Lab manual – Electronic Tinkering Workshop

Lab manual – Electronic Tinkering Workshop

Department of Electronics & Communication Engineering

GOVERNMENT POLYTECHNIC COLLEGE, PERUMBAVOOR

Koovappady P O, Perumbavoor, Ernakulam – 683544

www.gptcperumbavoor.ac.in

his document is compiled by

Dr. Aiju Thomas
Principal
Government Polytechnic College
Perumbavoor
www.gptcperumbavoor.ac.in

Index

General Remarks: (For office use only)							
Test 1:		Test 2:		Assign 1:		Assign 2	

VISION AND MISSION

Government Polytechnic College, Perumbavoor Vision and Mission

Vision

Excel as a centre of skill education moulding professionals who sincerely strive for the betterment of society.

Mission

- To impart state of the art knowledge and skill to the graduate and moulding them to be competent, committed and responsible for the well being of society.
- To apply technology in the traditional skills, thereby enhancing the living standard of the community

Department of Electronics & Communication Engineering

Vision

Excel as a centre of skill education in Electronics and Communication Engineering, moulding professionals who sincerely strive for the betterment of themselves and society.

Mission

- To impart state of the art knowledge, skill and attitude to the students and contributing to their sustainable development.
- To merge technologies in the field of Electronics and Communication Engineering with occupational skills, thereby improving quality of living.

Exp No. 1 Date: D D - M M - Y Y

SAFETY PROCEDURES

Problem Statement:

The safety instructions are presented to the attention of the students as a mean of preventing accidents while performing experiments and activities in the communication lab of the department. The purpose is to draw attention to the risks involved in lab activities to prevent human suffering and damage to equipment.

Safety in the laboratory:

Working in the lab is not allowed without following electricity precautions displayed.

No individual work is allowed in the lab.

Laboratory in charge is responsible for the arrangements of your lab activities; Listen carefully to his/her instructions and follow them.

To do and not to do:

Inform the lab in charge about dangerous conditions and faults in the lab or nearby environment.

Do not do any action that may harm people or equipment in the lab.

Do not misuse any of the tools or instruments belong to the lab.

Strict discipline should be maintained in the laboratory.

Turn off cell phones before entering the lab.

At the end and beginning of laboratory, follow 5S procedures and leave the work table clean and tidy.

Electrical Safety:

Consult Electrical Engineering section available in the campus for electrical safety queries.

The lab equipment is powered from electrical sockets installed on the tables. Do not use equipment that is powered from a damaged socket.

Do not use equipment that is powered from flexible cable with damaged insulation or if it’s plug is not assembled properly.

Do not repair or disassemble electrical equipment including replacement of fuses installed in the equipment.

Do not open the main fuse box, unless it is an emergency and you need to switch off main circuit breaker.

Emergency Switches:

The laboratory has circuit breakers, which is located in the main panel. Identify the place.
In an emergency condition, switch off circuit breakers immediately.

Result

Familiarization of safety precautions performed.

	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 2 Date: D D - M M - Y Y

HANDLING ELECTROSTATIC DISCHARGE (ESD)

Problem Statement:

Familiarize ESD handling procedures in the laboratory

Theory

In handling electronic devices, datasheets caution about ESD (Electrostatic Discharge) precautions. These devices are prone to damage because of electrostatic charges made by human body. These charges may be up to 4000 volts and cause damage without being noticed. It is recommended to follow ESD precautions on handling of these devices.

Points for the elimination of ESD damage to electronic components

1. Make sure you have a reliable ground point available near the table.
2. Do not wear clothing which generates static electric charges every time you move.
3. Do not handle static generating objects while working on electronics.
4. Store all chips and other components in appropriate anti-static containers.
5. Keep all ESD sensitive components and spares in anti-static envelopes for storage.
6. Be sure to turn off the power and remove the power plug from all equipment before working repairing or assembling.
7. Do not plug in or remove equipment while the power is on.

Result

Familiarization of ESD protection procedures performed.

	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 3 Date: D D - M M - Y Y

IDENTIFICATION OF TOOLS FOR THROUGH HOLE SOLDERING

Problem Statement : Identify tools and consumables used for soldering and de-soldering of “through hole” PCBs – soldering iron of different wattage, temperature-controlled soldering station, soldering iron stand, fume extractor, solder of various grade, flux, nipper, wire stripper, needle node plier, tweezer, de-soldering pump, de-soldering station, single layer, multi-layer, through hole and SMD PCBs, etc.

Soldering Iron

Ensure that proper wattage of soldering iron are selected for the work. For safety of components against leakage current, make sure that the soldering iron have an earth point and the socket do have proper earthing. A 25W “Soldren” Soldering iron is a good and non-expensive choice for hobbyists.

A 25W Soldron Soldering iron for PCB

A 50W Soldron Soldering iron for terminals or solder points with more surface area

A 100W Soldron Soldering iron for terminals including battery

Temperature-controlled soldering station for sensitive components

Soldering Iron stand

Ensure that you have good quality soldering iron stand with sufficient weight to prevent accident falling. Every time you prepare for soldering, make sure that the sponge pad is soaked in water.

Fume extractor

Long term breathing of solder fume is not good for health. Use proper fume extractor for soldering.

Solder of various grades

DIFFERENT TYPES OF SOLDER WIRE				
Grades of solder wire	**(Sn) Tin**	**(Pb) Lead**	**(Cu) Copper**	**Melting Range (°c)**
Solder wire (60/40)	60	40	-	183-189
Solder wire (63/37)	63	37	-	183-183
Solder wire (50/50)	50	50	-	183-214
Solder wire (40/60)	40	60	-	183-238
Solder wire (30/70)	30	70	-	183-258
Lead free Solder wire	100	-	-	
Solder wire(99.3/0.7)	99.3	-	0.7	227
Lead Wire	100	-	-	327
Solder wire(3/97)	3	97	-	326

Flux

Flux is a chemical cleaning agent used before and during the soldering process of electronic components onto circuit boards. Flux is used in both manual hand soldering as well as the different automated processes used by PCB contract manufacturers. The main purpose of the flux is to prepare the metal surfaces for soldering by cleaning and removing any oxides and impurities. Oxides are formed when metal is exposed to air and may prevent the formation of good solder joints.

Liquid flux

Flux Paste (Soldering paste)

Nipper

Wire Stripper

Needle Nose Piler

Tweezer

De soldering Pump

De soldering Wick

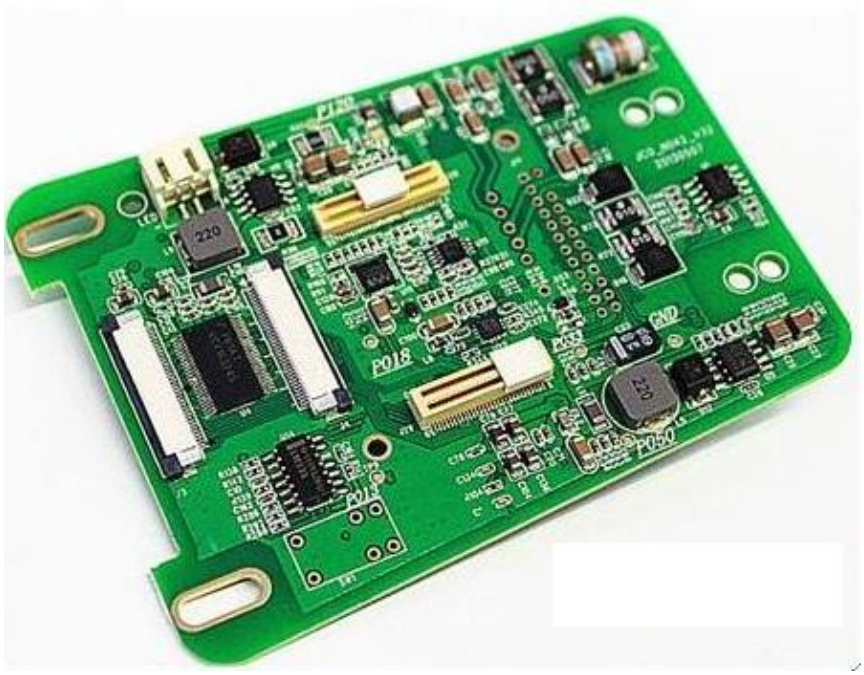
De soldering Station for through hole

SMD re workstation

Printed Circuit boards (PCB)

An assembled single sided PCB

An assembled double-sided PCB



Result

Tools and consumables used for soldering and de-soldering of “through hole” PCBs identified.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 4 Date: D D - M M - Y Y

Familiarize Manual SMD Work and Rework

Problem Statement : Identify tools used to perform soldering and de-soldering of SMD PCBs – soldering station, electronic rework station etc.

Courtesy: https://www.gadgetronicx.com/smd-soldering-tools-techniques/

SMT (Surface Mount Technology) or SMD (Surface mount devices) has become popular due to several reasons. First of all its cost efficient and requires less space as compared to Through Hole Components. As the name suggests SMD is mounted directly on the surface of either top or bottom side of PCB, while THT components are inserted into PCB. SMD components saves a lot of mounting areas in the PCB. Also, the SMD components are relatively smaller in size which reduces the overall dimension and complexity of circuit board. Many hobbyists and makers consider SMD soldering to be daunting but it's far from reality. All it needs is right tool and some practice.

Package Size

Code		Length (l)		Width (w)		Height (h)		Power
Imperial	Metric	inch	mm	inch	mm	inch	mm	Watt
0201	0603	0.024	0.6	0.012	0.3	0.01	0.25	1/20 (0.05)
0402	1005	0.04	1.0	0.02	0.5	0.014	0.35	1/16 (0.062)
0603	1608	0.06	1.55	0.03	0.85	0.018	0.45	1/10 (0.10)
0805	2012	0.08	2.0	0.05	1.2	0.018	0.45	1/8 (0.125)
1206	3216	0.12	3.2	0.06	1.6	0.022	0.55	1/4 (0.25)
1210	3225	0.12	3.2	0.10	2.5	0.022	0.55	1/2 (0.50)
1218	3246	0.12	3.2	0.18	4.6	0.022	0.55	1
2010	5025	0.20	5.0	0.10	2.5	0.024	0.6	3/4 (0.75)
2512	6332	0.25	6.3	0.12	3.2	0.024	0.6	1

Size is a very impotent factor when it comes to SMD components. Passive components like resistors, capacitors, diodes come in various package sizes such as 1206, 0805, 0603 etc. These numbers denote the actual size of that component.

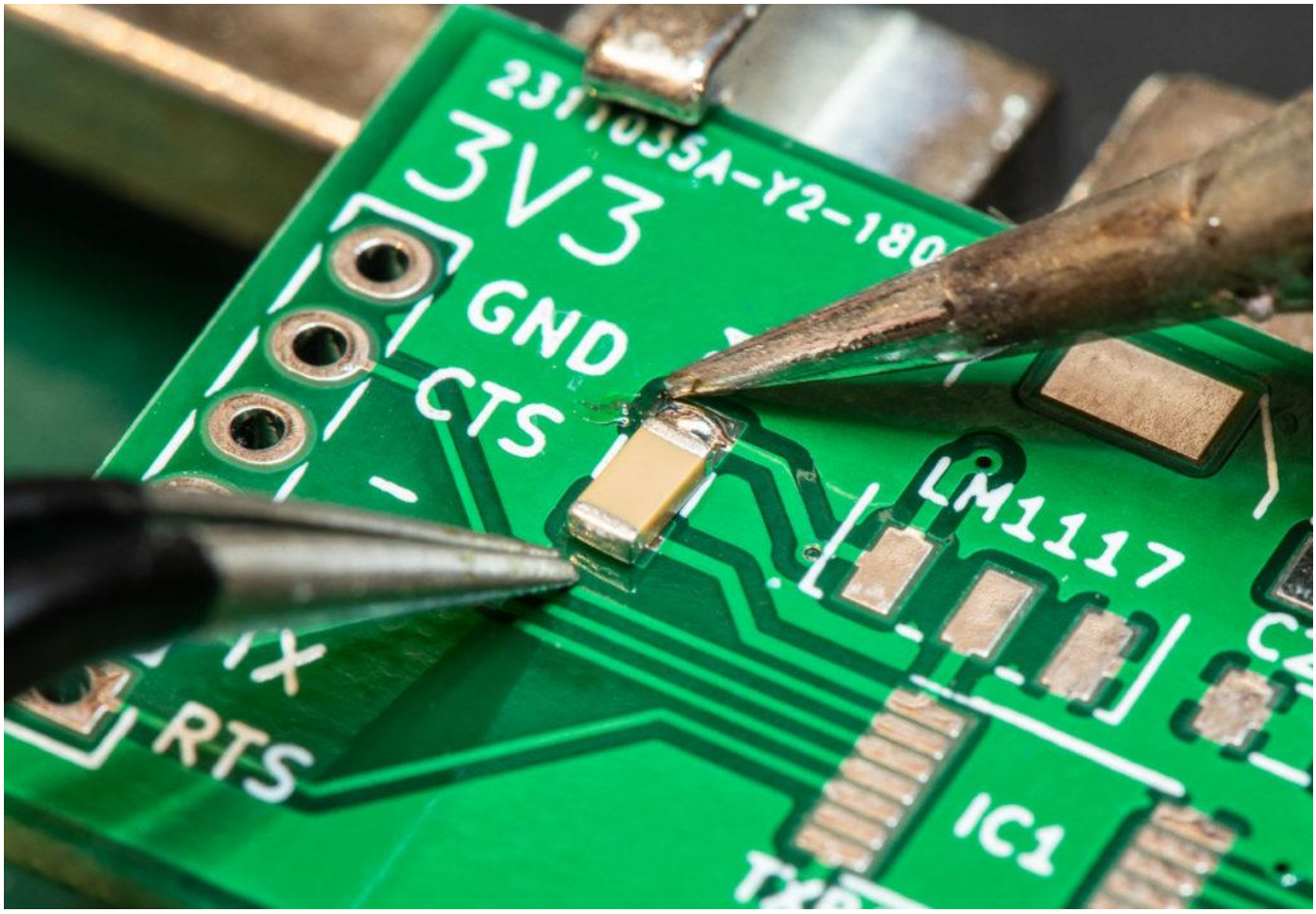
Other components such as Transistors and ICs also have different packages. For general purpose SMD transistors, the SOT23 (Small Outline Transistor) SMT package is mostly used. The SOT23 generally has three terminals of transistor, two of them on one side and third pin is on other side. But it can have more pins depending on the nature of IC, it is being used for. For example, small integrated circuits such as an operational amplifier, etc.

SMD Soldering techniques:

Soldering SMD components might seem difficult at first, but it's not that hard once you know the proper technique and have the right tool to do it. There exist different techniques for SMD soldering. Out of which these three types are most commonly used by the makers.

1. Hand Soldering
2. Hot Air Soldering

Hand Soldering:



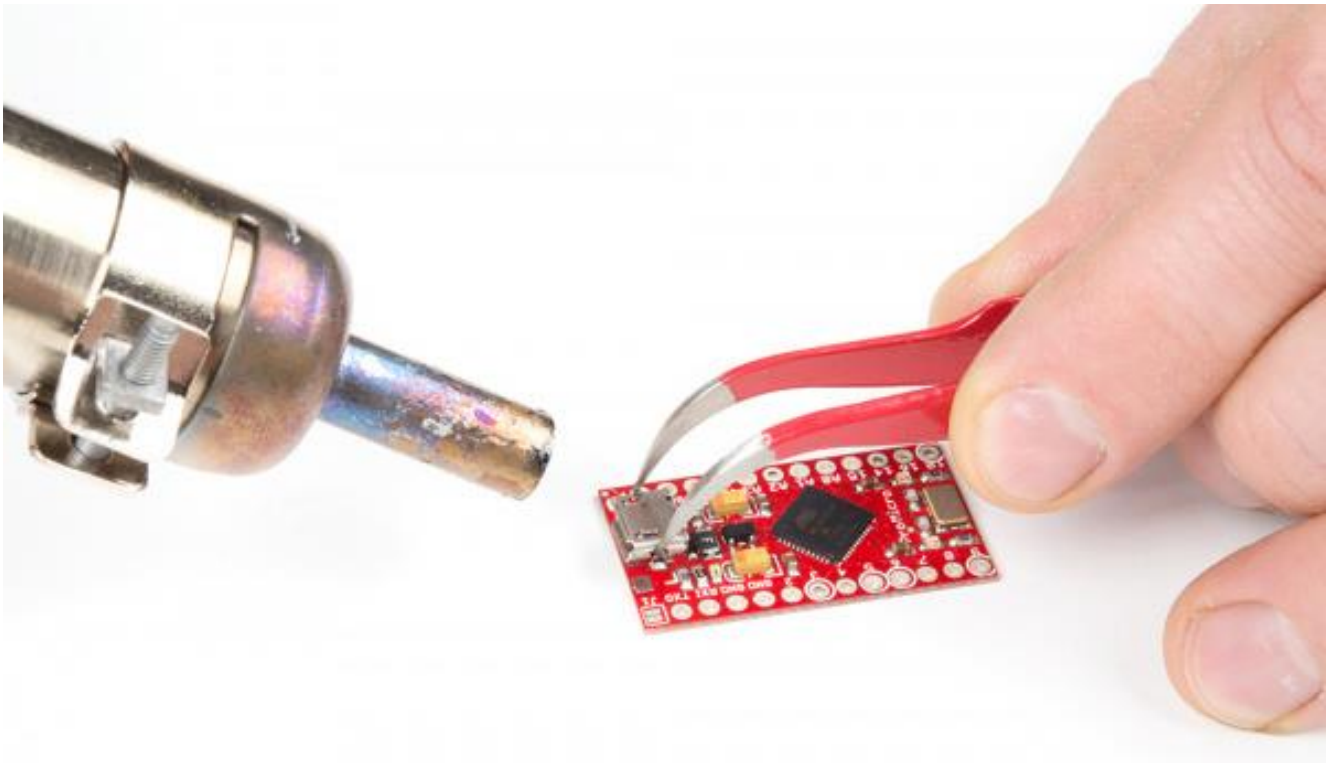
Hand Soldering is the most traditional way we solder the SMD components. This is done using a Soldering gun. This might be difficult at the beginning but after a few days of practice you will get used to it. You will need a very fine tip soldering iron (preferably one with temperature control) to get better precision, a lot of flux and some patience.

Generally, you can solder 1206 to 0603 packages by hand without much of a problem. But if you go

smaller than that, then you may need a microscope or a magnifying glass because these are so tiny. Let's see step by step on how to solder SMD components.

1. First add some flux to PCB footprints. It will help to keep the solder in place.
2. Proceed to add a little bit of solder in one pad.
3. Use a tweezer to grab the SMT component and align it with the PCB footprint
4. Finally push the component gently to the pad while heating it by the soldering iron
5. For IC's and components with more than two to three pins solder the diagonal pins first, this will help you keep the IC in place
6. Solder rest of the pins afterwards

Hot air Soldering:



This soldering method is bit easier than using a Soldering iron. In this process instead of soldering iron, Hot Air station and Solder paste are used to mount the components. Here is step by step instructions to perform Hot air soldering.

1. First soldering paste is applied to the footprints.
2. A dedicated stencil makes this process easier but if you don't have any then use a solder paste pen or injection to apply on pads.
3. Place all the components one by one. Then set the temperature of the hot air station to around 300-350 degree C and expose the board to hot air.

4. When paste starts melting it will automatically suck the component into its position.
5. One important thing to remember is that do overheat any component as it may damage it permanently.
6. Check datasheet of the component in order to understand its temperature profile. In case of a LED apply heat from the underneath of the PCB.

