versions. It's released under the GNU LGPLv3 license and is usable from C, C++, and RTB (BASIC) as well as many other languages with suitable wrappers.

1. First we will update our Pi with the latest versions of Raspbian using the command:

```
sudo apt-get update
```

2. Now we will install git by using this command:

```
sudo apt-get install git-core
```

3. Now obtain WiringPi using git by this command:

Here we have used `shell_exec()` command in php code, this command is used to run the shell command from the PHP script. Learn more about [shell_exec here](#). If you run the command inside `shell_exec` directly from the terminal of Raspberry pi, you can directly make GPIO pin 27 low or high. Below are two commands to test the LED directly from terminal.

```
/usr/local/bin/gpio -g write 27 0
```

```
/usr/local/bin/gpio -g write 27 1
```

After completing this, run the code in your browser by typing the IP address of raspberry pi in the browser. You will see 2 buttons - ON, OFF to control your LED by clicking these buttons.

5. LIQUID CRYSTAL DISPLAYS (LCD) WITH ARDUINO

The [Liquid Crystal library](#) allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.



Fig. 21 Output of the sketch in 16x2 LCD

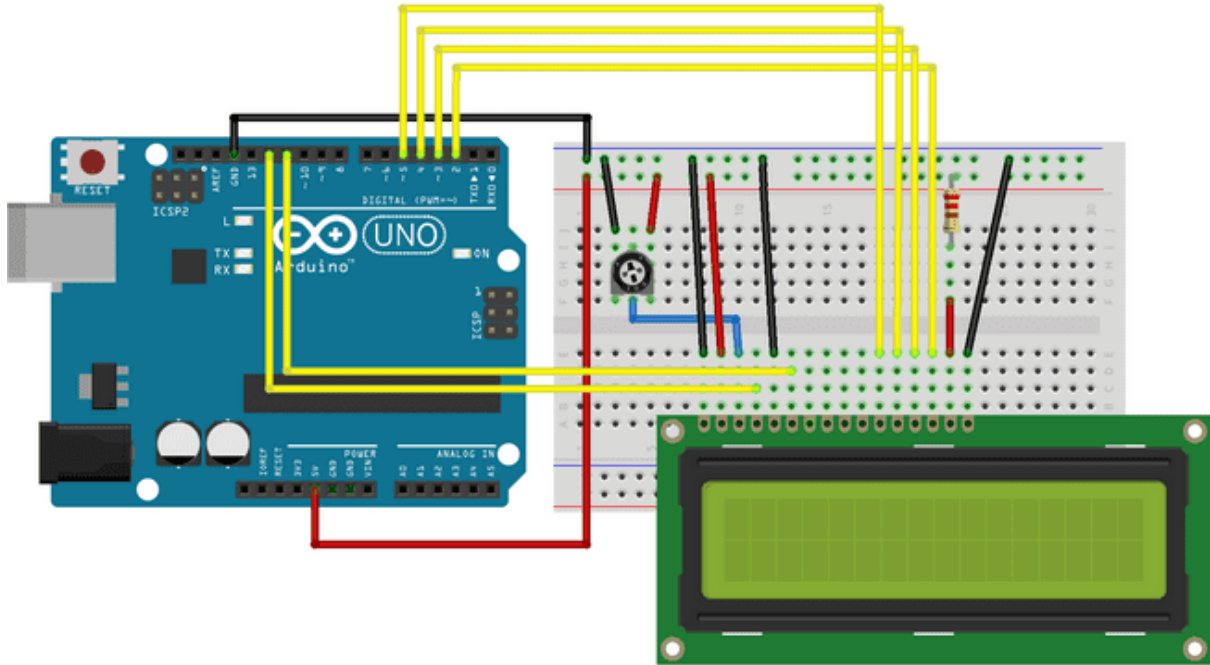


Fig. 22 Arduino with LCD

The circuit (made using Fritzing).

Schematic

The schematic (made using Fritzing).


```

for (int thisChar = 0; thisChar < 10; thisChar++) {

    lcd.print(thisChar);

    delay(500);

}

// turn off automatic scrolling

lcd.noAutoscroll();

// clear screen for the next loop:

lcd.clear();

}

```

6.ARDUIINO WITH KEYPAD

The buttons on a keypad are arranged in rows and columns. A 3X4 keypad has 4 rows and 3 columns, and a 4X4 keypad has 4 rows and 4 columns:



Fig. 23 Keypad

Beneath each key is a membrane switch. Each switch in a row is connected to the other switches in the row by a conductive trace underneath the pad. Each switch in a column is connected the same way – one side of the switch is connected to all of the other switches in that column by a conductive trace. Each row and column is brought out to a single pin, for a total of 8 pins on a 4X4 keypad:

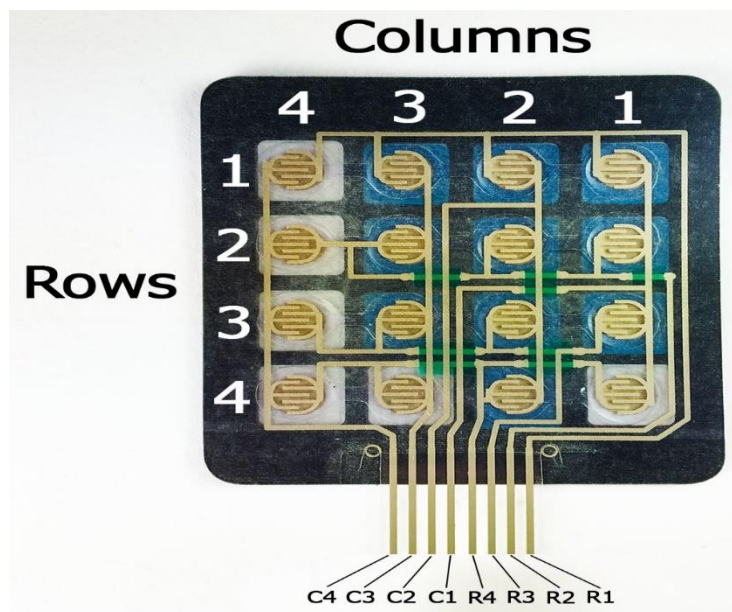


Fig. 24 keypad Structure

Pressing a button closes the switch between a column and a row trace, allowing current to flow between a column pin and a row pin. The Arduino detects which button is pressed by detecting the row and column pin that's connected to the button.