Result

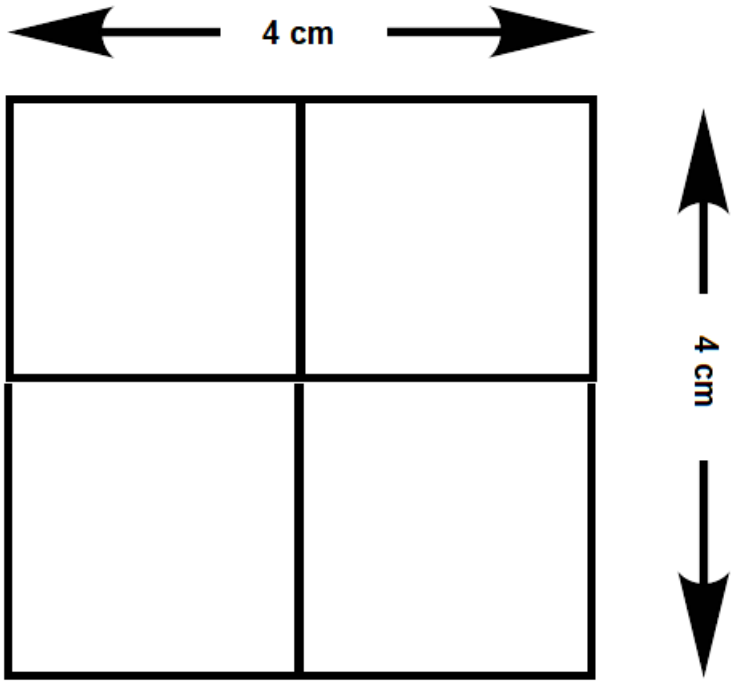
Identified tools used to perform manual soldering and de-soldering of SMD PCBs.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 5 Date: D D - M M - Y Y

Soldering Practice 1

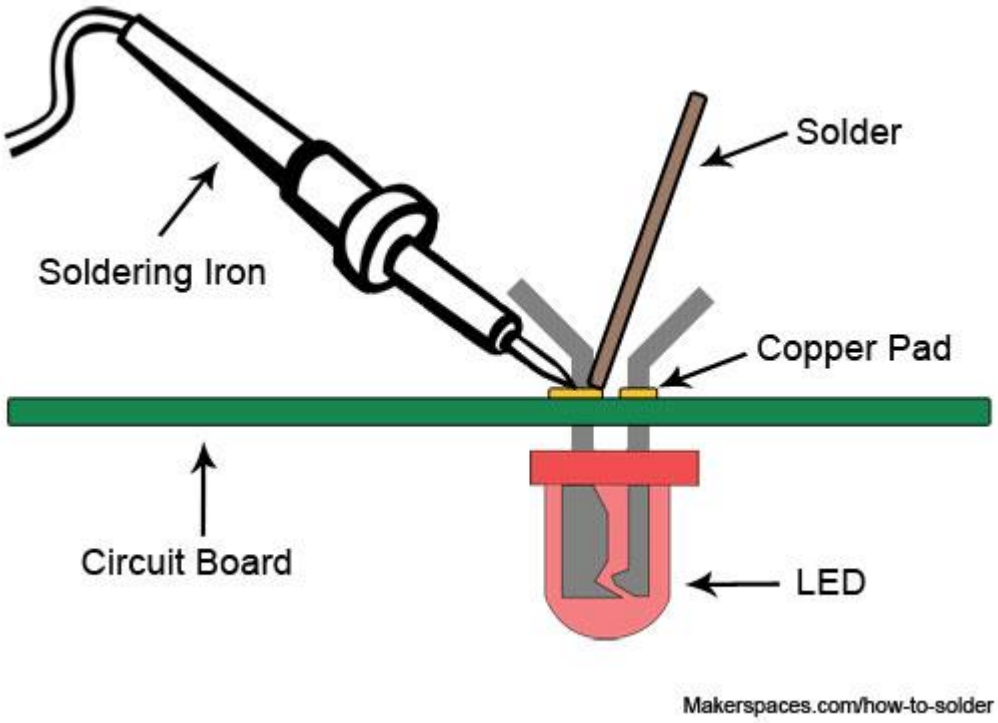
Problem Statement : Model 2x2 mesh by soldering single strand copper wire.



Courtesy: https://www.makerspaces.com/how-to-solder/

What Is Soldering?

If you were to take apart any electronic device that contains a circuit board, you'll see the components are attached using soldering techniques. Soldering is the process of joining two or more electronic parts together by melting solder around the connection. Solder is a metal alloy and when it cools it creates a strong electrical bond between the parts. Even though soldering can create a permanent connection, it can also be reversed using a de soldering.



Tinning The Tip

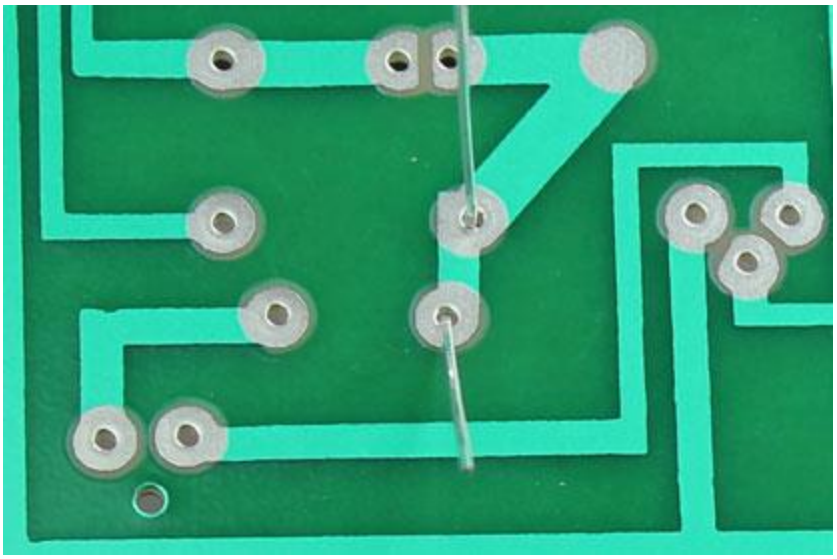
Before you can start soldering, you need to prep your soldering iron by tinning the tip with solder. This process will help improve the heat transfer from the iron to the item you're soldering. Tinning will also help to protect the tip and reduce wear.

Step 1: Begin by making sure the tip is attached to the iron and screwed tightly in place.
Step 2: Turn on your soldering iron and let it heat up. If you have a soldering station with an adjustable temp control, set it to 400′ C/ 752′ F.
Step 3: Wipe the tip of the soldering iron on a damp wet sponge to clean it. Wait a few seconds to let the tip heat up again before proceeding to step 4.
Step 4: Hold the soldering iron in one hand and solder in the other. Touch the solder to the tip of the iron and make sure the solder flows evenly around the tip.

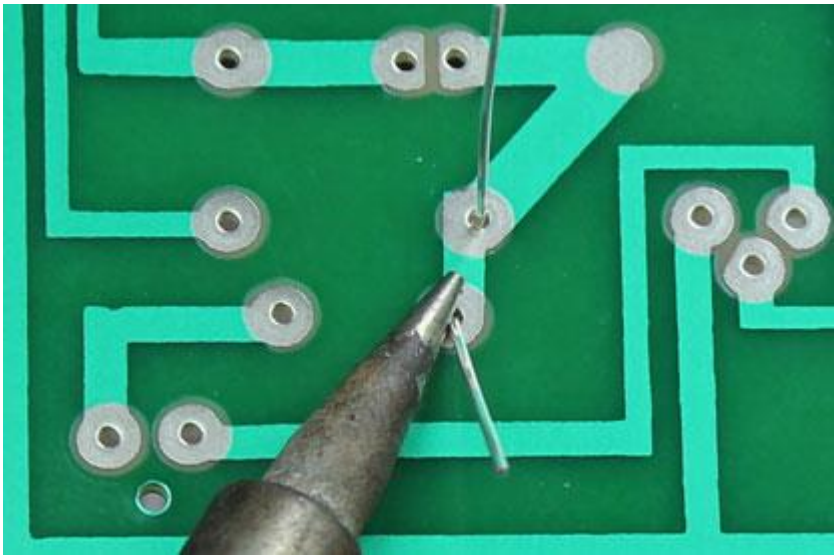
You should tin the tip of your iron before and after each soldering session to extend its life. Eventually, every tip will wear out and will need replacing when it becomes rough or pitted.

How to solder

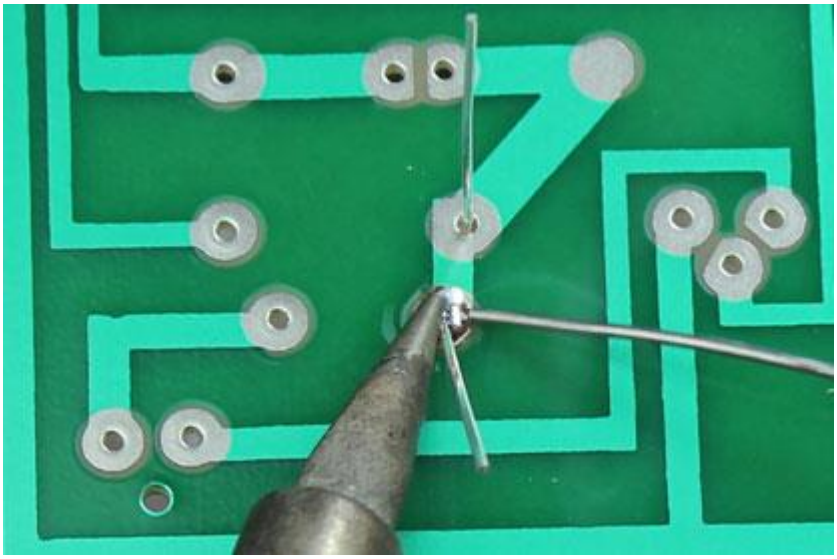
Step 1: Mount the Component – Begin by inserting the leads of the component into the holes of the circuit board. Flip the board over and bend the leads outward at a 45′ angle. This will help the component make a better connection with the copper pad and prevent it from falling out while soldering.



Step 2: Heat the Joint – Turn your soldering iron on and if it has an adjustable heat control, set it to 400’C. At this point, touch the tip of the iron to the copper pad and the resistor lead at the same time. You need to hold the soldering iron in place for 3-4 seconds in order to heat the pad and the lead.

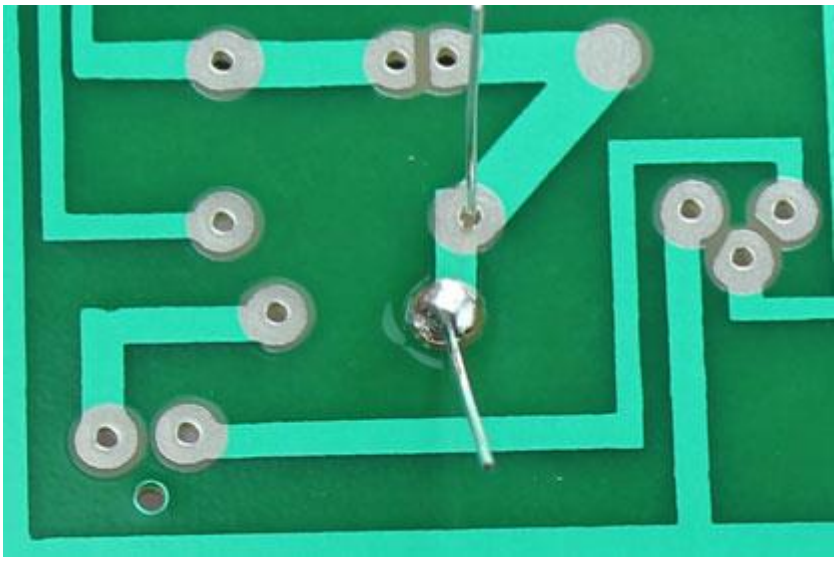


Step 3: Apply Solder to Joint – Continue holding the soldering iron on the copper pad and the lead and touch your solder to the joint. IMPORTANT – Don't touch the solder directly to the tip of the iron. You want the joint to be hot enough to melt the solder when it's touched. If the joint is too cold, it will form a bad connection.



Step 4: Snip the Leads – Remove the soldering iron and let the solder cool down naturally. Don't blow on the solder as this will cause a bad joint. Once cool, you can snip the extra wire from leads.

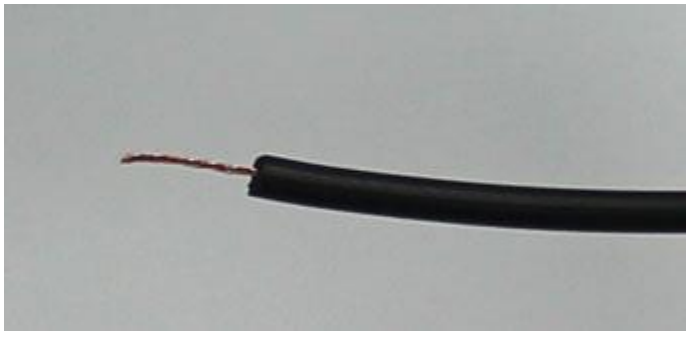
A proper solder joint is smooth, shiny and looks like a volcano or cone shape. You want just enough solder to cover the entire joint but not too much so it becomes a ball or spills to a nearby lead or joint.



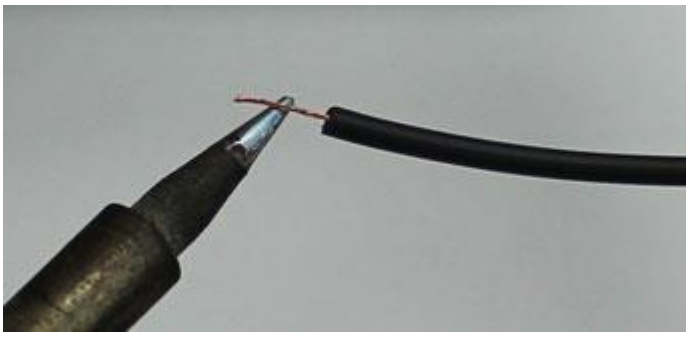
How To Solder Wires

Now it’s time to show you how to solder wires together. For this process, it’s recommended to use helping hands or other type of clamp device.

Begin by removing the insulation from the ends of both wires you are soldering together. If the wire is stranded, twist the strands together with your fingers.



Make sure your soldering iron is fully heated and touch the tip to the end of one of the wires. Hold it on the wire for 3-4 seconds.



Keep the iron in place and touch the solder to the wire until it’s fully coated. Repeat this process on the other wire.

Hold the two tinned wires on top of each other and touch the soldering iron to both wires. This process should melt the solder and coat both wires evenly.

Remove the soldering iron and wait a few seconds to let the soldered connection cool and harden. Use heat shrink to cover the connection.

Apply the method to prepare 2 x 2 mesh as shown

Result

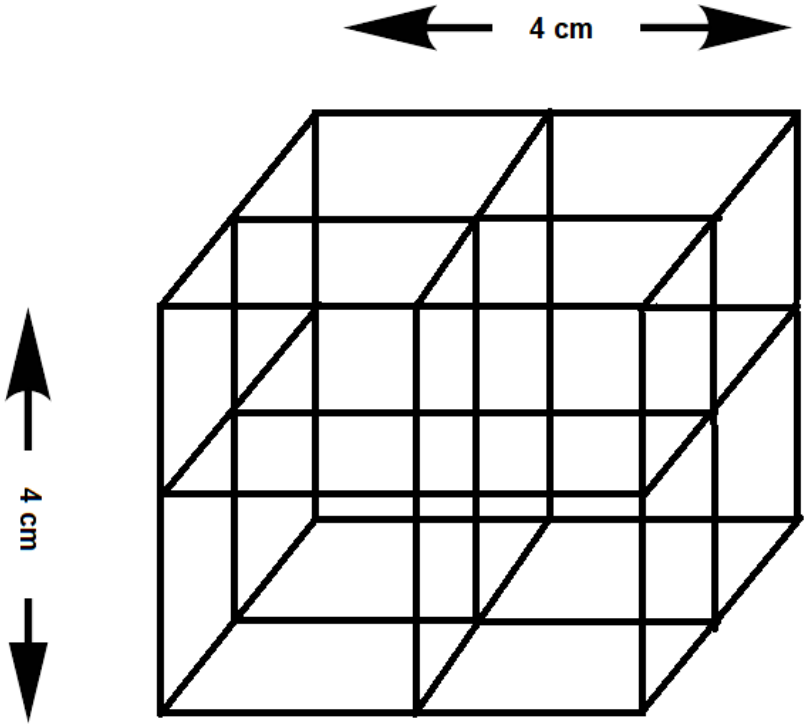
2x2 mesh prepared using soldering techniques

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 6 Date: D D - M M - Y Y

Soldering Practice 2

Problem Statement: Model 2x2x2 mesh polygonal cube by soldering single strand copper wire. (Perform tinning before soldering)



Apply the method explained in the previous experiment to prepare 2 x 2 x 2 cube as shown

Result

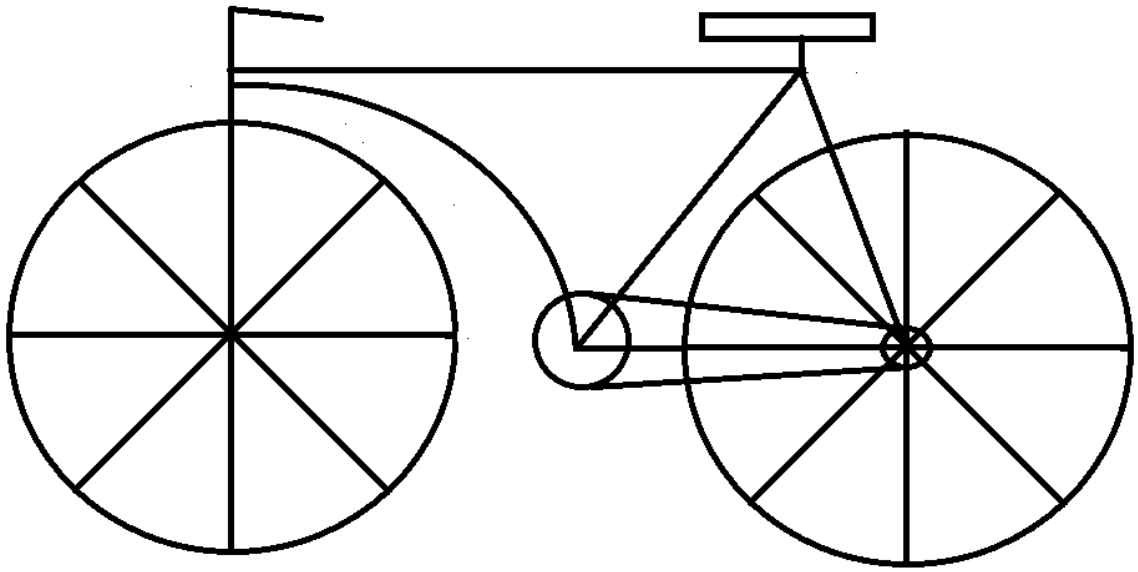
2x2x2 cube prepared using soldering techniques

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 7 Date: D D - M M - Y Y

Soldering Practice 3

Problem Statement: Prepare the following curio using soldering techniques



Apply the method explained in the previous experiment to prepare the curio.

Result

The mentioned curio prepared using soldering techniques

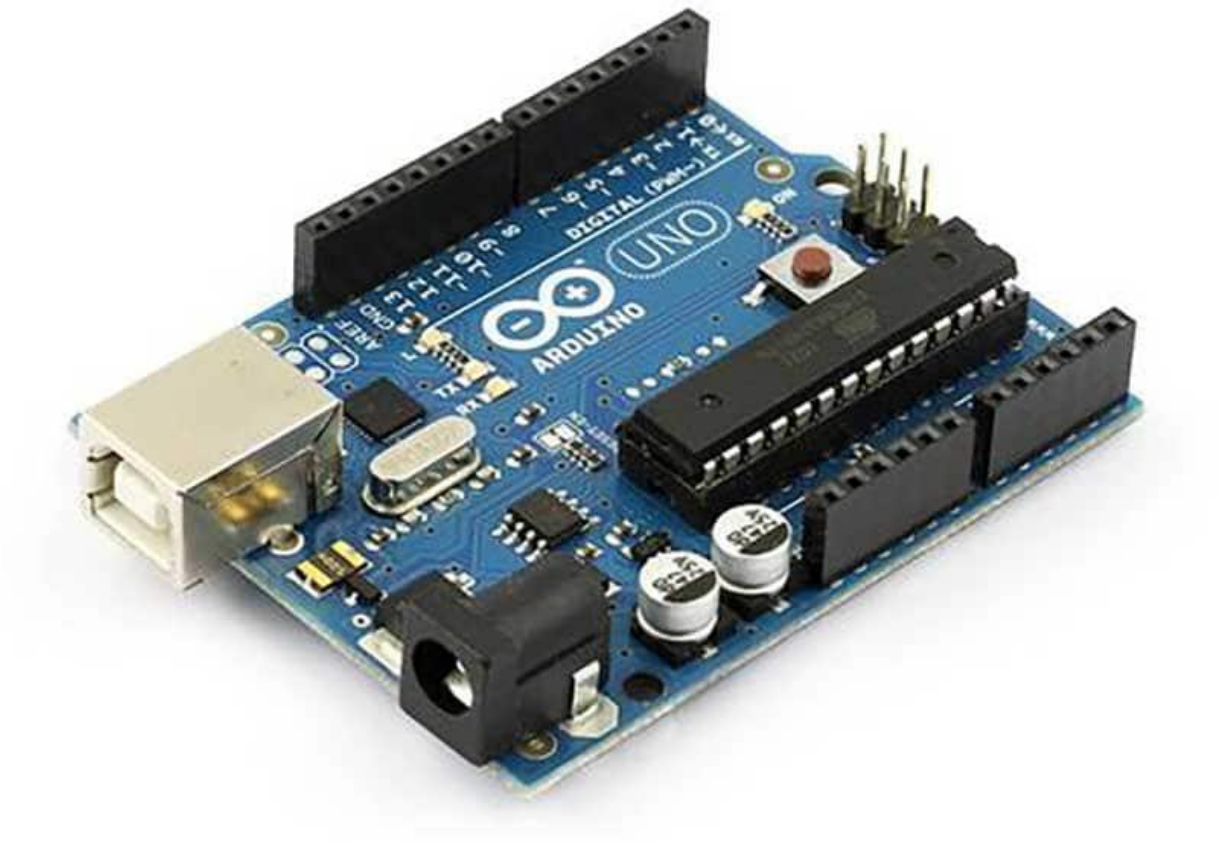
For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 8 Date: D D - M M - Y Y

Familiarize Arduino and IDE

Problem Statement : To install and configure Arduino IDE for Arduino Uno.

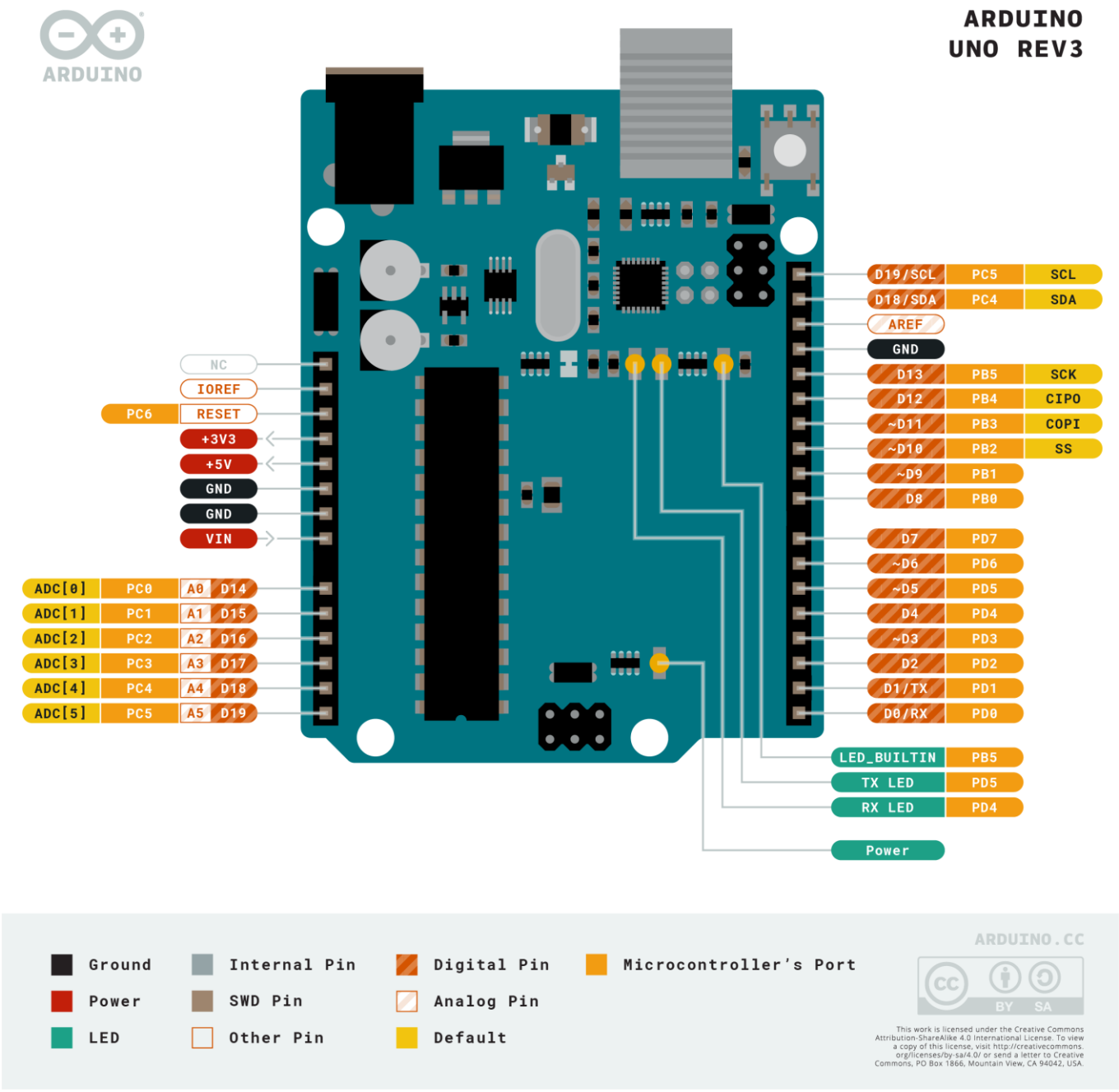
Arduino Uno



Overview

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your Uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of 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.



Installing Arduino IDE

Steps

For installation you need

- Arduino Uno board
- USB B Cable
- Windows 10, Windows 8, Windows 7, Mac, or Linux OS
- Arduino IDE

- About 15 minutes

Step 1: Download and Install the IDE

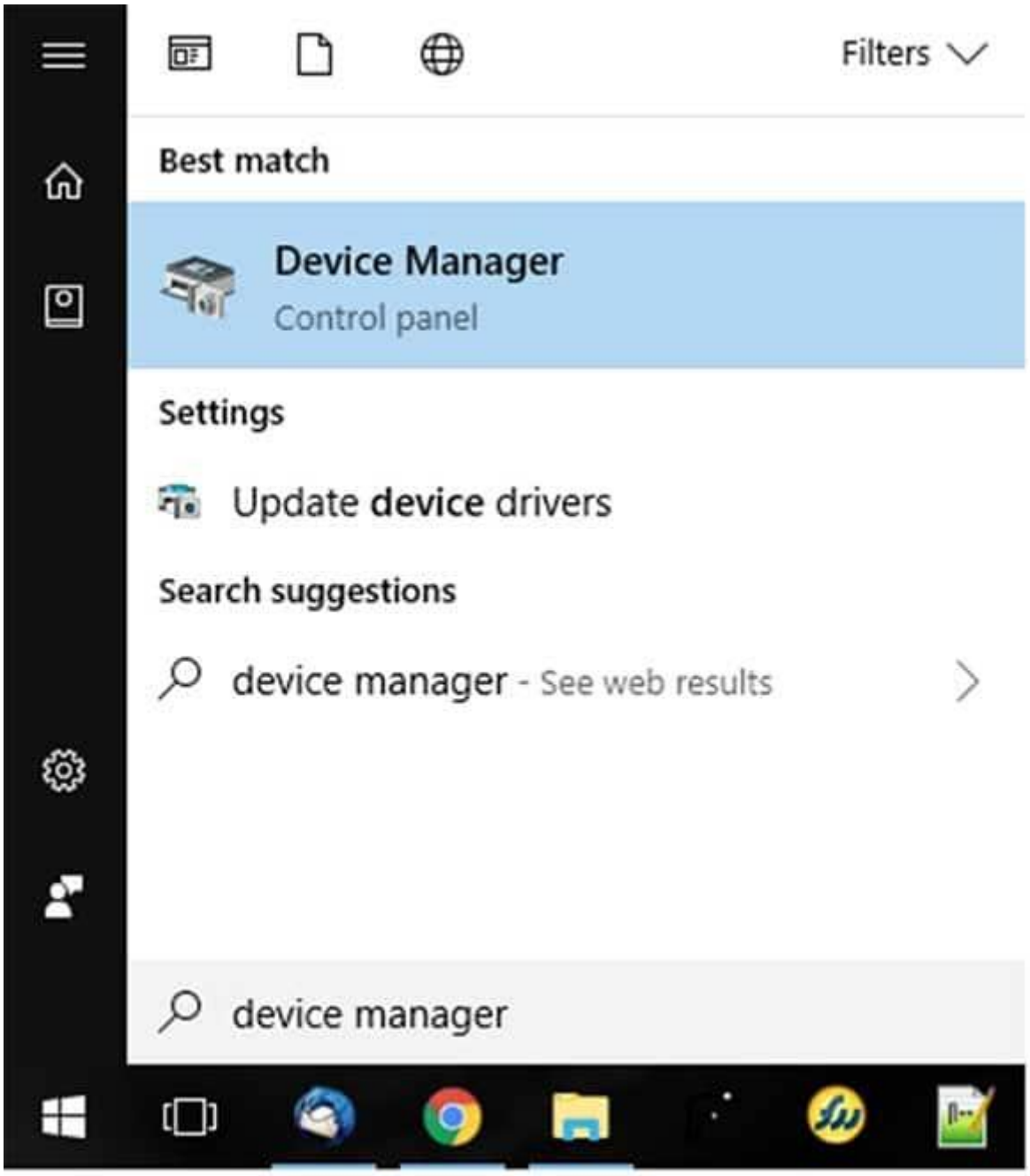
You can download the IDE from the official Arduino website. Since the Arduino uses a USB to serial converter (which allow it to communicate with the host computer), the Arduino board is compatible with most computers that have a USB port. Of course, you will need the IDE first. Luckily, the Arduino designers have released multiple versions of the IDE for different operating systems, including Windows, Mac, and Linux. In this tutorial, we will use Window 10, so ensure that you download the correct version of the IDE if you do not have Windows 10.

Go for downloading the Zip version and unzip the file in suitable folder.

Step 2: Get the Arduino COM Port Number

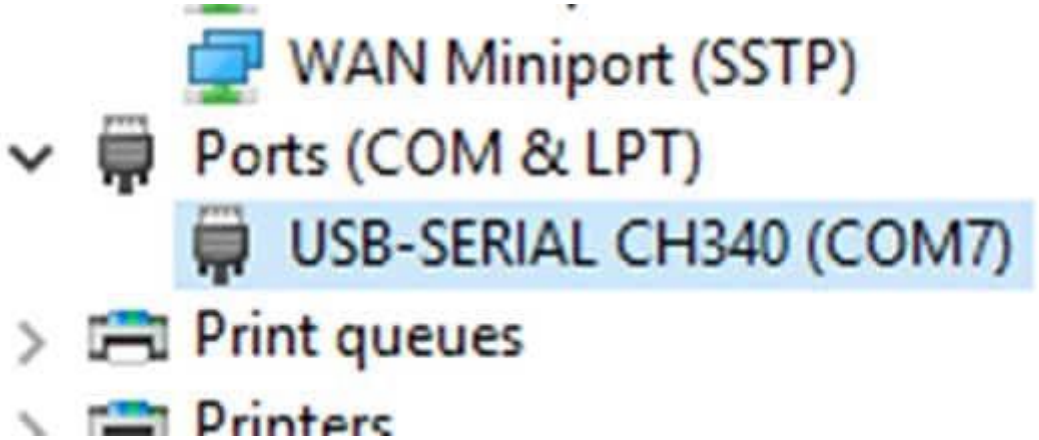
Next, you'll need to connect the Arduino Uno board to the computer. This is done via a USB B connection. Thanks to the wonderful world of USB, we do not need to provide power to the Arduino, as the USB provides 5V up to 2A. When the Arduino is connected, the operating system should recognize the board as a generic COM port (for example, my Arduino Uno uses a CH340G, which is an RS-232 serial to USB converter). Once it's recognized, we will need to find out what port number it has been assigned. The easiest way to do this is to type "device manager" into Windows Search and select Device Manager when it shows.

Finding the device manager option in Windows 10



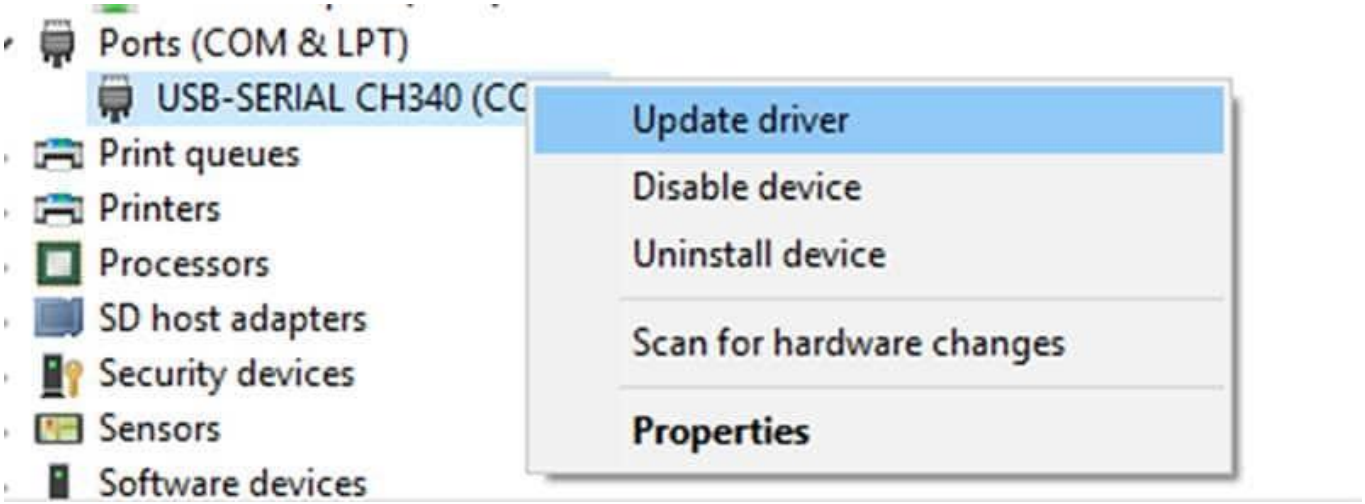
In the Device Manager window, look for a device under “Ports (COM & LPT)”, and chances are the Arduino will be the only device on the list.

The Arduino in my case is a CH340, and it shows on COM7 (port 7).

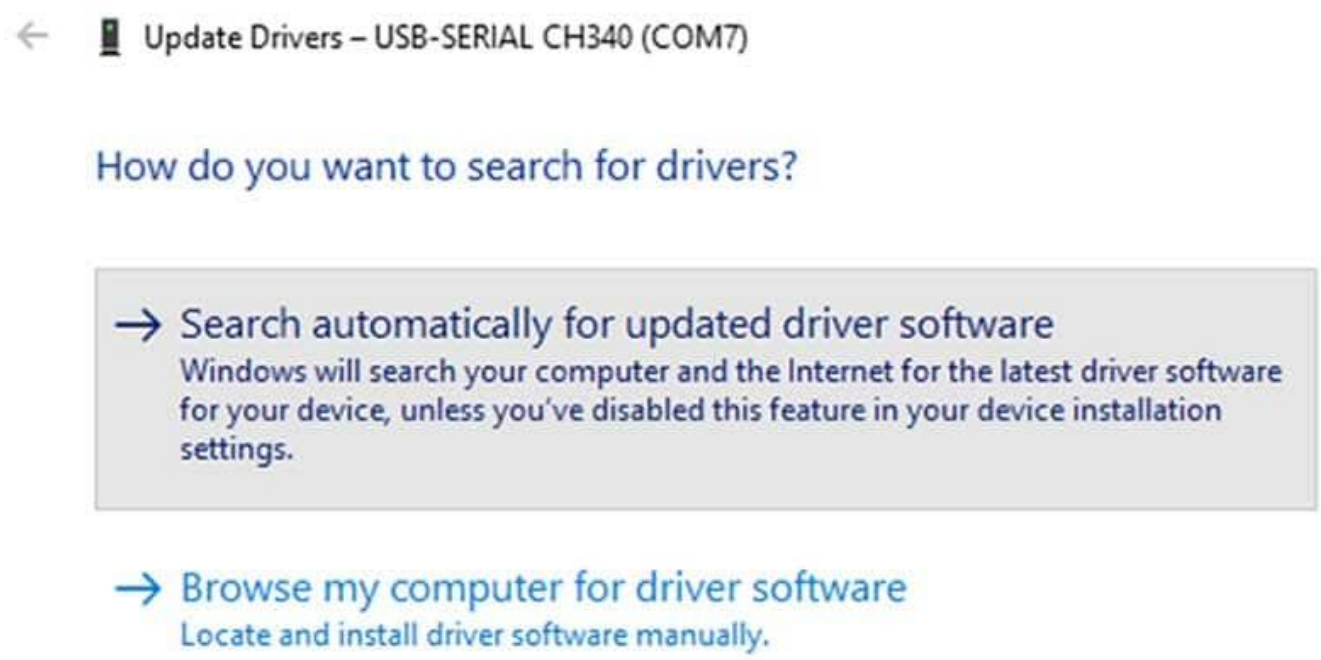


Be warned, the Arduino won't always be recognized automatically. If your Arduino is not recognized. then uninstall the driver, remove the Arduino, reinsert the Arduino, find the unrecognized device, right click "Update driver", and then click "Search automatically".

If the Arduino is not recognized, update the driver.



In the window that appears, click "Search automatically".