This happens in four steps:

1. First, when no buttons are pressed, all of the column pins are held HIGH, and all of the row pins are held LOW:

2. When a button is pressed, the column pin is pulled LOW since the current from the HIGH column flows to the LOW row pin:

3. The Arduino now knows which column the button is in, so now it just needs to find the row the button is in. It does this by switching each one of the row pins HIGH, and at the same time reading all of the column pins to detect which column pin returns to HIGH:

4. When the column pin goes HIGH again, the Arduino has found the row pin that is connected to the button:

The pin layout for most membrane keypads will look like this:

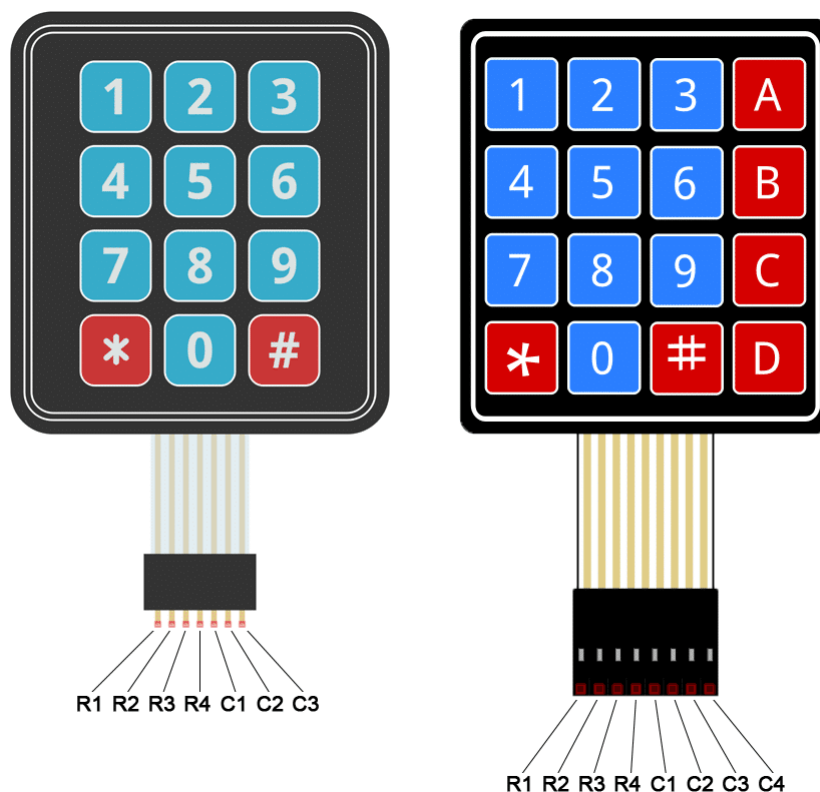


Fig 25 Pin Layout

```

if (customKey){

  Serial.println(customKey);

}

}

```

7.ARDUINO WITH SENSORS

7.1 Ultrasonic sensor

It works by sending sound waves from the transmitter, which then bounce off of an object and then return to the receiver. You can determine how far away something is by the time it takes for the sound waves to get back to the sensor. Let's get right to it!. Connections

The connections are very simple:

- VCC to 5V
- GND to GND
- Trig to pin 9
- Echo to pin 10

You can actually connect Trig and Echo to whichever pins you want, 9 and 10 are just the ones I'm using.

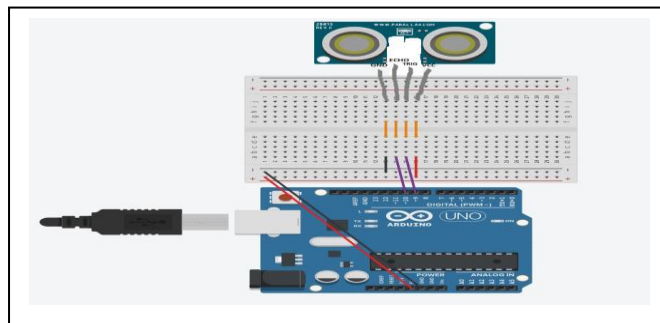


Fig 26. Ultra Sonic Sensor

$$\text{Centigrade temperature} = [(\text{analog voltage in mV}) - 500] / 10$$

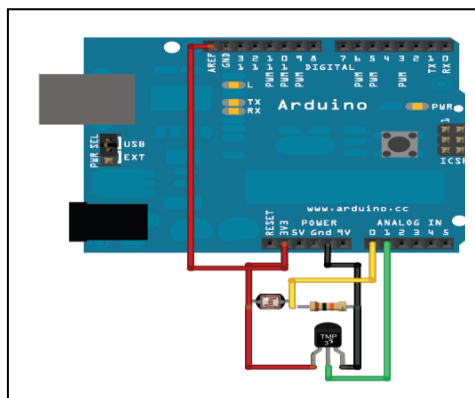


Fig 27 Temperature Sensor

This example code for Arduino shows a quick way to create a temperature sensor, it simply prints to the serial port what the current temperature is in both Celsius and Fahrenheit. For better results, using the 3.3v reference voltage as ARef instead of the 5V will be more precise and less noisy.

```
int sensorPin = 0;
```

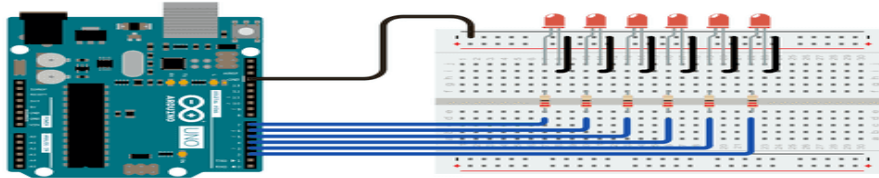


Fig. 1 Arduino with LED

Schematic:

The code below begins by utilizing a

```
for()
loop to assign digital pins 2-7 as outputs for the 6 LEDs used.
```

In the main loop of the code, two

```
for()
loops are used to loop incrementally, stepping through the LEDs, one by one, from pin 2 to pin seven. Once pin 7 is lit, the process reverses, stepping back down through each LED.
```

/*

For Loop Iteration

*/

```
int timer = 100; // The higher the number, the slower the timing.
```

```
void setup() {
```