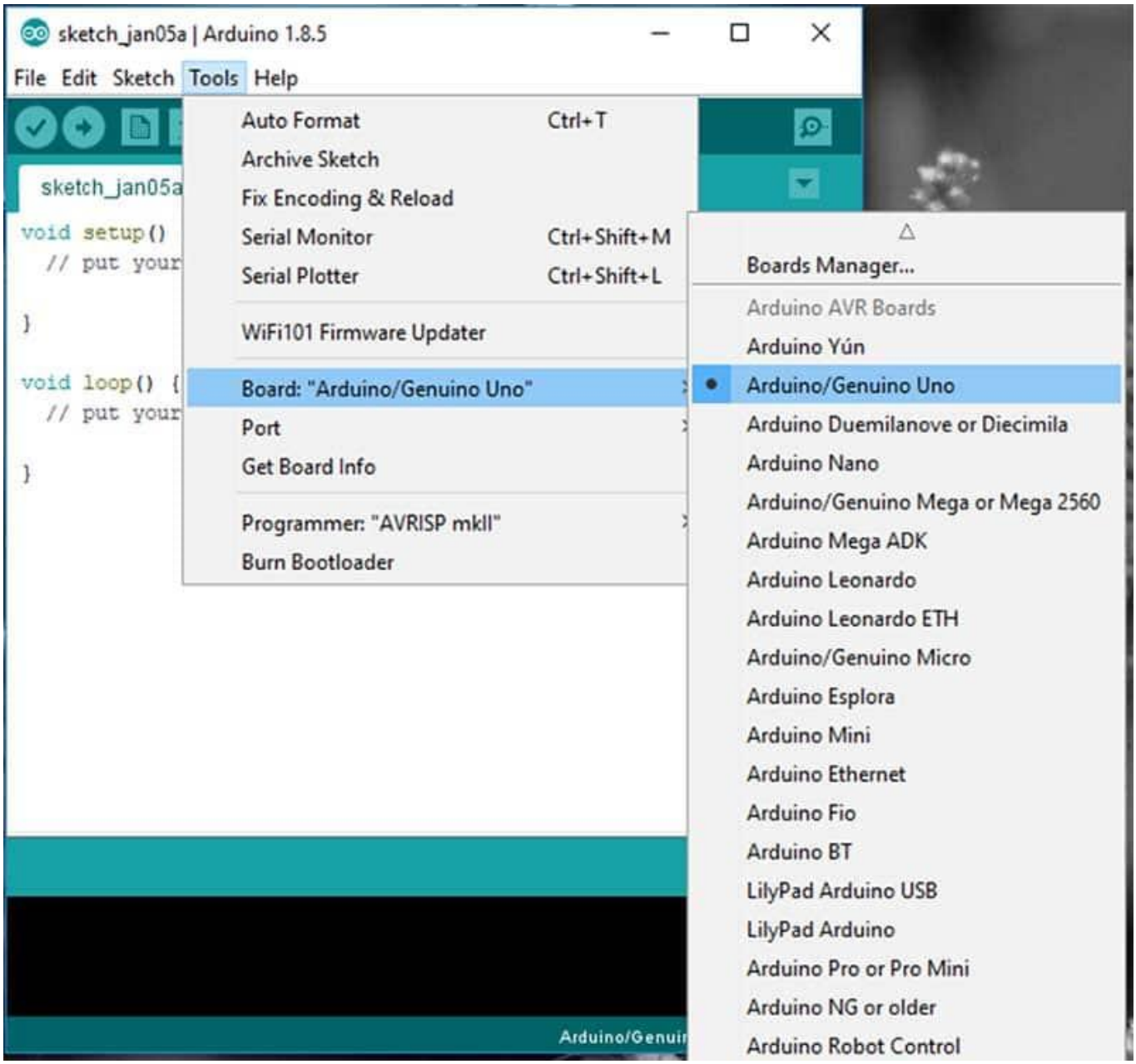
Windows can be a real pain sometimes with COM ports, as it can magically change their numbers between connections. In other words, one day, your Arduino may be on port 7 (as shown here), but then on other days, Windows may shift it to a different port number. This happens when you connect other COM ports to your system (which I do frequently).

So, if you can't find your Arduino on the port that you usually use, just go to your Device Manager and check what port it's actually on and, if necessary, update your driver.

Step 3: Configure the IDE

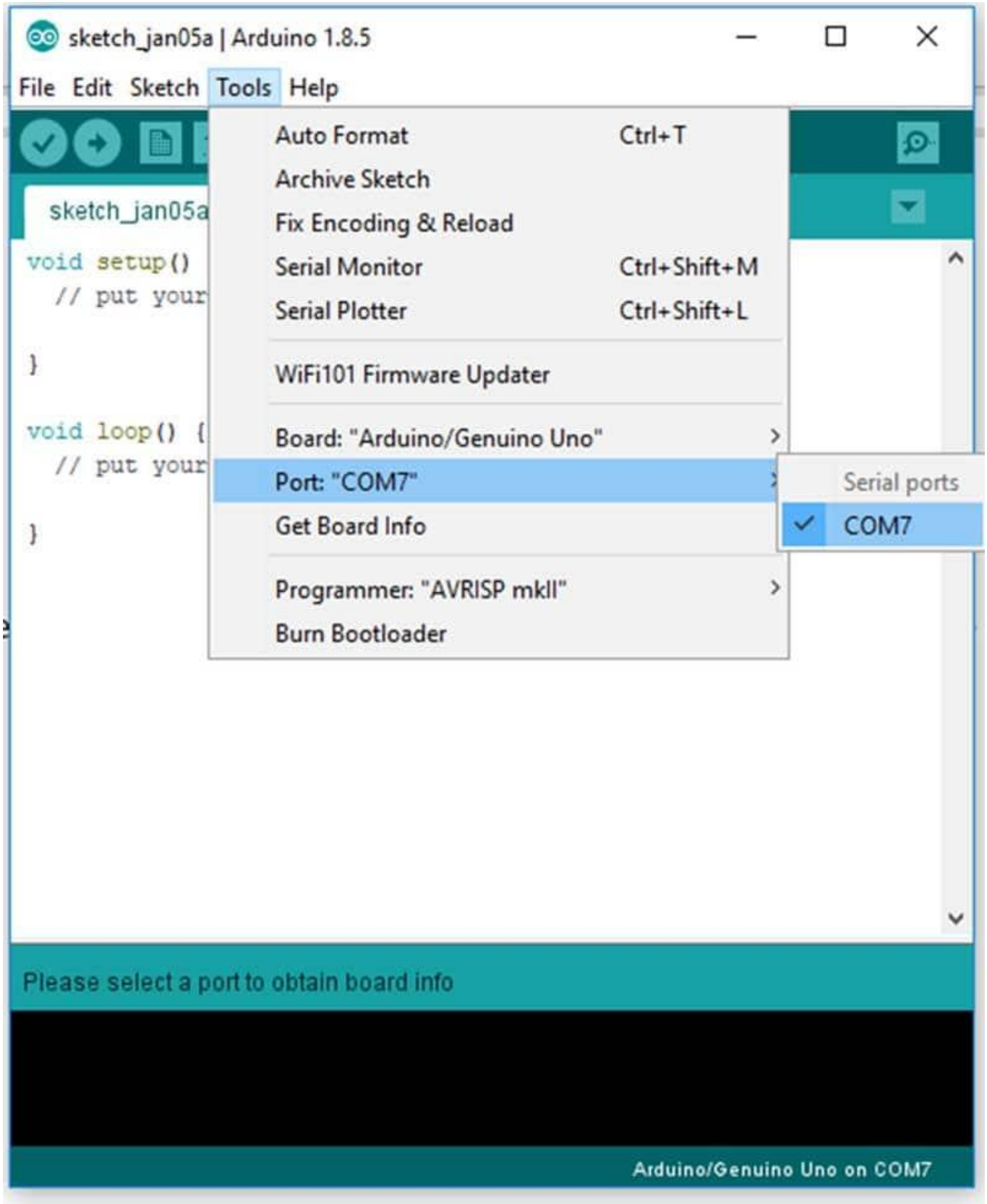
Now that we have determined the COM port that the Arduino is on, it's time to load the Arduino IDE and configure it to use the same device and port. Start by loading the IDE. Once it's loaded, navigate to Tools > Board > Arduino Uno. However, if you are using a different board (i.e., not the Arduino Uno), you must select the proper board!

Tell the IDE which board you are using.



Next, you must tell the IDE which COM port the Arduino is on. To do this, navigate to Tools > Port > COM7. Obviously, if your Arduino is on a different port, select that port instead.

Next, you must tell the IDE which COM port the Arduino is on. To do this, navigate to Tools > Port > COM7. Obviously, if your Arduino is on a different port, select that port instead.


sketch_jan05a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_jan05a
void setup()
// put your
}
void loop() {
// put your
}
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM7"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM7
Please select a port to obtain board info
Arduino/Genuino Uno on COM7

Result

The Arduino is a powerful prototyping tool for many reasons, including its lack of a dedicated programmer, its wide range of available libraries, and the simplicity of its IDE.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

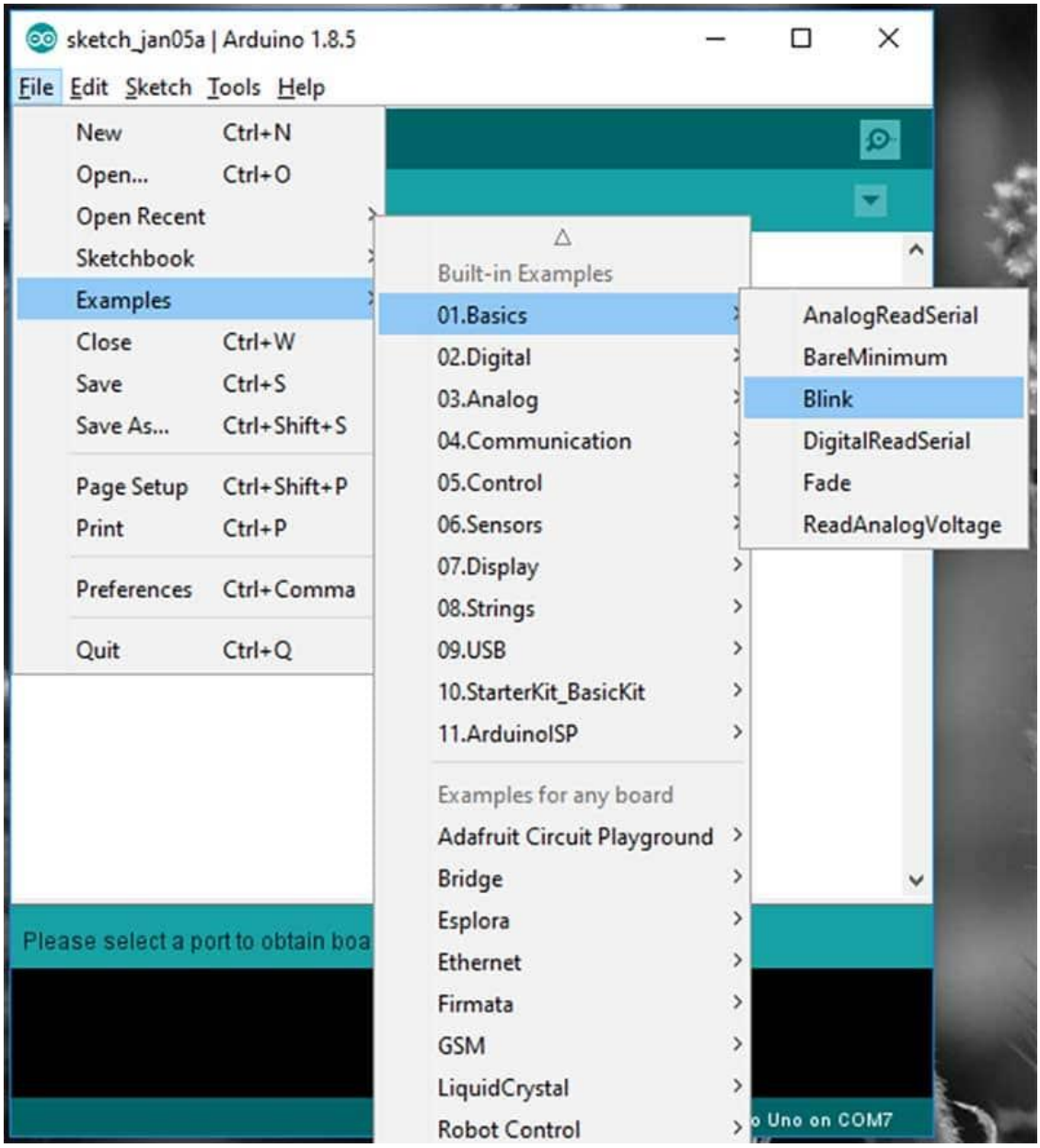
Exp No. 9 Date: D D - M M - Y Y

Blink LED connected to P13 at 1 sec interval. Interface relay module with port and switch relay on and off at different intervals.

Connect Arduino Uno to the computer with Arduino IDE installed. Proceed with loading examples.

Loading a Basic Example

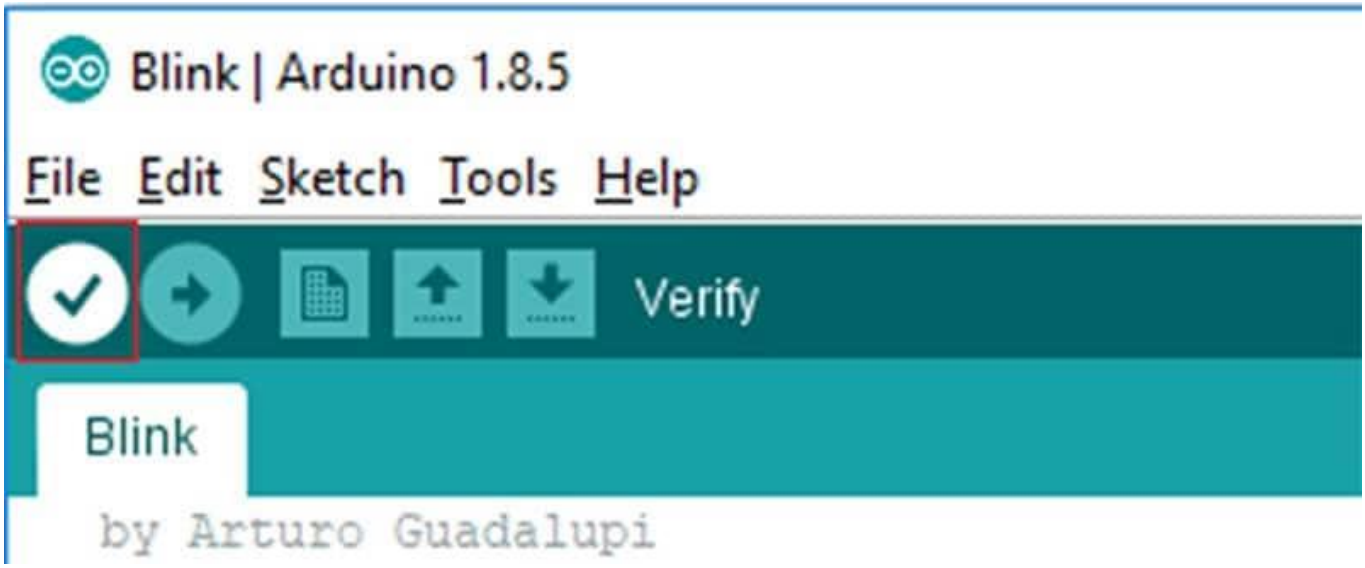
Let us load an example project that the Arduino IDE comes with. This example will make the onboard LED blink for a second continuously. To load this example, click File > Examples > 01.Basics > Blink.



With the example loaded, it's time to verify and upload the code. The verify stage checks the code for errors, then compiles the ready-for-uploading code to the Arduino. The upload stage actually takes the binary data, which was created from the code, and uploads it to the Arduino via the serial port.

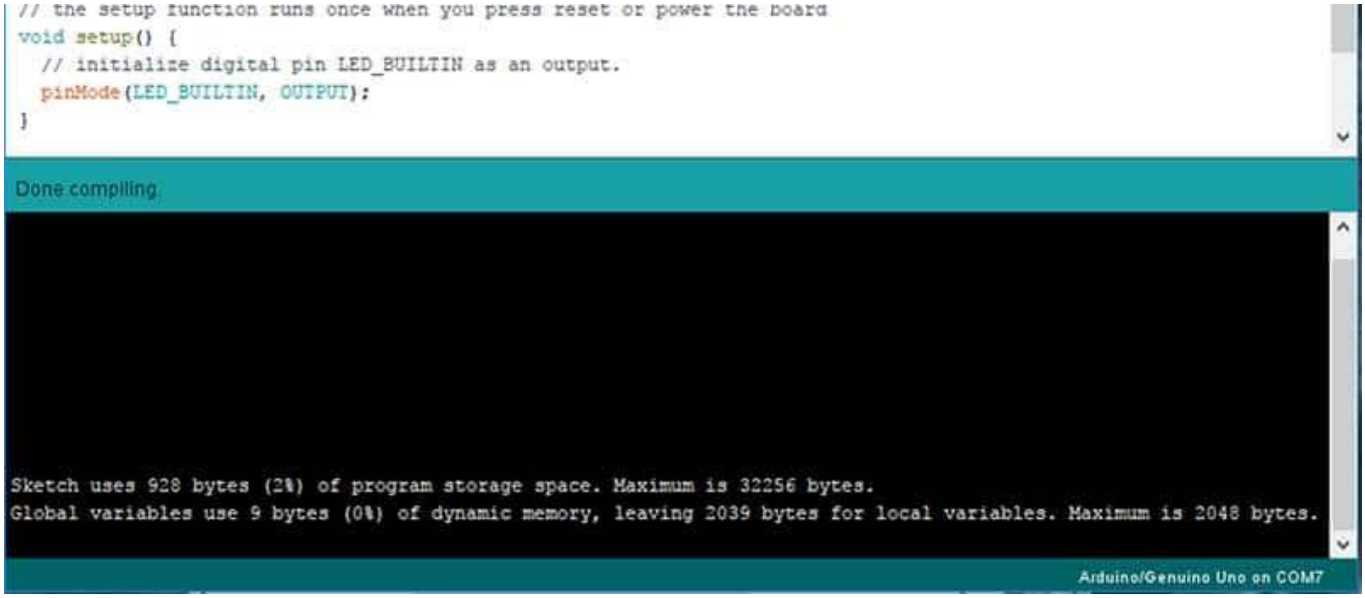
To verify and compile the code, press the check mark button in the upper left window.

The "Verify" button will compile the Arduino code.



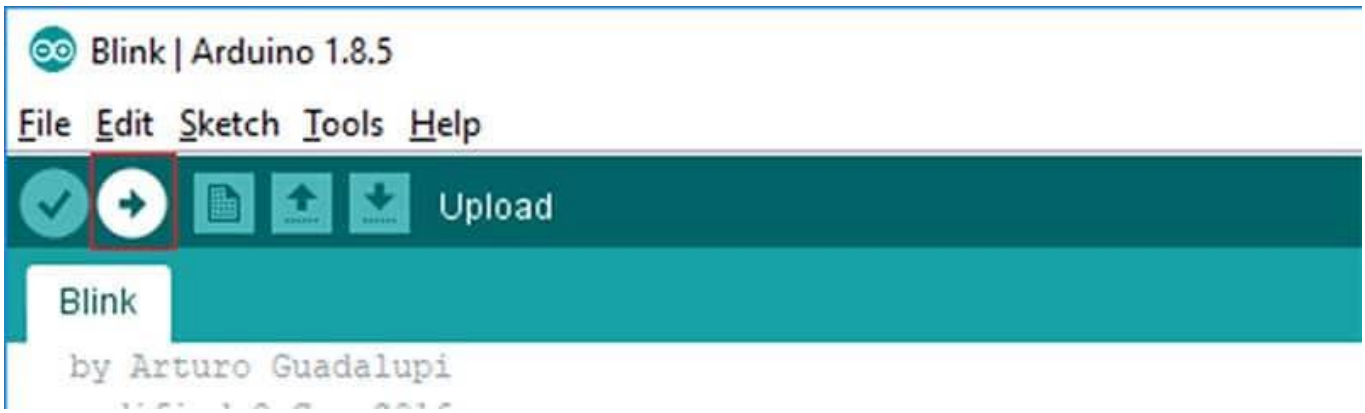
If the compilation stage was successful, you should see the following message in the output window at the bottom of the IDE. You might also see a similar message—just it's one that does not have words like "ERROR" and "WARNING".

This is a successful compilation.



With the code compiled, you must now upload it the Arduino Uno. To do this, click the arrow next to the check mark.

The "Upload" button will program the Arduino with your code.



Code Explained

After you build the circuit plug your Arduino board into your computer, start the Arduino Software (IDE) and enter the code below. You may also load it from the menu File – Examples - 01.Basics - Blink. The first thing you do is to initialize LED_BUILTIN pin as an output pin with the line

pinMode(LED_BUILTIN, OUTPUT);

In the main loop, you turn the LED on with the line:

digitalWrite(LED_BUILTIN, HIGH);

This supplies 5 volts to the LED anode. That creates a voltage difference across the pins of the LED, and lights it up. Then you turn it off with the line:

digitalWrite(LED_BUILTIN, LOW);

That takes the LED_BUILTIN pin back to 0 volts, and turns the LED off. In between the on and the off, you want enough time for a person to see the change, so the delay() commands tell the board to do nothing for 1000 milliseconds, or one second. When you use the delay() command, nothing else happens for that amount of time.

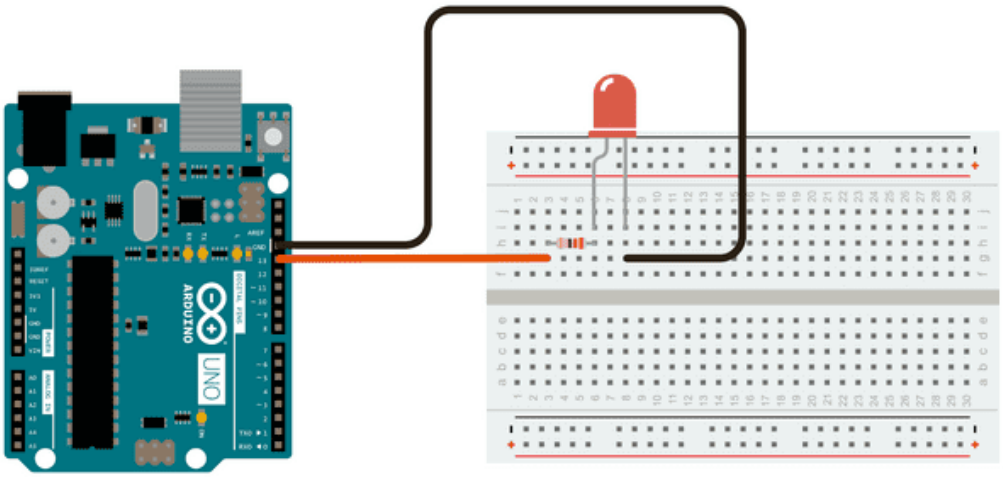
Code

```
/*
  Blink
  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and
  ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is
  set to the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your
  Arduino model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage
                                    //level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the
                                    //voltage LOW
  delay(1000);                      // wait for a second
}
```

Now try this

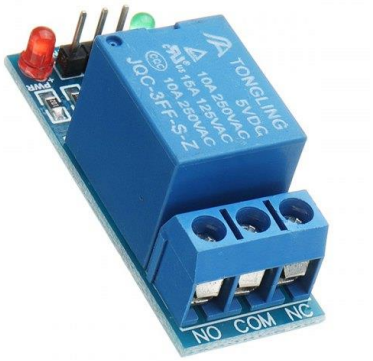
If you want to lit an external LED with this sketch, you need to build this circuit, where you connect one end of the resistor to the digital pin correspondent to the *LED_BUILTIN* constant. Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor. Connect the short leg of the LED (the negative leg, called the cathode) to the GND. In the diagram below we show an UNO board that has D13 as the LED_BUILTIN value.

The value of the resistor in series with the LED may be of a different value than 220 ohm; the LED will lit up also with values up to 1K ohm.



Interfacing relay module

Let us now interface a relay module with Arduino so as to control a 230V bulb. There many types of relay modules available like 1 relay, 2 relay, 5 relay and 8 relay but we are using the 1 relay Arduino module. Once you will understand this then you can use all other types.

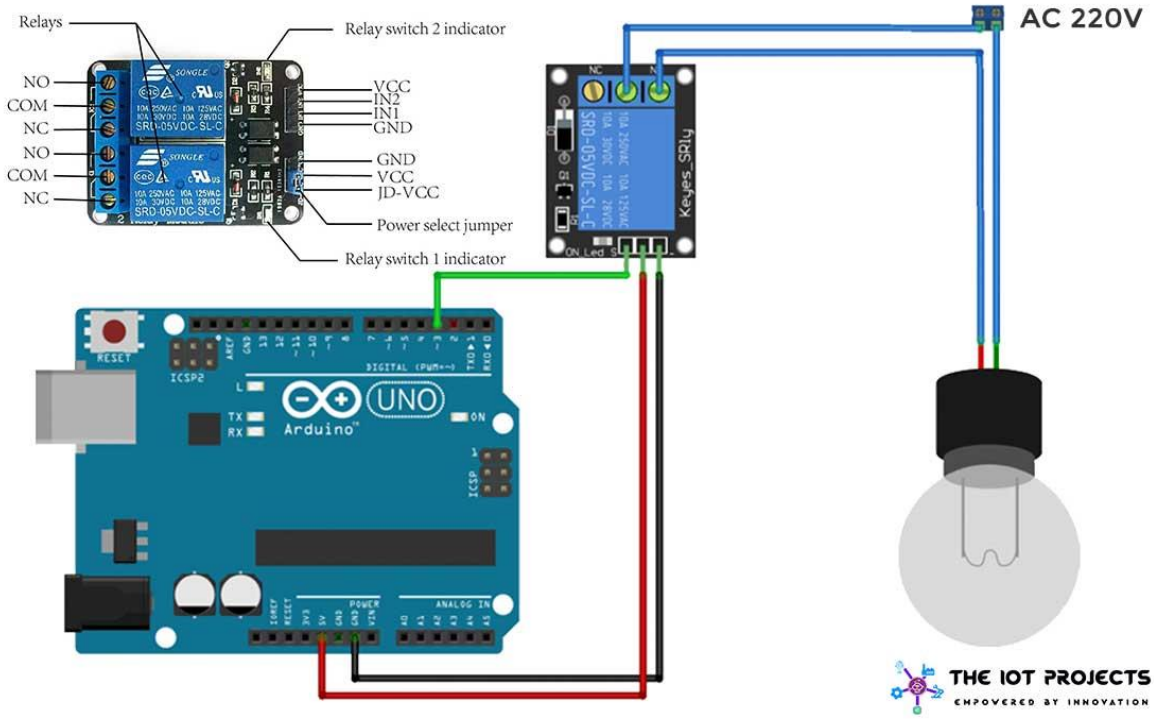


Input Connections of relay module

- Logic GND: This will be connected to GND on your Arduino.
- Input 1 (IN 1): This will be connected to digital pin on your Arduino. If you have multiple relay modules, you can leave the input unconnected if you do not want to use the channel.
- Logic VCC: This will be connected to the 5v pin of the Arduino.

Output Connections of relay module

- Before getting into the output connections of the relay, we have to understand the NO, COM and NC connections.
- COM (Common connection): The COM is the center terminal of the relay and it is used in both (Normally open and normally closed) connections.
- NO (Normally open): This act like a switch. In normally open connection, there will be no contact between COM and NO, since it is normally open. But when we will activate the relay module, then it will get connected to the COM and will supply power to the load, which will power up the light. Thus, the circuit will initially be in open state until we trigger the state.
- NC (Normally closed): its behaviour is opposite to the normally open connection. It always remains in contact with COM, even when relay module is not powered. When we will trigger the relay module then it will open the circuit, so the connection is lost.



Now let us control the relay module using I/O pin 3. Let us modify the code as

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(3, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(3, HIGH);  // turn the LED on (HIGH is the voltage
                          //level)
  delay(1000);                       // wait for a second
  digitalWrite(3, LOW);   // turn the LED off by making the
                          //voltage LOW
  delay(1000);                       // wait for a second
}
```

Result

While we only got a light to blink in this project, you can expect much more in the future.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 10 Date: D D - M M - Y Y

PIR sensor interfacing

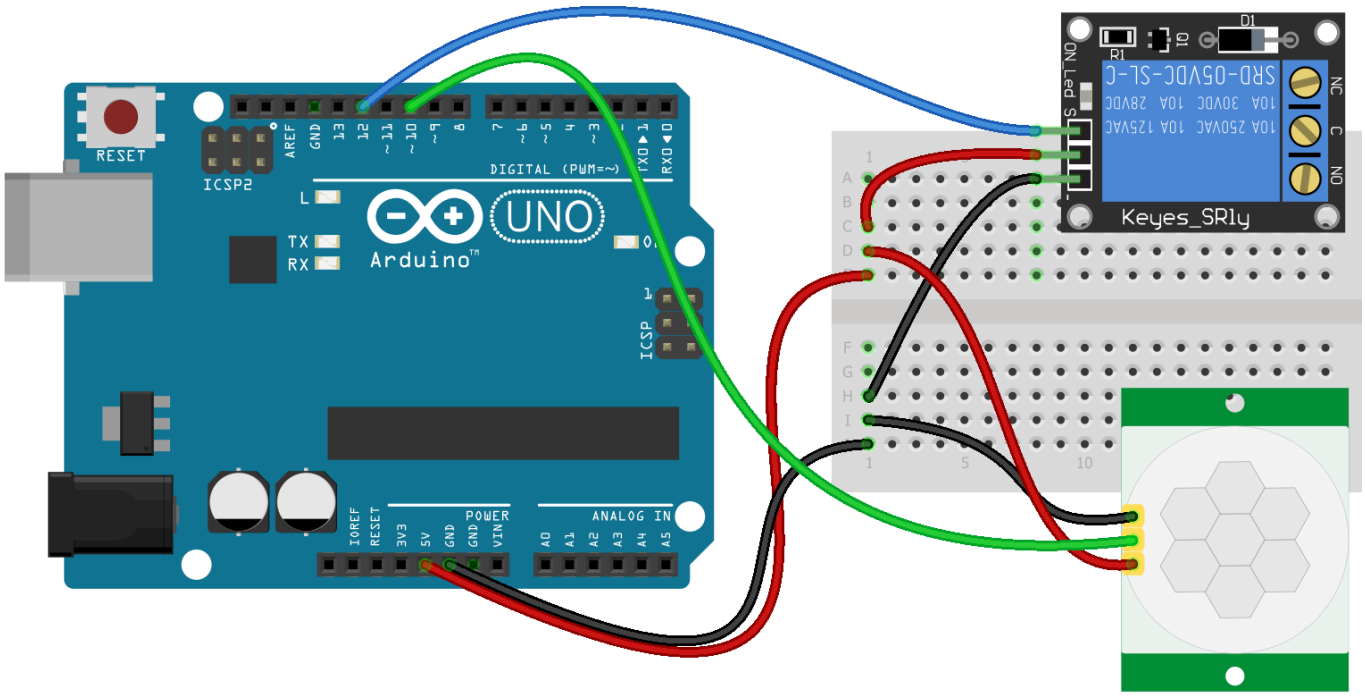
Objective: Connect PIR sensor (motion sensor) to port and switch relay connected to port on detecting motion.

PIR sensor:

The PIR sensor stands for Passive Infrared sensor. It is a low-cost sensor which can detect the presence of Human beings or animals. There are two important materials present in the sensor one is the pyroelectric crystal which can detect the heat signatures from a living organism (humans/animals) and the other is a Fresnel lenses which can widen the range of the sensor. Also, the PIR sensor modules provide us some options to adjust the working of the sensor as shown in above image.

Items required

1. Arduino Uno
2. PIR motion sensor
3. Relay module
4. Some female to male jumper Wires
5. Lamp



Let us use I/O pin 10 as input for PIR sensor and 12 for relay control. Let us edit the previous code as

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(12, OUTPUT);      //Pin 12 as output
  pinMode(10, INPUT);       //Pin 10 as input

}

// the loop function runs over and over again forever
void loop() {
```

let us copy digital read instruction from some examples. I have copied it from examples – Basics – DigitalReadSerial and us If statement as

```
  if (digitalRead(10)== HIGH)
{

  digitalWrite(12, HIGH);   // turn the Relay on
  delay(1000);              // wait for a second
}
  else
{
```

```
  digitalWrite(12, LOW);     // turn the relay off
  delay(1000);               // wait for a second
}
        }
```

See the usage of { } in if - else statement

Result

Interfacing of PIR sensor as input is performed. Usage of if – else statement and parenthesis observed.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

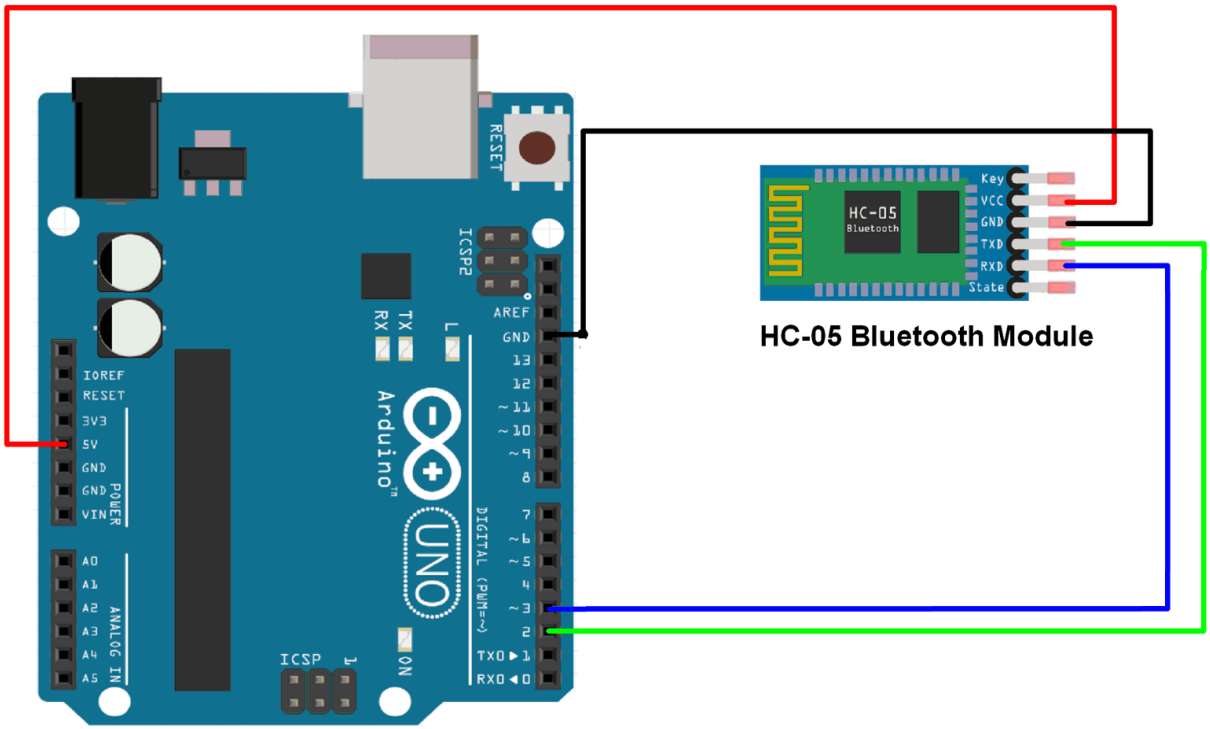
Exp No. 11 Date: D D - M M - Y Y

Bluetooth Module interfacing

Objective: Familiarize blue tooth module (HC 05 or HC 06) to switch relay connected to port using smart phone.

Interfacing Bluetooth Module (HC-05) with Arduino Uno

HC-05 is a Bluetooth device used for wireless communication with Bluetooth enabled devices (like smartphone). It communicates with microcontrollers using serial communication (USART). Default settings of HC-05 Bluetooth module can be changed using certain AT commands.



Default Bluetooth name of the device is “HC-05” and default PIN (password) for connection is either “0000” or “1234”. Here, we will transmit data from Smartphone via Bluetooth to the Arduino Uno and display it on Serial Monitor of PC.

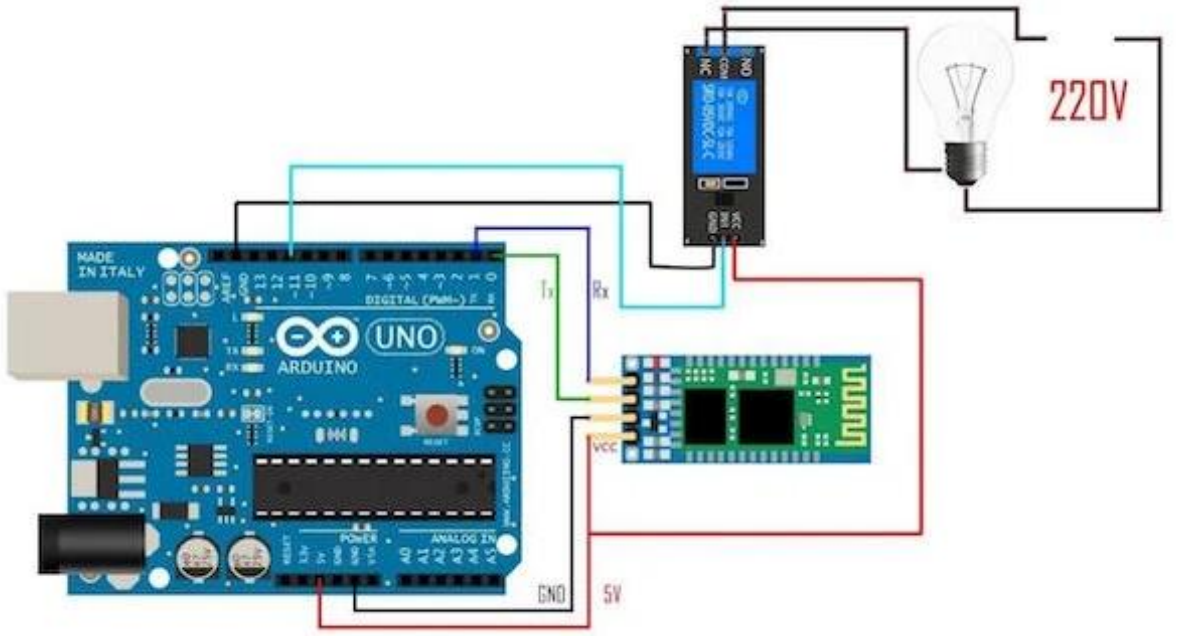
Download and install a Bluetooth terminal application on your phone and use it to connect to the HC-05 Bluetooth module. Data is sent from the Smartphone using the Bluetooth terminal application.

Let us now control LED bult in On and Off using Bluetooth.

```
void setup() {
 Serial.begin(9600);          // initializing serial port at 9600 baud
```

```
 pinMode(13,OUTPUT);           // pin 13 as output

}

void loop() {
while(Serial.available()>0)  // checking for data at serial port
{
  char x = Serial.read();    // reading serial port
  if (x=='A'){
  digitalWrite(13,HIGH);     // if A LED is On
  }
  else if (X=='B')
  {
  digitalWrite(13,LOW);      // if B LED is off
  }
  }
}
```

Now edit the code and control relay module with lamp connected to IO pin 10



Result

Bluetooth interfacing is done.

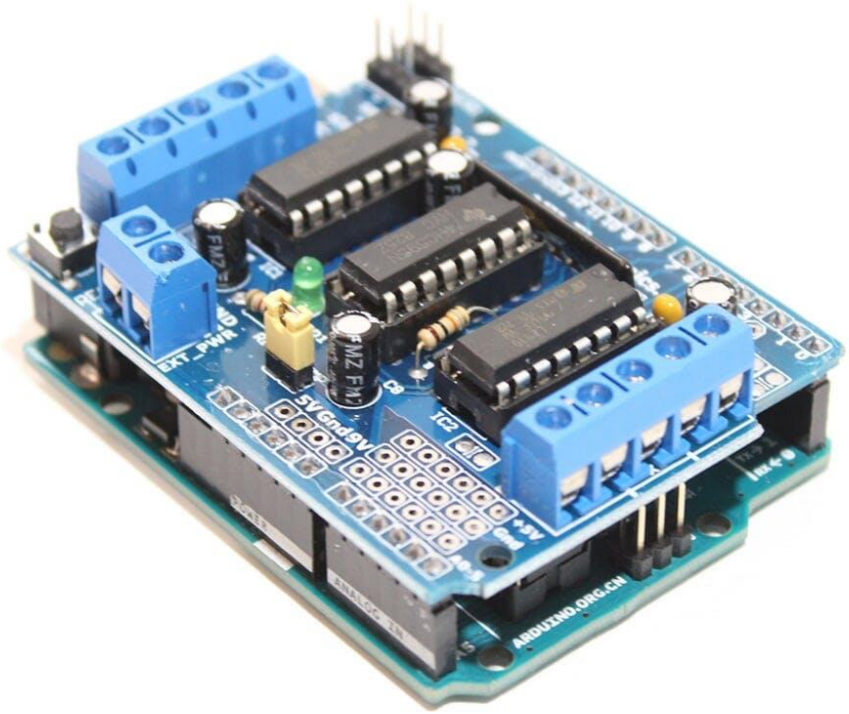
For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 12 Date: D D - M M - Y Y

Motor Shield interfacing

Objective: Familiarize DC motor control shield L293D and control DC motor for forward movement, reverse movement and stop.