// use a for loop to initialize each pin as an output:

```
for (int thisPin = 2; thisPin < 8; thisPin++) {
```

```
pinMode(thisPin, OUTPUT);
```

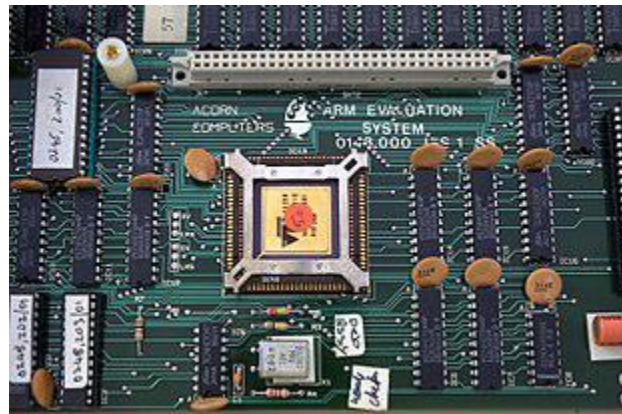



Fig. 2 ARM1 processor

5.2 ARM2

The result of the simulations on the ARM1 boards led to the late 1986 introduction of the ARM2 design running at 8 MHz, and the early 1987 speed-bumped version at 10 to 12 MHz.[b] A significant change in the underlying architecture was the addition of a Booth multiplier, whereas previously multiplication had to be carried out in software.[37] Additionally, a new Fast Interrupt reQuest mode, FIQ for short, allowed registers 8 through 14 to be replaced as part of the interrupt itself. This meant FIQ requests did not have to save out their registers, further speeding interrupts. The ARM2 was roughly seven times the performance of a typical 7 MHz 68000-based system like the Commodore Amiga or Macintosh SE. It was twice as fast as a Intel 80386 running at 16 MHz, and about the same speed as a multi-processor VAX-11/784 supermini. The only systems that beat it were the Sun SPARC and MIPS R2000 RISC-based workstations.[39] Further, as the CPU was designed for high-speed I/O, it dispensed with many of the support chips seen in these machines, notably, it lacked any dedicated direct memory access (DMA) controller which was often found on workstations. The graphics system was also simplified based on the same set of underlying assumptions about memory and timing. The result was a dramatically simplified design, offering performance on par with expensive workstations but at a price point similar to contemporary desktops. The

module that is compatible with wide range of devices including smartphone, laptops and tablets. Adding a Bluetooth to Arduino can take your project to the next level. It opens up lots of possibilities for user interface (UI) and communication.

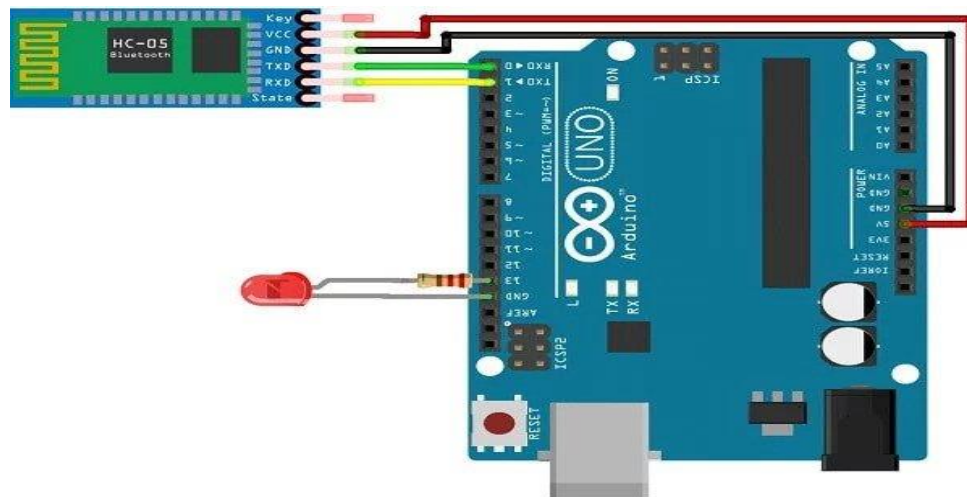


Fig. 5 Arduino with Bluetooth Connection

There are three main parts to this module. An Android smartphone, a Bluetooth transceiver, and an Arduino. HC 05/06 works on serial communication. The Android app is designed to send serial data to the Arduino Bluetooth module when a button is pressed on the app. The Arduino Bluetooth module at the other end receives the data and sends it to the Arduino through the TX pin of the Bluetooth module (connected to RX pin of Arduino). The code uploaded to the Arduino checks the received data and compares it. If the received data is 1, the LED turns ON. The LED turns OFF when the received data is 0. You can open the serial monitor and watch the received data while connecting.

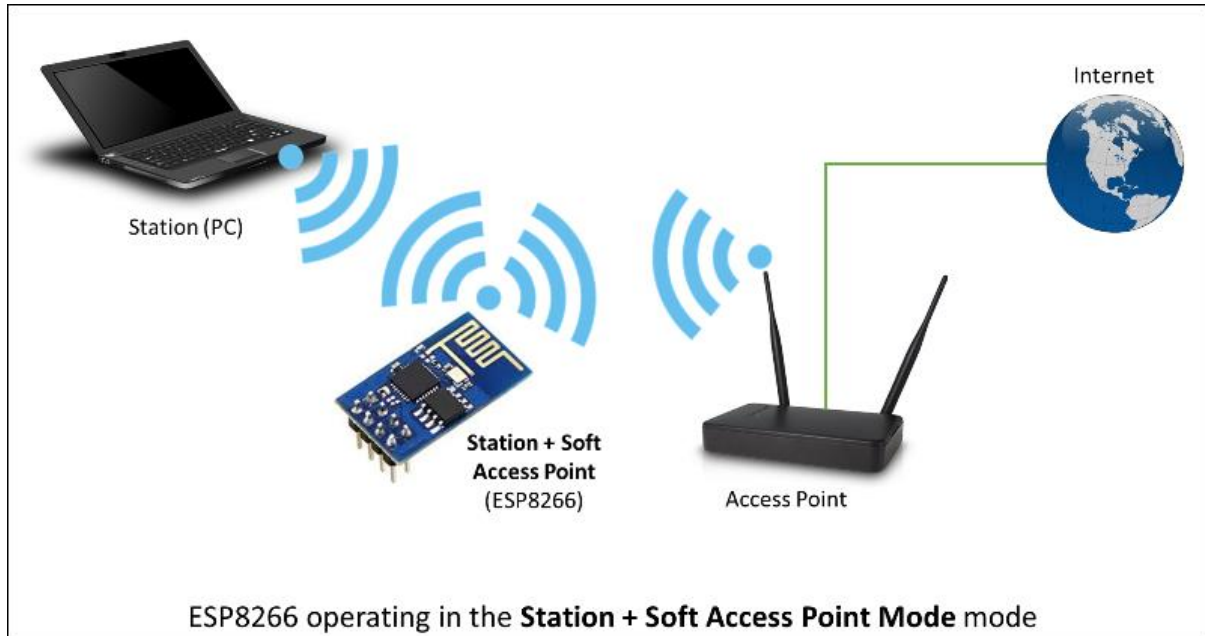


Fig. 6 ESP8266 Module

5. I2C and SPI

UART, I2C and SPI are one of the most common and basic hardware communication peripherals that makers and electricians use in microcontroller development. Similarly, for the Arduino, they contain UART, I2C and SPI peripheral too. The Arduino Uno Rev 3 is a microcontroller board based on the ATmega328, an 8-bit microcontroller with 32KB of Flash memory and 2KB of RAM.

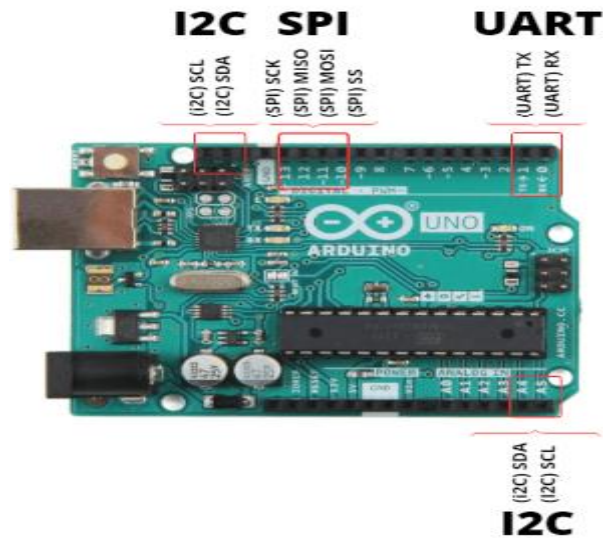


Fig.7 Arduino with I2C and SPI

- It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button

5.1 UART

UART stands for Universal Asynchronous Reception and Transmission and is a simple communication protocol that allows the Arduino to communicate with serial devices. The UART system communicates with digital pin 0 (RX), digital pin 1 (TX), and with another computer via the USB port. This peripheral, found on all Arduino boards, allows the Arduino to directly communicate with a computer thanks to the fact that the Arduino has an onboard USB-to-Serial converter. Therefore, programs written on a Windows, Mac, or Linux OS can be used with an Arduino connected to a USB port as if it was a serial port (serial port communication is trivial compared to USB communication). UART, which stands for Universal Asynchronous Reception and Transmission, is a simple serial communication protocol that allows the host (Arduino) to communicate with serial devices. UART supports bidirectional, asynchronous and serial data transmission. It uses 2 data lines to communicate with each other which are: TX (Pin 1) and RX (Pin 0).

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can easily interfaced with any microcontroller such as Arduino, Raspberry Pi, etc... to measure humidity and temperature instantaneously. DHT11 humidity and temperature sensor are available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor.

The working of the DHT sensor is pretty simple. DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture-holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with an increase in temperature. To get a larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

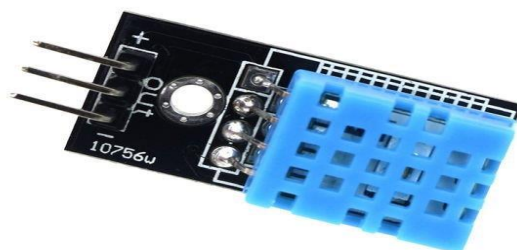


Fig. 8 DHT11 Sensor

The temperature range of DHT11 is from 0 to 50 degrees Celsius with a 2-degree accuracy. The humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this

In this experiment, we will connect the power and ground pins directly to the Arduino 5V and GND pins. The PWM input will be connected to one of the Arduino's digital output pins.

Experiment 1

Hardware Required

1 x TowerPro SG90 servo motor

1 x Arduino Mega2560

3 x jumper wires

Wiring Diagram

The best thing about a servo motor is that it can be connected directly to an Arduino. Connect to the motor to the Arduino as shown in the table below:

Servo red wire – 5V pin Arduino

Servo brown wire – Ground pin Arduino

Servo yellow wire – PWM(9) pin Arduino

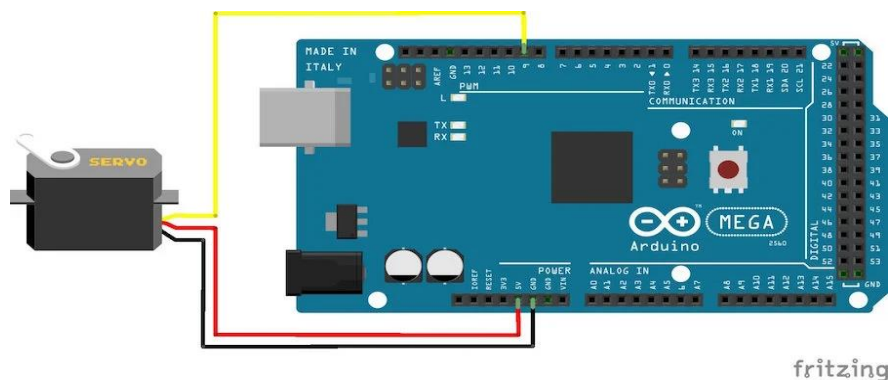


Fig. 11 Arduino with Servo motor

1.1 NodeMCU Specifications

The NodeMCU is available in various package styles. Common to all the designs is the base ESP8266 core. Designs based on the architecture have maintained the standard 30-pin layout. Some designs use the more common narrow (0.9") footprint, while others use a wide (1.1") footprint – an important consideration to be aware of.