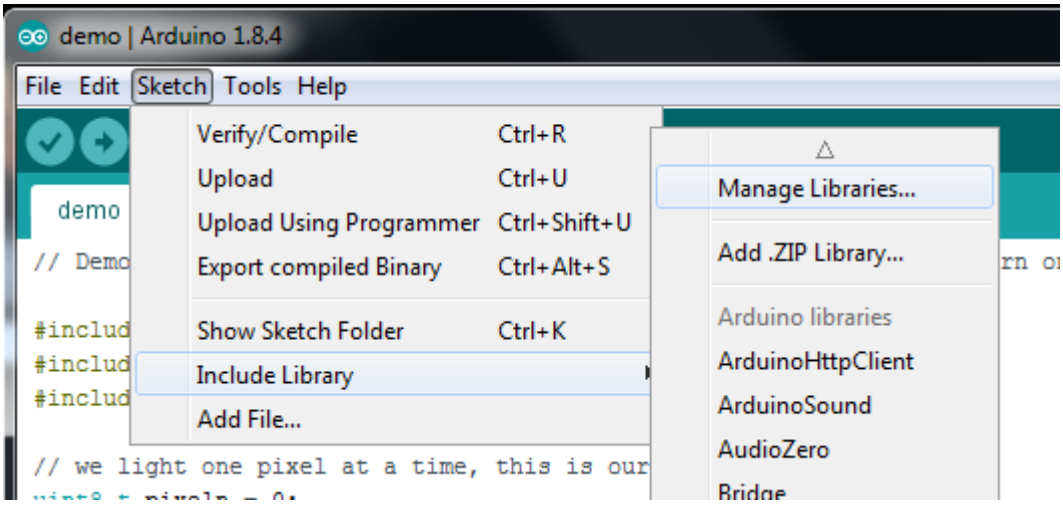
L293D shield is a driver board based on L293 IC, which can drive 4 DC motors and 2 stepper or Servo motors at the same time.



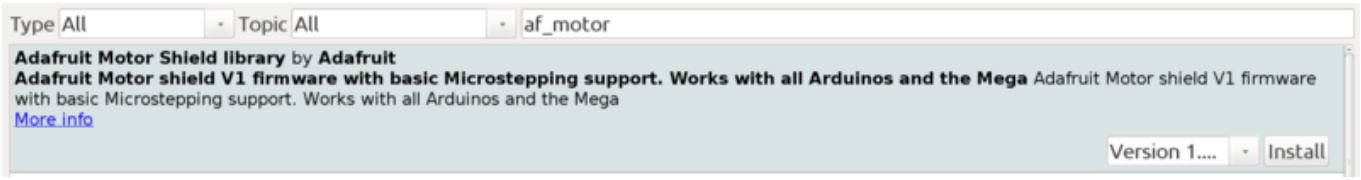
Each channel of this module has the maximum current of 1.2A and doesn't work if the voltage is more than 25v or less than 4.5v. So be careful with choosing the proper motor according to its nominal voltage and current.

First Install the Arduino Library

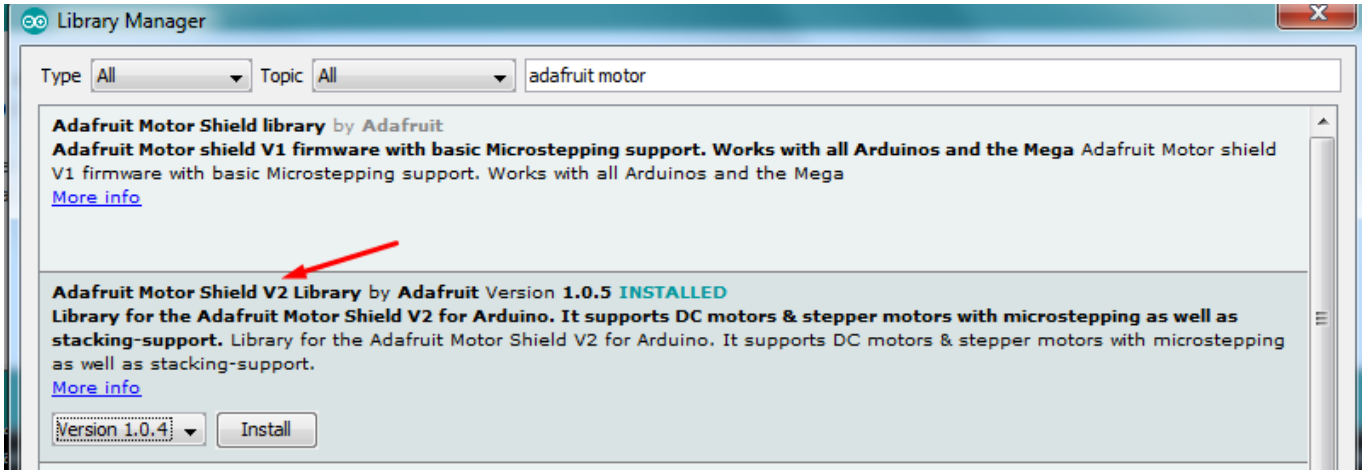
Before you can use the Motor shield, you **must** install the **AF_Motor** Arduino library - this will instruct the Arduino how to talk to the Adafruit Motor shield, and it isn't optional! Open up the Arduino library manager:



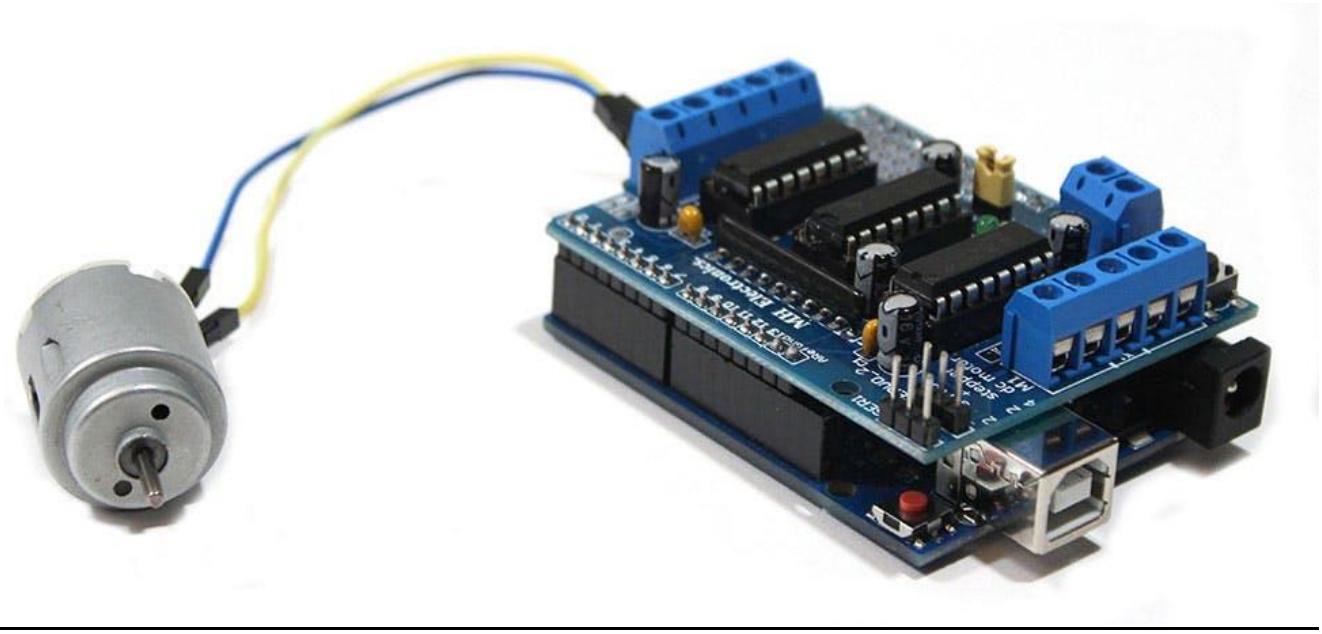
Search for Adafruit Motor library and install it. Make sure it is the library for the V1 motor shield.



Once installation is complete,



Driving DC Motor



```
// Adafruit Motor shield library
// copyright Adafruit Industries LLC, 2009
// this code is public domain, enjoy!

#include <AFMotor.h>

AF_DCMotor motor(1);              // edit port to which the motor is
connecetd

void setup() {
  Serial.begin(9600);             // set up Serial library at 9600 bps
  Serial.println("Motor test!");

  // turn on motor
  motor.setSpeed(200);

  motor.run(RELEASE);
}

void loop() {
  uint8_t i;

  Serial.print("forward");

  motor.run(FORWARD);
```

```
  for (i=0; i<255; i++) {
    motor.setSpeed(i);
    delay(10);
 }

  for (i=255; i!=0; i--) {
    motor.setSpeed(i);
    delay(10);
 }

  Serial.print("backward");

  motor.run(BACKWARD);
  for (i=0; i<255; i++) {
    motor.setSpeed(i);
    delay(10);
 }

  for (i=255; i!=0; i--) {
    motor.setSpeed(i);
    delay(10);
 }


  Serial.print("stop");
  motor.run(RELEASE);
  delay(1000);
}
```

Result

Interfacing of motor shield is done and motor tested.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

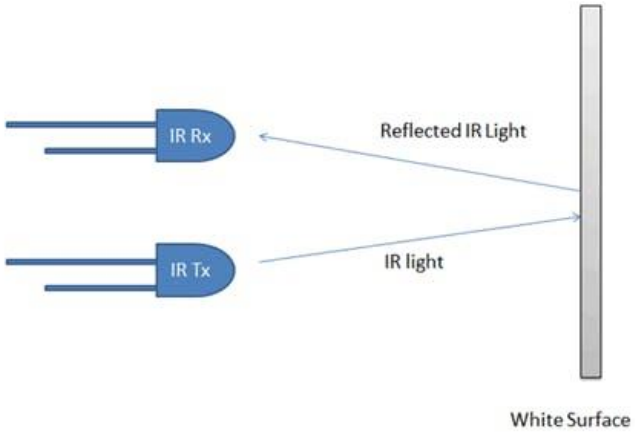
Exp No. 13 Date: D D - M M - Y Y

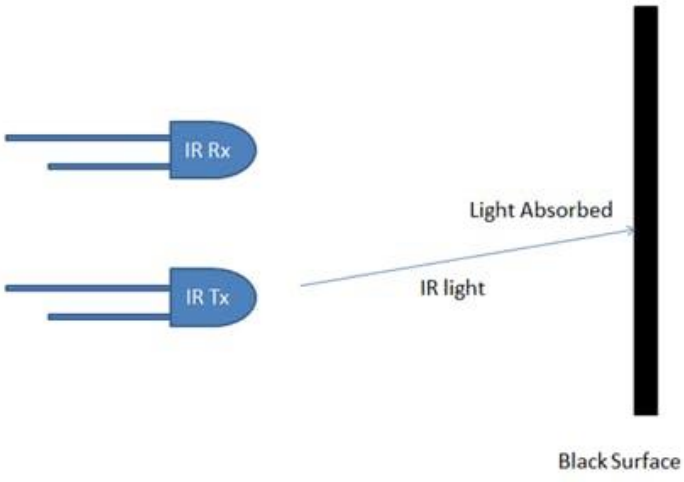
Line follower robot

Objective: Use wheeled robot kit (containing two geared DC motors, caster wheel, Arduino uno, motor driver shield, 9V battery and line sensor) to make line follower robot.

Concepts of Line Follower

Concept of working of line follower is related to light. We use here the behaviour of light at black and white surface. When light fall on a white surface it is almost full reflected and in case of black surface light is completely absorbed. This behaviour of light is used in **building a line follower robot**.

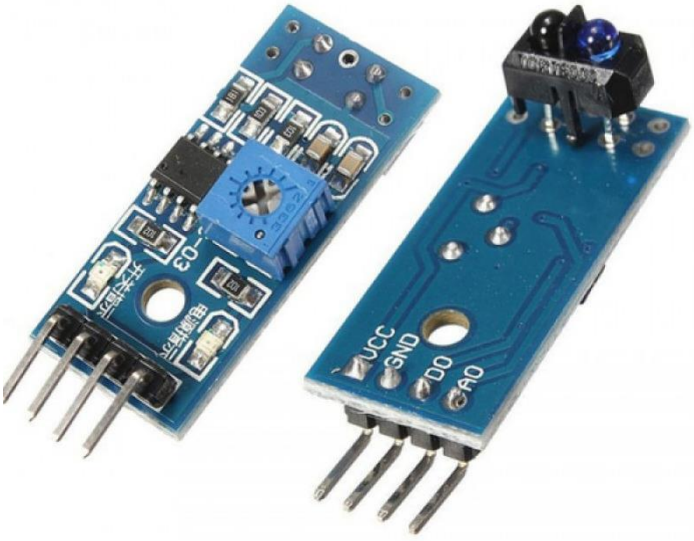


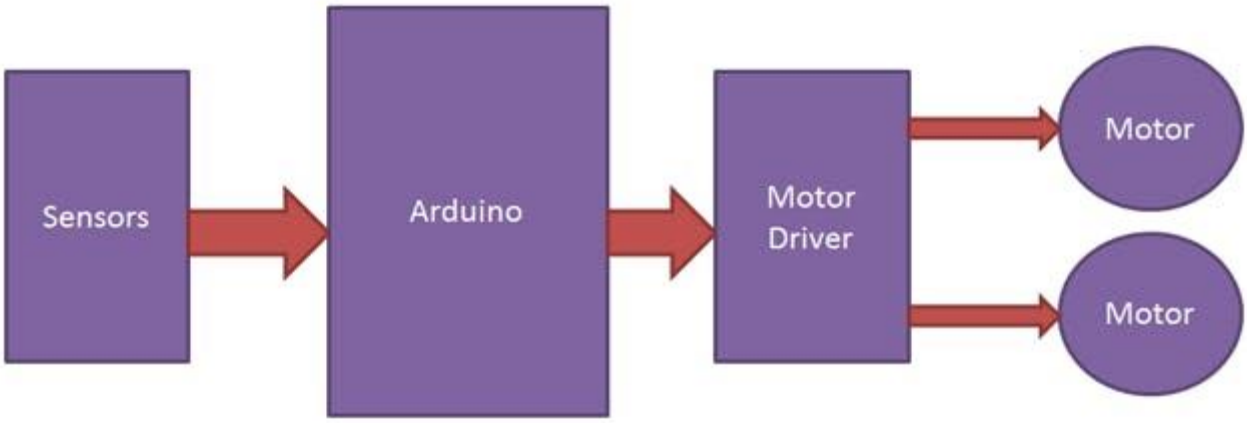


In this Arduino based line follower robot we have used IR Transmitters and IR receivers also called photo diodes. They are used for sending and receiving light. IR transmits infrared lights. When infrared rays fall on white surface, it's reflected back and catches by photodiodes which generates some voltage changes. When IR light falls on a black surface, light is absorbed by the black surface and no rays are reflected back, thus photo diode does not receive any light or rays.

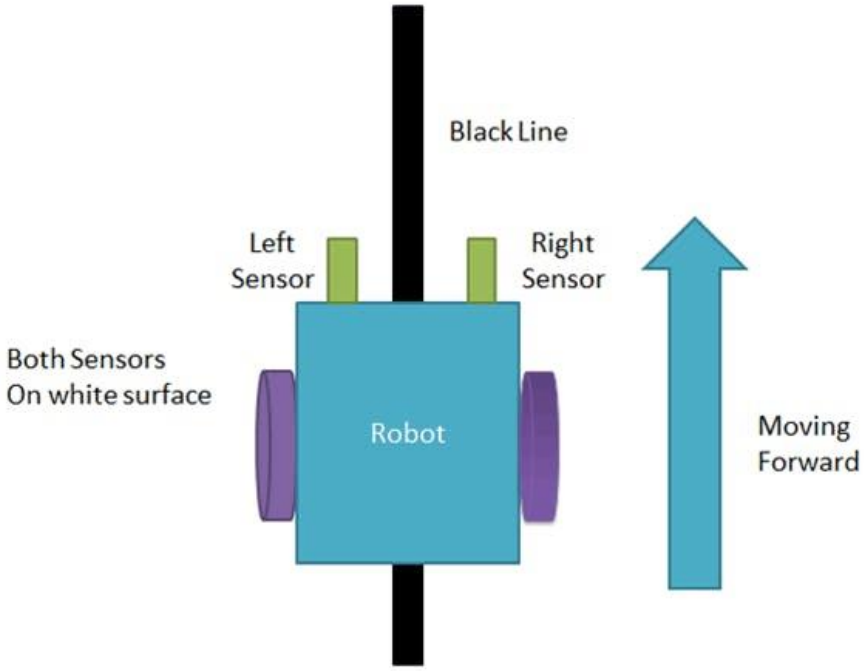
Here in this Arduino line follower robot when sensor senses white surface then Arduino gets 1 as input and when senses black line Arduino gets 0 as input.

A line follower sensor

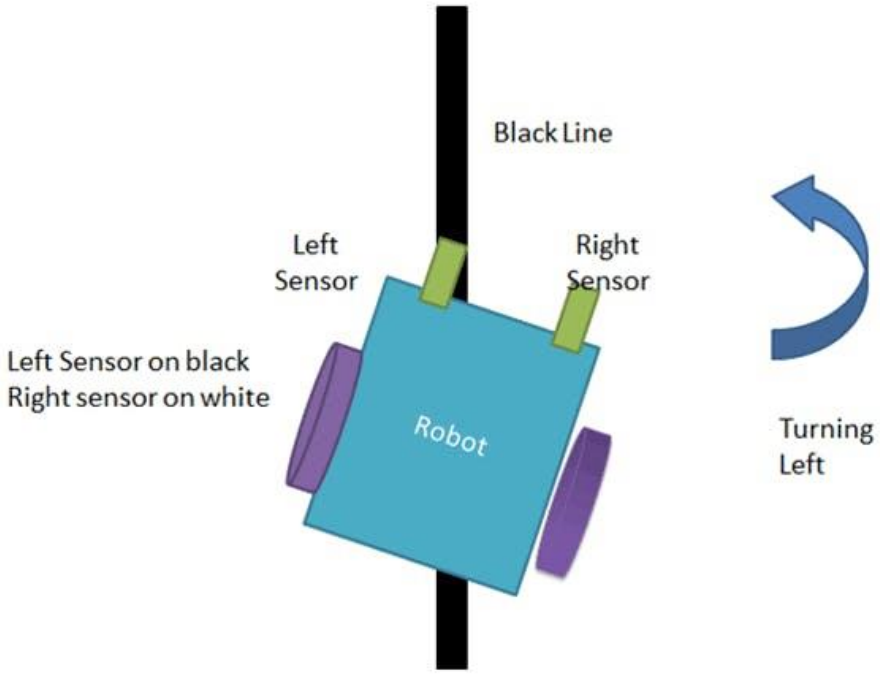




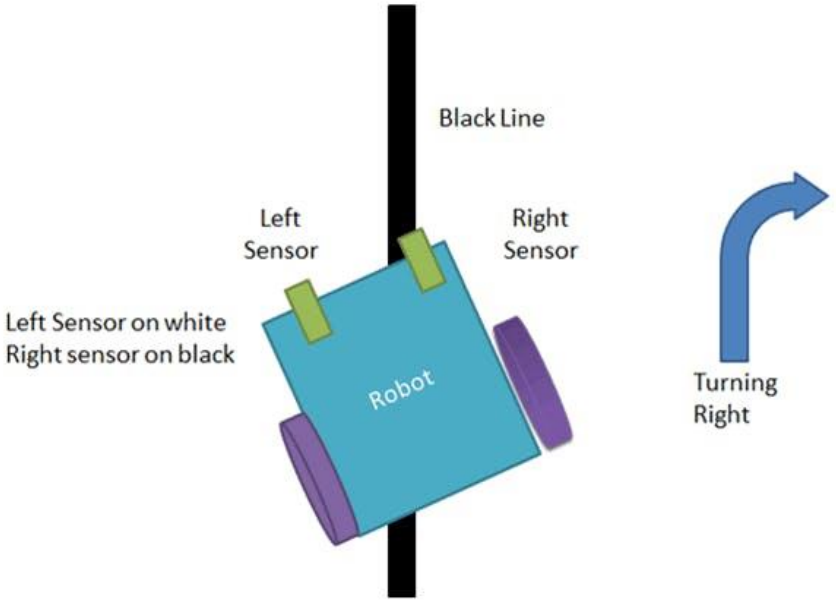
Here in this project, we are using two IR sensor modules namely left sensor and right sensor. When both left and right sensor senses white then robot move forward.