The most common models of the NodeMCU are the Amica (based on the standard narrow pin-spacing) and the LoLin which has the wider pin spacing and larger board. The open-source design of the base ESP8266 enables the market to design new variants of the NodeMCU continually.

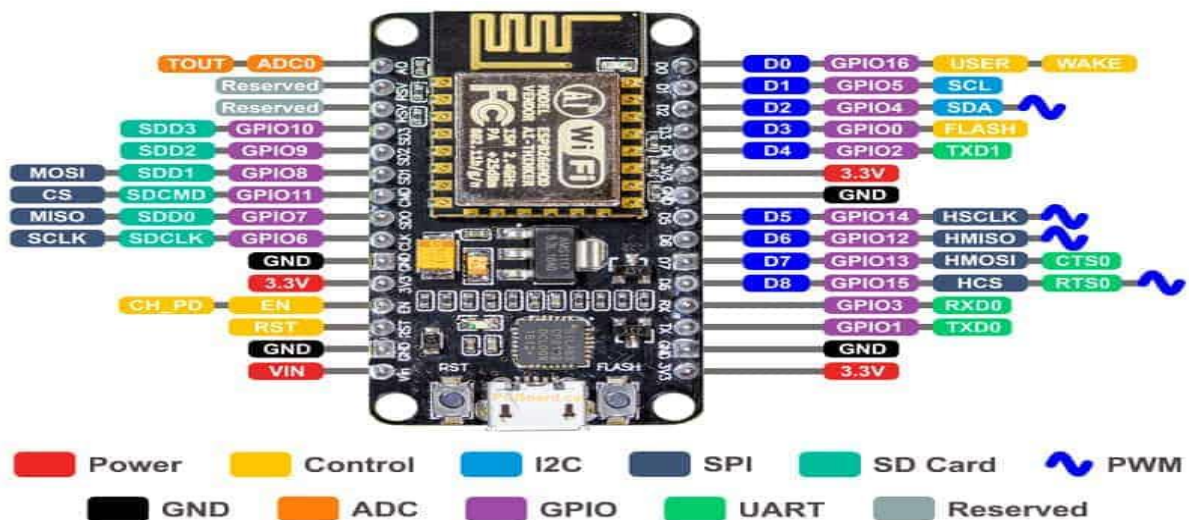


Fig. 1 ESP 8266 Components

- Power Pins There are four power pins. VIN pin and three 3.3V pins.
- VIN can be used to directly supply the NodeMCU/ESP8266 and its peripherals. Power delivered on VIN is regulated through the onboard regulator on the NodeMCU module – you can also supply 5V regulated to the VIN pin

Fig. 3 Installation of Libraries

After the installation select the ESP-01 board by going to: Tools → Board: → Generic ESP8266 Module

As known the Arduino UNO board contains Microchip ATmega16U2 microcontroller which is used as USB-to-serial converter. This chip (ATmega16U2) can be used to program (flash) the ESP-01 Wi-Fi module, circuit connections are shown below:

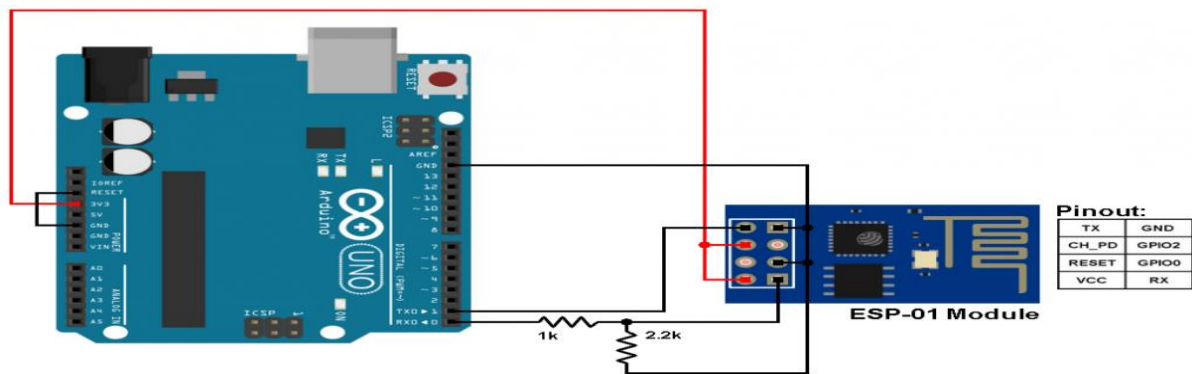


Fig. 4 ESP8266 (ESP-01) Module with Arduino UNO board

In the circuit there are 2 resistor one of 1k ohm and the other one of 2.2k ohm. The two resistors are used to step down the 5V which comes to arduino into about 3.43V which goes to the ESP-01 board (connected to RX pin of the ESP-01) because the ESP8266EX chip works with 3.3V only and applying a 5V directly may damage it.

On the other hand, the TX pin of the ESP-01 is connected directly to the Arduino board without any voltage level converter because here the ESP-01 sends data (at 3.3V) to the Arduino board using this pin. This is a simple example which we should start with, it's the LED blinking example. In this example I'm going to connect one LED to ESP-01 board GPIO2 pin, and make this LED blinks. Circuit diagram is shown below:

temperature from DHT11 sensor and Process it and give it to a ESP8266 Module.ESP8266 is a WiFi module, it is one of the leading platform for Internet of Things. It can transfer a data to IOT cloud.

Hardware Requirements

- Arduino Uno
- ESP8266-01
- DHT11
- AMS1117-3.3V
- 9V battery

Software Requirements

- Arduino IDE

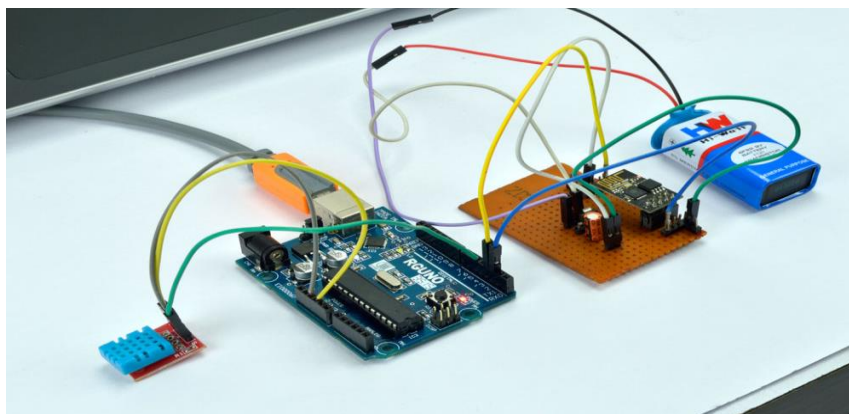


Fig. 5 Arduino with DHT sensor

The 2nd pin is of DHT11 is a data pin, it can send a temperature and humidity value to the 5th pin of Arduino Uno.1st and 4th pin of DHT11 is a Vcc and Gnd and 3rd pin is no connection. The Arduino Uno process a temperature and humidity value and send it to a ESP8266 WiFi module. The Tx and Rx pin of ESP8266 is connected to the 2nd (Rx) and 3rd (Tx) of Arduino

Uno. Make sure that input voltage of ESP8266 must be 3.3V, not a 5V (otherwise it would damage a device). For that, we are using AMS1117 Voltage regulator circuit. It can regulate a voltage from 9V to 3.3V and will give it to Vcc pin of ESP8266. The Ch_Pd is a chip enable pin of ESP8266 and should be pullup to 3.3V through 3.3K Ω resistor. For reset the module pull down the RST pin of ESP8266 to Gnd. ESP8266 have 2 GPIO pins GPIO 0 and GPIO 2.

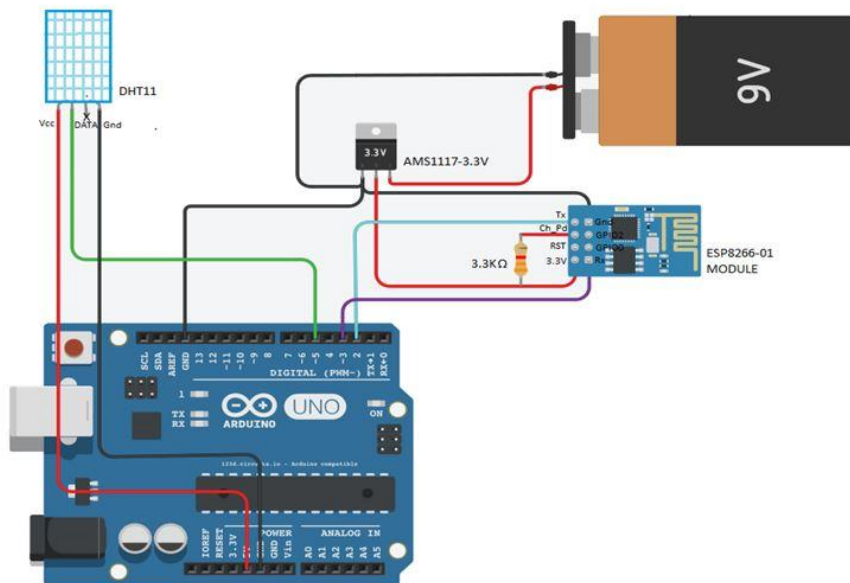


Fig. 6 Circuit diagram for monitoring Humidity and Temperature in IOT cloud

ThingSpeak is an open source platform to store and retrieve a data for Internet of Things application. To use this, you need to register in ThingSpeak cloud and then login to your account. After create a new channel with temperature in one field and humidity in another field as shown in Fig: 1.2. Once you created a new channel, it will generate a two API keys, they are READ API keys and WRITE API keys. First, copy the WRITE API keys from ThingsSpeak and paste it into the line (`String apiKey = "OX9T8Y9OL9HD0UBP";`) of the program. Next, replace the Host_Name and Password with your WiFi name and WiFi password in the two lines given below in the program. (`String Host_Name = "Pantech"` and `String Password = "pantech123"`). The Arduino program Uses DHT library, if it is not presented in your arduino IDE, select Sketch à Include library à Manage libraries à Install DHT Sensor library. Then

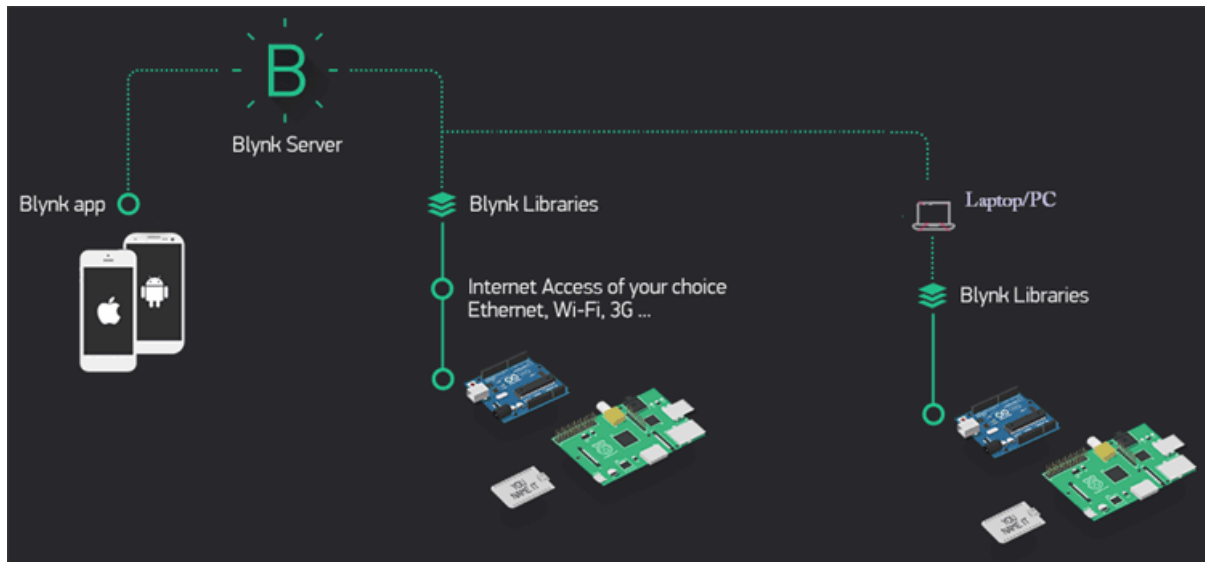


Fig. 10 Blynk Application

6.1 Features of Blynk

1. You can add a notifications service using the Blynk app without using any third-party platform like IFTTT. For example, you can post the data on Twitter and get the e-mail when something reaches its threshold. This can be possible just by configuring the Blynk app.
2. In IoT projects, the hardware part is easy as compare to the software part. But using Blynk, the software part also becomes easier than the hardware. There is very less coding required and all the code is included in its library. Blynk is perfect for building simple projects.
3. Most of the microcontroller available in the market is supported by Blynk and these microcontrollers can be controlled using Blynk app via Wi-fi, BLE, USB, GSM, and Ethernet.
4. You can create your own local **Blynk server** to control the appliances locally just by using few steps and can control easily using the Blynk app.
5. One of the most interesting features of Blynk is the use of virtual pins. Virtual Pin is a concept invented by Blynk to provide the exchange of any data between hardware and the Blynk mobile app. These pins are different from Digital and Analog pins, they don't have any physical properties.