If left sensor comes on black line then robot turn left side.



If right sensor sense black line, then robot turn right side until each sensor comes at white surface. When white surface comes robot starts moving on forward again.



Connect both the sensors to A4 and A5.

Code

```
#include<AFMotor.h>

AF_DCMotor motor1(2,MOTOR12_1KHZ); //Motors
AF_DCMotor motor2(3, MOTOR12_1KHZ);

int sensorPin1 = A4; //LDR
int sensorPin2 = A5;

void setup()

 {
  Serial.begin(9600);   // Start Serial & set pin to output
  Serial.println("Motor test!");
  motor1.setSpeed(250);
  motor2.setSpeed (250);
  }

void loop()
{
  int sensorValue1 = analogRead(sensorPin1);// Read the sensor pin
  int sensorValue2 = analogRead(sensorPin2);

  //if light intensity it High
  if (sensorValue1 > 50)
  {
  Serial.println("Turn Right");
  motor1.run(FORWARD);
  }
  else
  {
    motor1.run(RELEASE);
  }

  if (sensorValue2 > 50)
  {
  Serial.println("Turn Left");
  motor2.run(FORWARD);
  }
  else
  {
    motor2.run(RELEASE);
  }
}
```

Result

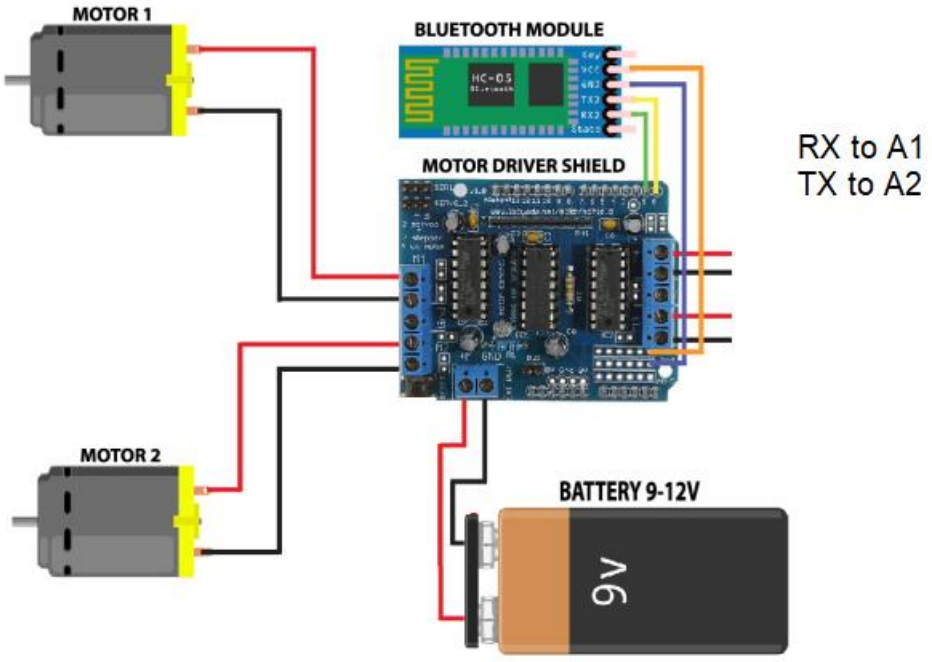
Line follower robot assembled coded and tested.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 14 Date: D D - M M - Y Y

Bluetooth controlled Robot

Use wheeled robot kit and blue tooth module to control robot using smart phone.



Code

```
// Adafruit Motor shield library
// copyright Adafruit Industries LLC, 2009
// this code is public domain, enjoy!

#include <AFMotor.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(A1, A2); // RX, TX
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
void setup() {
  mySerial.begin(9600);           // set up Serial library at 9600 bps
  motor1.setSpeed(200);
  motor2.setSpeed(200);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
}

void loop() {
  if (mySerial.available()) {
    char x = mySerial.read();

    if (x == 'A')
    {
      motor1.run(FORWARD);      // robot runs forward
      motor2.run(BACKWARD);
      delay (100);              // for 100 msec
      motor1.run(RELEASE);      // and stops
      motor2.run(RELEASE);
      x = ' ';
    }
    else if (x == 'B')
    {
      motor2.run(FORWARD);      // robot runs backward
      motor1.run(BACKWARD);
      delay (100);              // for 100 msec
      motor1.run(RELEASE);      // and stops
      motor2.run(RELEASE);
      x = ' ';
    }
    else if (x == 'C')
    {
      motor1.run(FORWARD);      // robot runs backward
      motor2.run(RELEASE);
      delay (100);              // for 100 msec
      motor1.run(RELEASE);      // and stops
      x = ' ';
```

```
    }
    else if (x == 'D')
    {
      motor2.run(BACKWARD);       // robot runs backward
      motor1.run(RELEASE);
      delay (100);               // for 100 msec
      motor2.run(RELEASE);       // and stops
      x = ' ';
    }
  }
}
```

Result

Bluetooth controlled robot assembled, coded and tested.

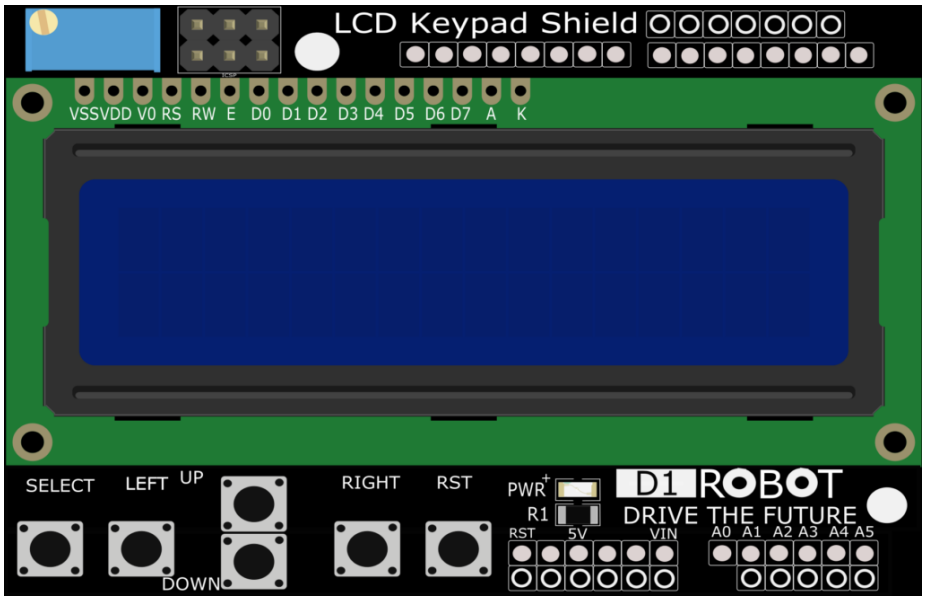
For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 15 Date: D D - M M - Y Y

LCD Shield interfacing

Objective: Use LCD shield and display your name and class number.

The 16X2 LCD Keypad Shield build for Arduino UNO, MEGA, Leonardo and other microcontrollers. This shield is fully compatible with Arduino 4-bit "LiquidCrystal" library. It includes 16X2 LCD display with 6 push Buttons, among which 5 are connected with A0 analog input through resistor to give different voltage for each button and saving on input/output pins. And Pins 4, 5, 6, 7, 8, 9 and 10 are used to interface with the LCD.



```
#include <LiquidCrystal.h>          // header file

LiquidCrystal lcd( 8,9,4,5,6,7 );  //interfacing pins

void setup()
{

   // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);                 // initializing LCD.
  lcd.setCursor(0,0);               // set cursor to first row first
column
  lcd.print("Your Name           ");
  lcd.setCursor(0,1);               // set cursor to second row first
column
  lcd.print("Your Register no.   ");
}

void loop()
{
}
```

Result

Interfacing of LCD shield done

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 13 Date: D D - M M - Y Y

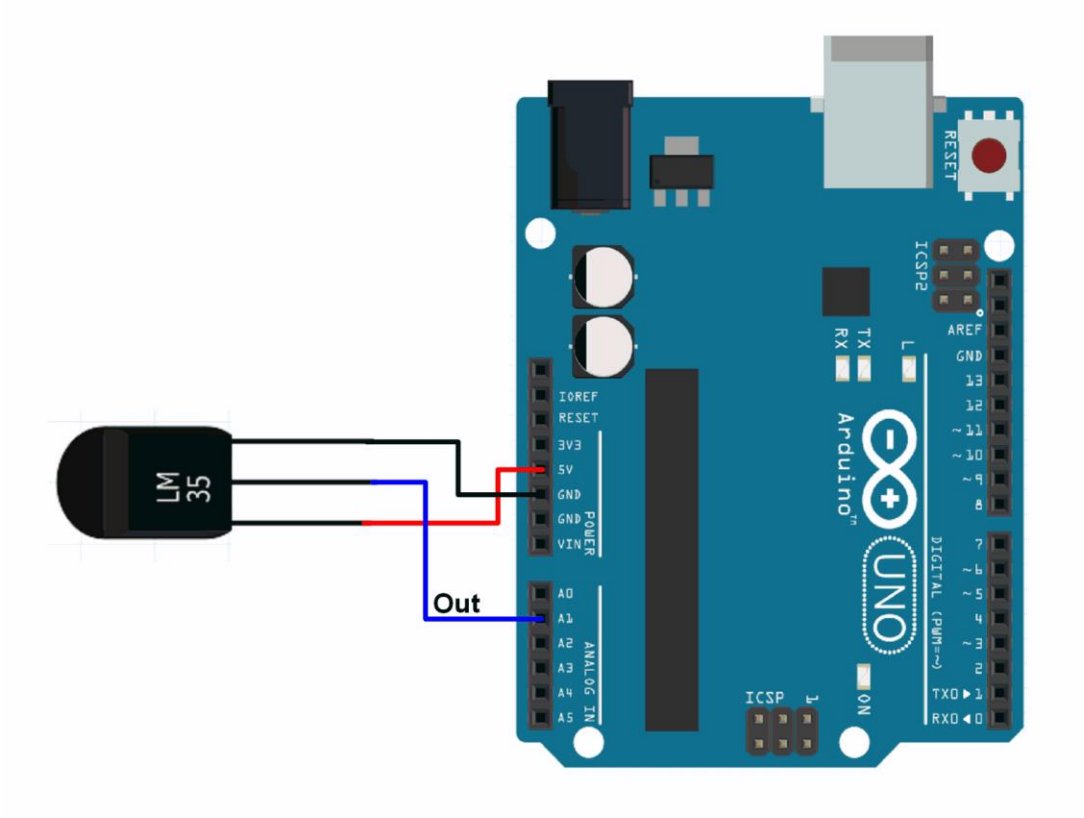
An Arduino DIY (Do It Yourself) project

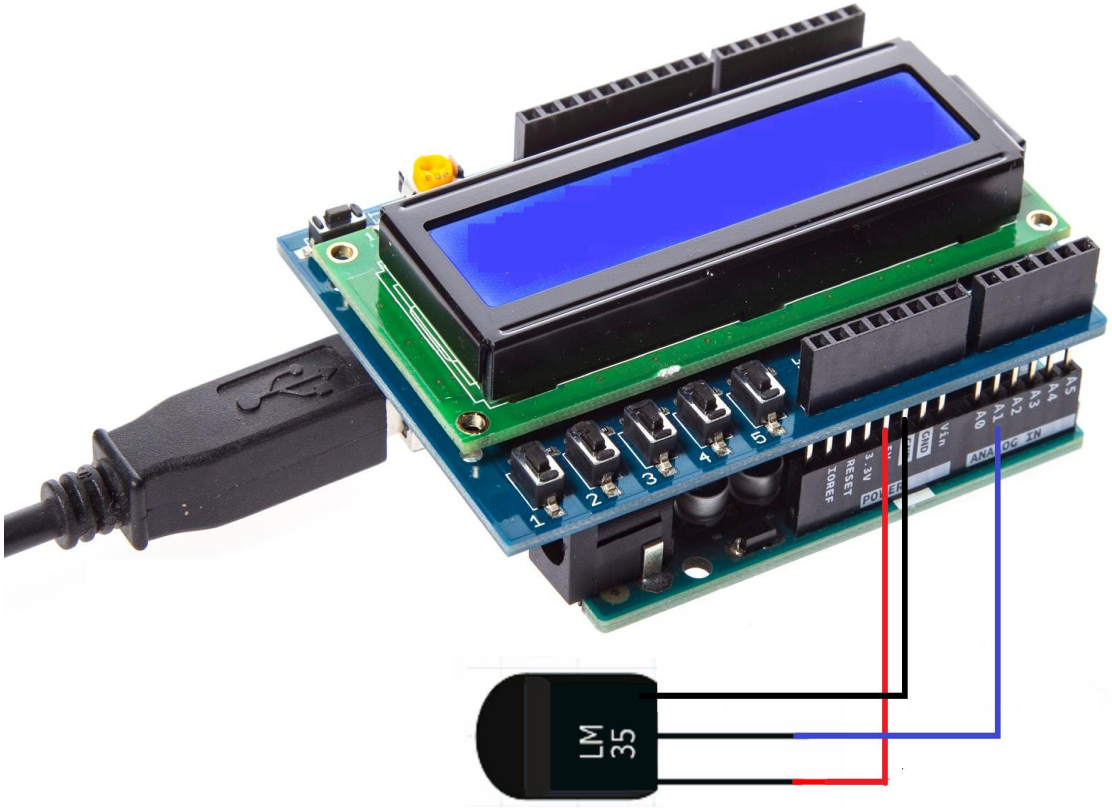
Objective: Interface LCD shield and LM35 (temperature sensor to display temperature

LM35

LM35 is a temperature sensor which can measure temperature in the range of -55°C to 150°C. It is a 3-terminal device that provides analog voltage proportional to the temperature. Higher the temperature, higher is the output voltage. The output analog voltage can be converted to digital form using ADC so that a microcontroller can process it.

Interfacing


LM
35
Out
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX
RX
L
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
Arduino™
UNO
DIGITAL (PWM=~)
TXD 1
RXD 0
ICSP
ON


LM
35

Code

```
#include <LiquidCrystal.h>          // header file

LiquidCrystal lcd( 8,9,4,5,6,7 );  //interfacing pins

void setup()
{

   // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);                // initializing LCD.
  lcd.setCursor(0,0);              // set cursor to first row first column
  lcd.print("Temperature =           ");
  lcd.setCursor(0,1);              // set cursor to second row first column
  lcd.print("GPC Perumbavoor");
}

void loop()
{
  int temp_adc_val;
  float temp_val;
  temp_adc_val = analogRead(A1); /* Read Temperature */
  temp_val = (temp_adc_val * 4.88); /* Convert adc value to equivalent
voltage */
  temp_val = (temp_val/10); /* LM35 gives output of 10mv/°C */
  lcd.setCursor(0,12);             // set cursor to first row first column
  lcd.print(temp_val);
  delay(2000);
}
```

Result

Room temperature displayed

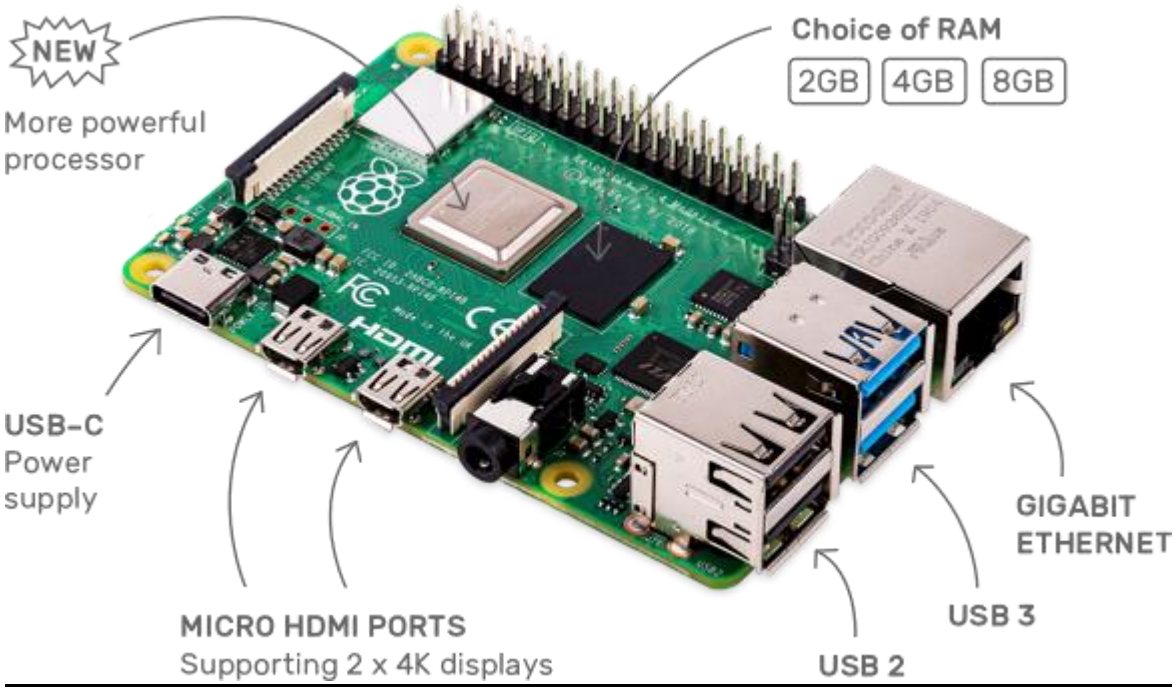
For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 14 Date: D D - M M - Y Y

Familiarize Raspberry Pi and compatibles.

The following are raspberry pi and compatibles

1. Raspberry Pi - 4



Raspberry Pi is a series of small single-board computers (SBCs) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries. The original model became more popular than anticipated, selling outside its target market for uses such as robotics. It is widely used in many areas, because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of HDMI and USB devices.

2. Raspberry pi case

3. Micro SD Card

Raspberry Pi Operating system is stored in micro-SD card.

4. Micro SD card adaptor

Micro SD card adaptor is used to transfer OS from computer to SD card.