So, if you want any data from the virtual pin, the Blynk app will send the data to a defined virtual pin and then this data can be accessed on MCU pins. Also, the data can be sent from the Blynk app to any virtual pin, and then the data can be easily accessed on the app.

6.2 Circuit Diagram

The circuit diagram is very simple, just connect one LED to PWM pin (5) and the other LED to digital pin 4 of Arduino Uno.

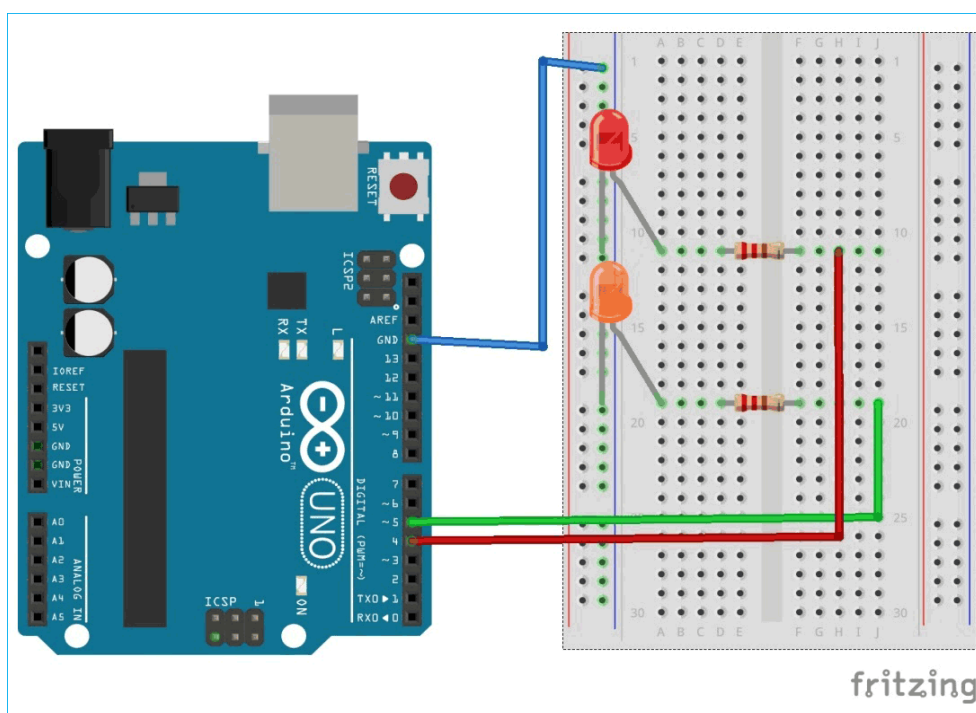
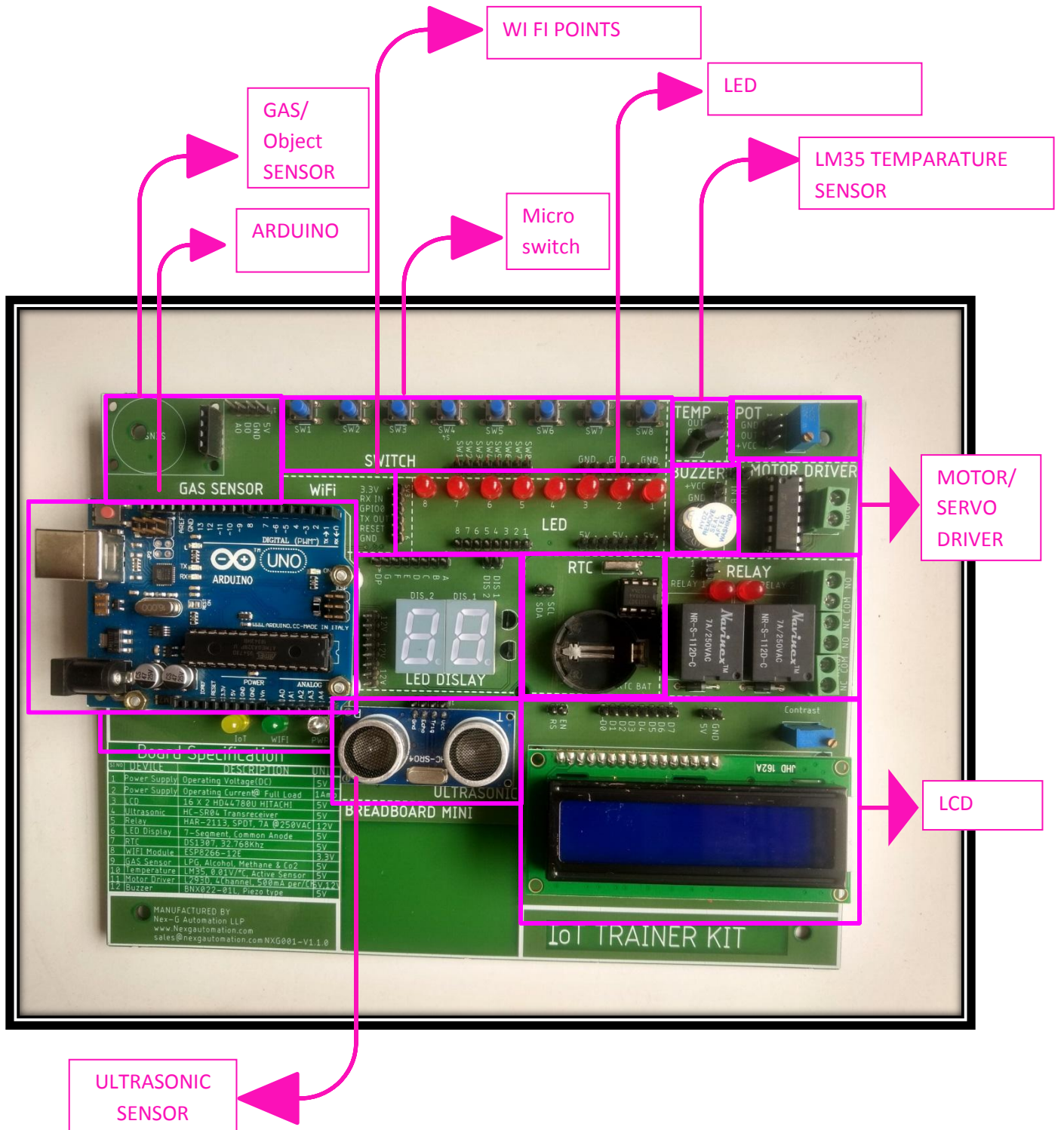


Fig. 11 Arduino with LED

6.3 Installing and Configuring Blynk App to control LED

1. Download the [Blynk app](#) from the play store. It is available for both Android and iOS users. Open the app and create an account by entering your e-mail ID and password.
2. Now, we will create a New Project. So, tap on New Project.

IoT Trainer Kit Description:



1. LED Interfacing:

Below image display in detailed connection diagram, in which it have 8 No. of LED and its input pins.

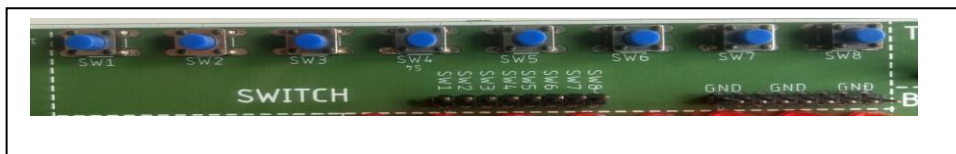
Note: All LED's are Active HIGH i.e. Common Anode.



2. Switch Interfacing:

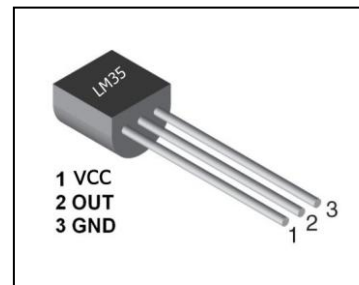
In this image it shows 8 Buttons and its respective Pin out. All Button are of Logic LOW output.

Note: If you press the BUTTON/SWITCH output: LOW
If you Release the BUTTON/SWITCH output: HIGH.



3. Temperature Sensor LM35

LM35 is a temperature measuring device having an analog output voltage proportional to the temperature. It provides output voltage in Centigrade (Celsius). It does not require any external calibration circuitry. The sensitivity of **LM35** is 10 mV/degree Celsius.



4. Gas Sensor/ IR(Object) Sensor:

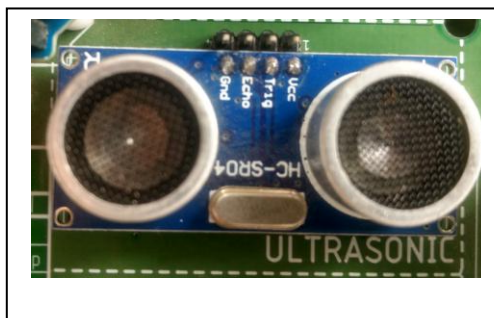
An infrared sensor is an electronic device that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it that is called a passive IR sensor. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation.



Connection	
5V	5 V Supply
GND	GND
DO	UNO pin 2-13
AO	UNO Analog pin A0—A5

5. Ultra Sonic Sensor:

As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.



Connection	
Vcc	5 V Supply
GND	GND
Trig	UNO pin 2-13 as OUTPUT
Echo	UNO pin 2-13 as INPUT

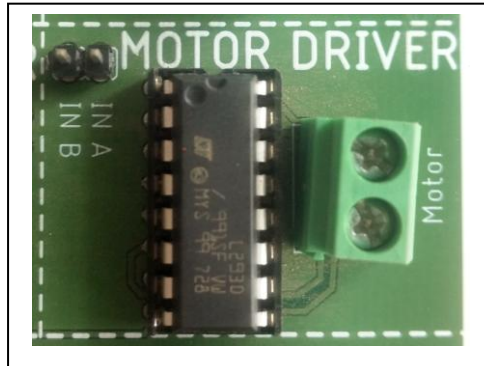
6. Motor Driver/Servo Motor

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC.

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor.

Servo motor Connection:

BROWN: Ground **RED:** Vcc(+5V) **YELLOW:** OUTPUT



7. Buzzer

A **buzzer** or **beeper** is an audio signaling device which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Connection: +VCC → Input to buzzer GND → Ground



8. Liquid Crystal Display(LCD) 16x2:

The 16×2 liquid crystal display can be used in embedded systems where displaying a limited amount of data is necessary. This display comes with two lines of data and each line is divided into sixteen columns.

Connection:

+VCC → +5V
GND → Ground
RS → pin 2 to 13
EN → pin 2 to 13
D0 – D3 → Not used
D4 → pin 2 to 13
D5 → pin 2 to 13
D6 → pin 2 to 13
D7 → pin 2 to 13



Contrast Adjustment

Possible Experiments:

1. Interfacing 16x2 LCD with Hello Word Display
2. Blinking LED and Pattern generation with LED'S
3. Interfacing Switch for changing the Pattern of LED's
4. Interfacing IR Sensor as object detector.
5. Interfacing LM35 Active temperature sensor.
6. Interfacing ultrasonic sensor for distance measurements.
7. Interfacing Servo motor for Forward and Reverse rotation.
8. LM 35 data logger to the thing speak Server.

Package Includes:

1. IoT trainer kit with Wooden box.
2. USB type-A to type-B cable (for Firmware upload to Arduino UNO board).
3. Male to male and male to female Burg connecting wires.
4. Power adaptor 12V, 2Amps.
5. CD (contains all Source codes for the possible experiments)