For USB

for laptop with SD card slot

5. Power adaptor (3A 5V)

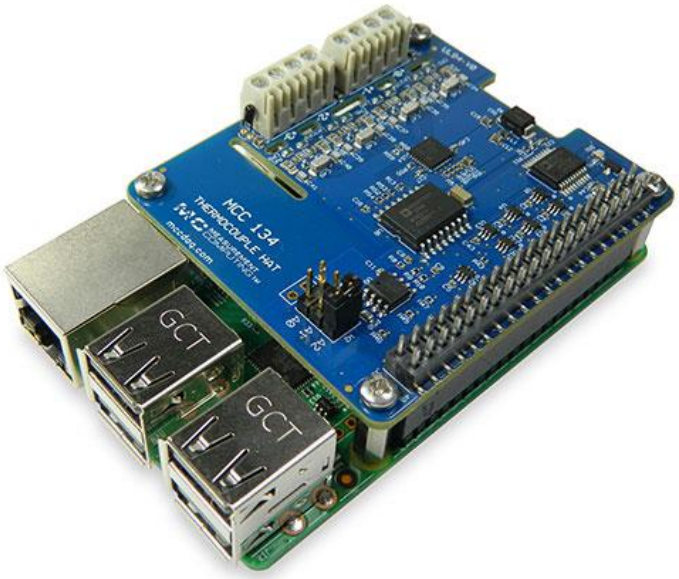
A two-ampere mobile adaptor can be used for powering Raspberry pi

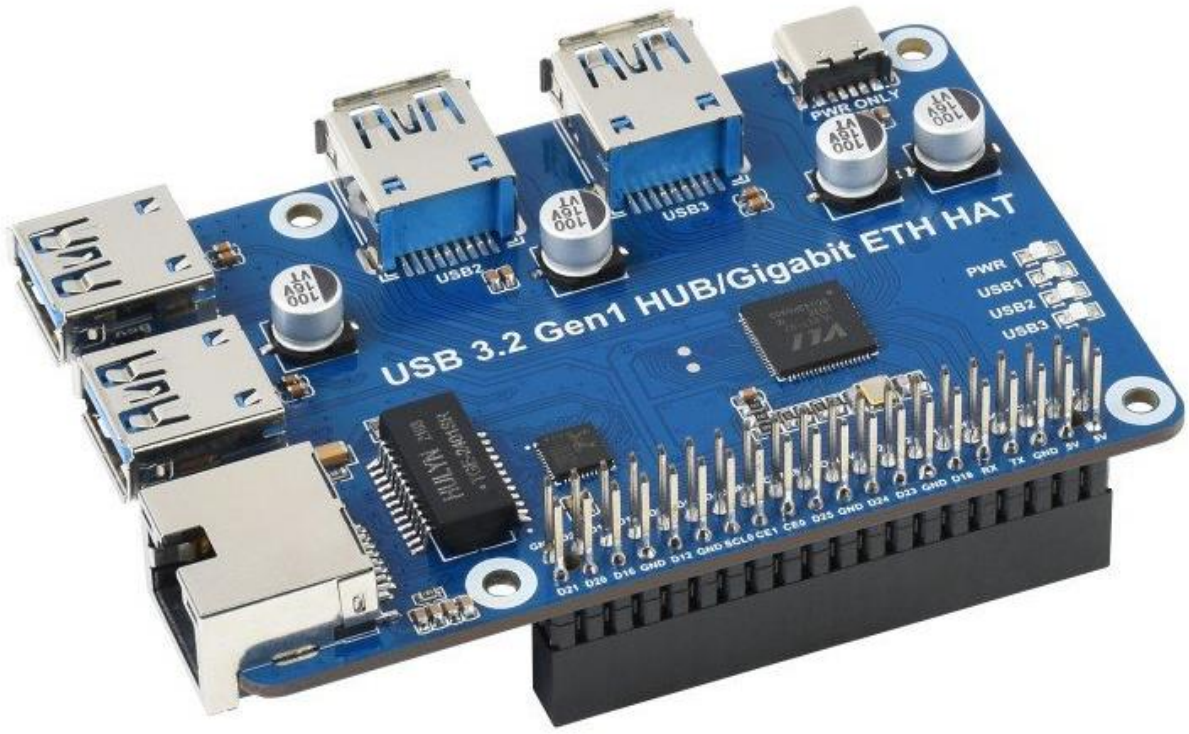
6. HDMI to VGA Adaptor

HDMI to VGA adaptor is required for connecting Raspberry Pi with computer monitor which do not have an HDMI port. Check for HDMI port for type, where Raspberry Pi 4 had a micro-HDMI port.

7. Raspberry Pi Hat

A Hat is an addon card which can be connected to IO port for specific interfacing application





Result

Raspberry Pi and various accessories are familiarized

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 15 Date: D D - M M - Y Y

Install suitable raspberry pi operating system (Raspberry Pi OS)

Keep Raspberry pi and the items mentioned in the previous experiment for installation

- Power adaptor (5V)
- MicroSD card (at least 8GB)
- Keyboard (wired or wireless)
- Mouse or another pointing device (could be built into the keyboard)
- Monitor with HDMI to VGA convertor or TV to connect to (via HDMI)
- HDMI cables

Note that the HDMI cable you need varies based on the Raspberry Pi you are using. Raspberry Pi 4 B has dual, micro-HDMI out ports so it requires micro-HDMI to HDMI cables or adapters. The Raspberry Pi Zero / Zero W have mini-HDMI and therefore need mini-HDMI to HDMI cables to connect to a display. All other Raspberry Pi models, including the 3 B, have standard HDMI ports and can use HDMI male to male cables to attach to your monitor or TV.

Powering the Pi

The Raspberry Pi 4 B and Raspberry Pi 400 (which is just a 4 B inside a keyboard) are powered via a USB Type-C port, which requires a charger that can output 5 volts and 3 amps. Most USB Type-C phone chargers don't have enough amps to get the job done, unless they have USB PD capability, but USB-C laptop chargers should all work. While it's unlikely to be a problem, note that Pi 4 models that were manufactured in 2019 or early 2020 have a bug which prevents them from charging over high-speed data cables that support USB 3.x 5 or 10 Gbps connections.

All other Raspberry Pi models, including the Raspberry Pi 3 B and Pi Zero / Zero W, get power via a micro-USB port.

An OS on a microSD Card

There are more than a dozen different OSes for Raspberry Pi, and there's even a way to run full Windows 10 on the Pi 3B. However, Raspberry Pi OS, a special version of Debian Linux that's optimized for the Pi, is the best platform for most use cases.

The Raspberry Pi has no internal storage, but instead boots off of a microSD memory card that you provide. Be sure to get a card that's at least 8GB, preferably 32GB or higher, and has class 10 speed, but you'll need some kind of card reader to write the OS to it from your PC.

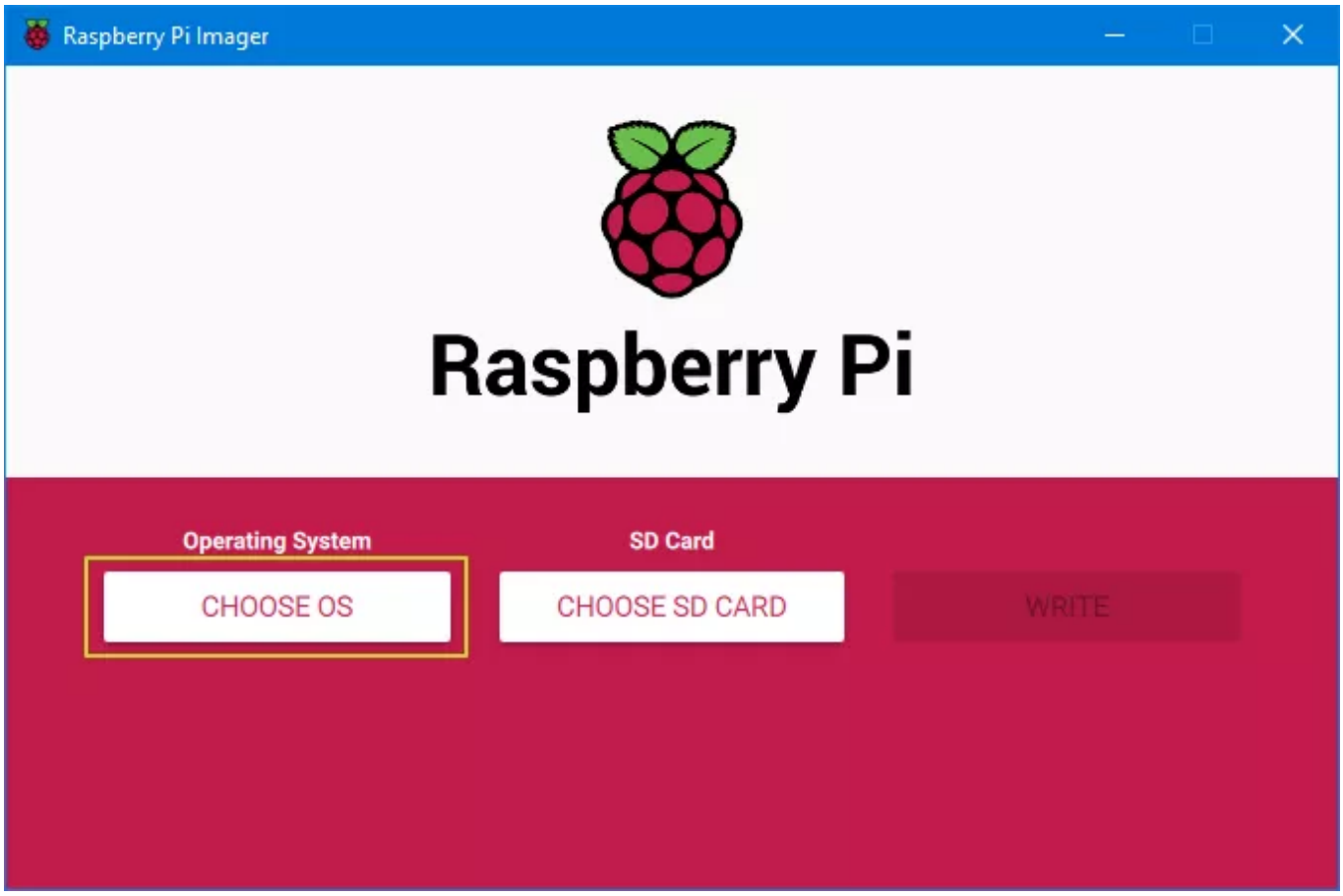
Downloading and Installing Raspberry Pi OS

Once you have all the components you need, use the following steps to set up your Raspberry Pi using a Windows, Mac or Linux-based PC.

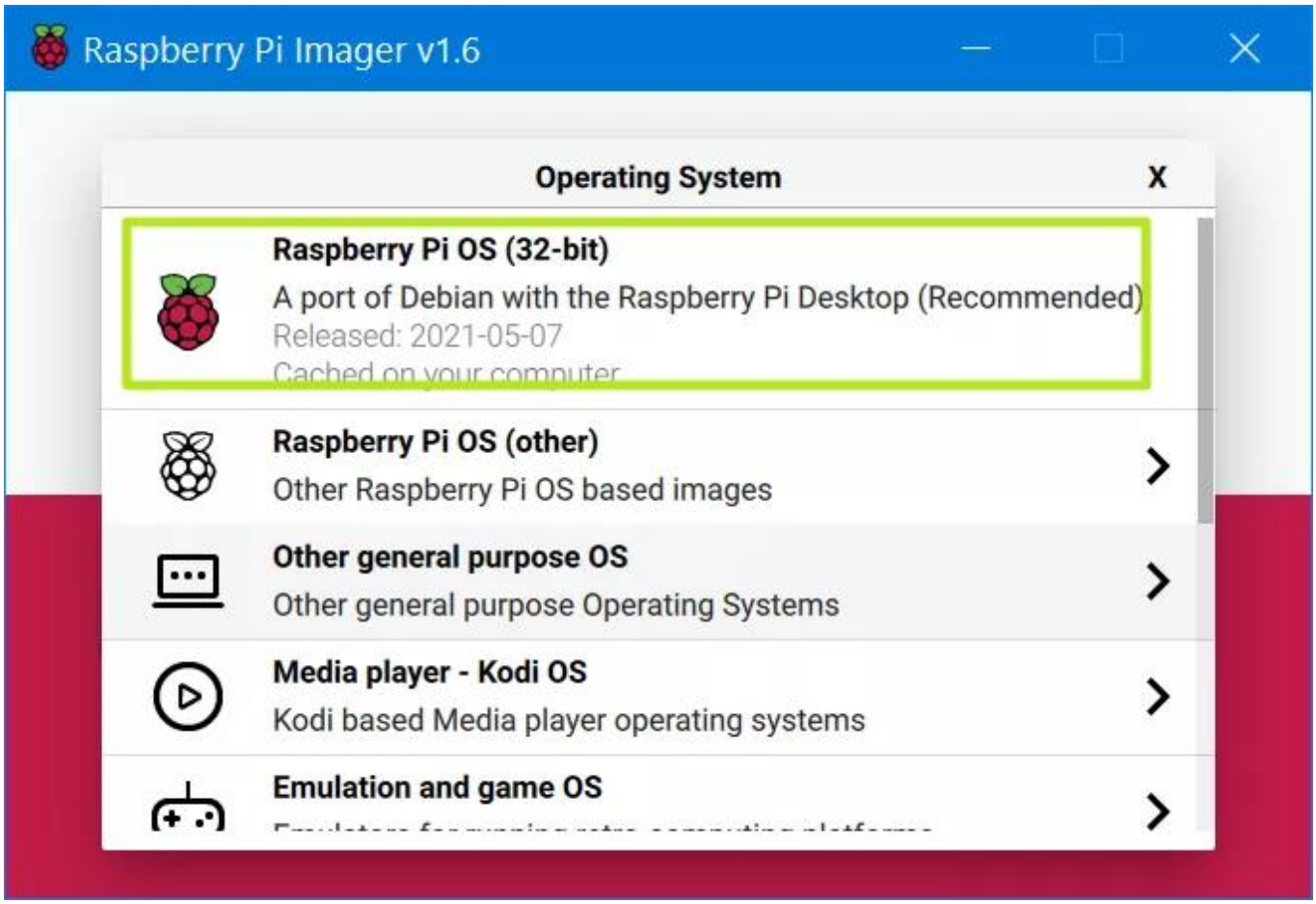
1. Insert a microSD card / reader into your computer.

2. Download and install the official Raspberry Pi Imager. Available for Windows, macOS or Linux, this app will both download and install the latest Raspberry Pi OS. There are other ways to do this, namely by downloading a Raspberry Pi OS image file and then using a third-party app to “burn it,” but the Imager makes it easier.

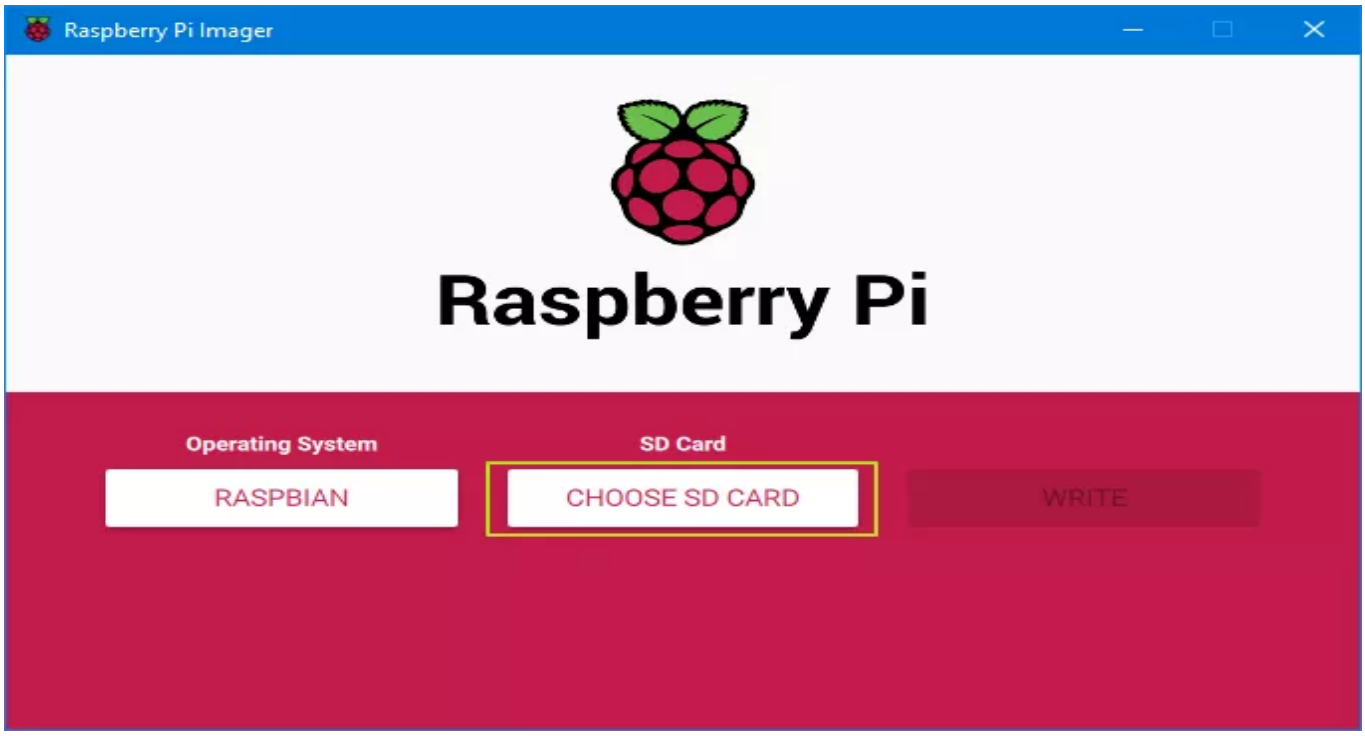
3. Click Choose OS and select Raspberry Pi OS (32-bit) from the OS menu (there are other choices, but for most uses, 32-bit is the best).



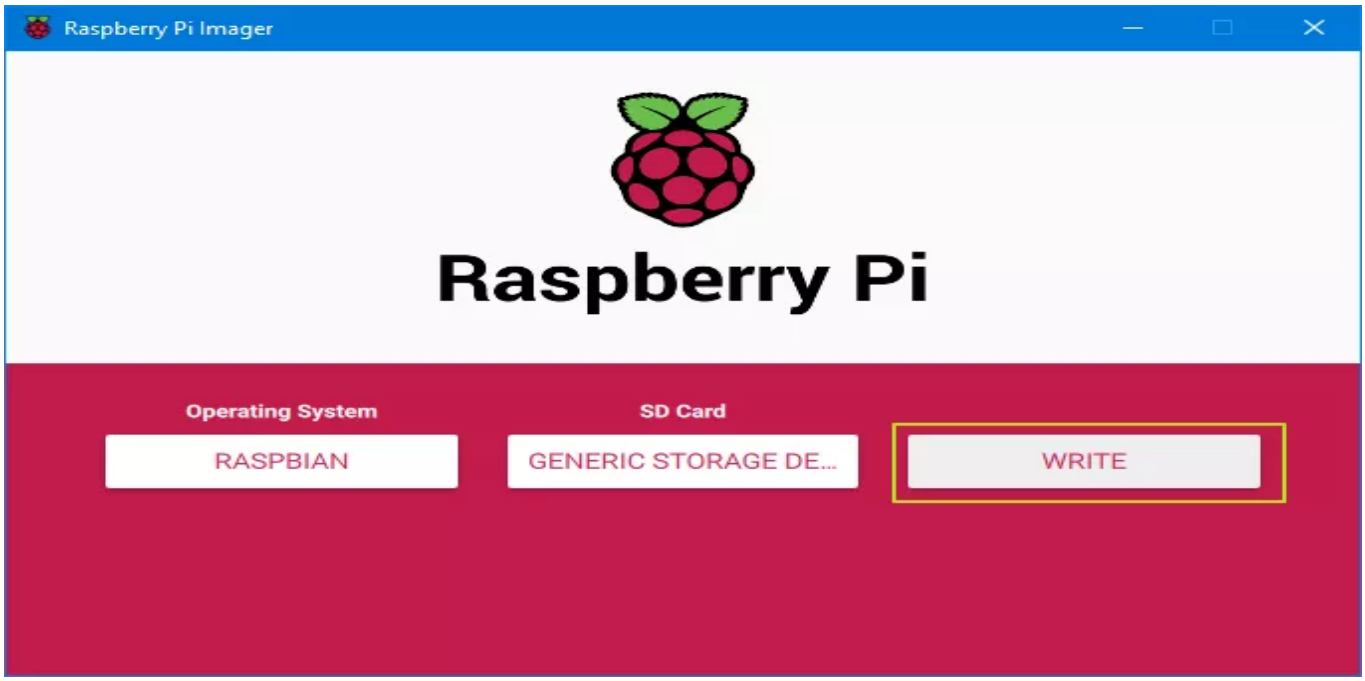
Click Choose OS in Raspberry Pi Imager (Image credit: Raspberry Pi Imager)



4. Click Choose SD card and pick the one you're using.



5. Click Write. The app will now take a few minutes to download the OS and write to your card.



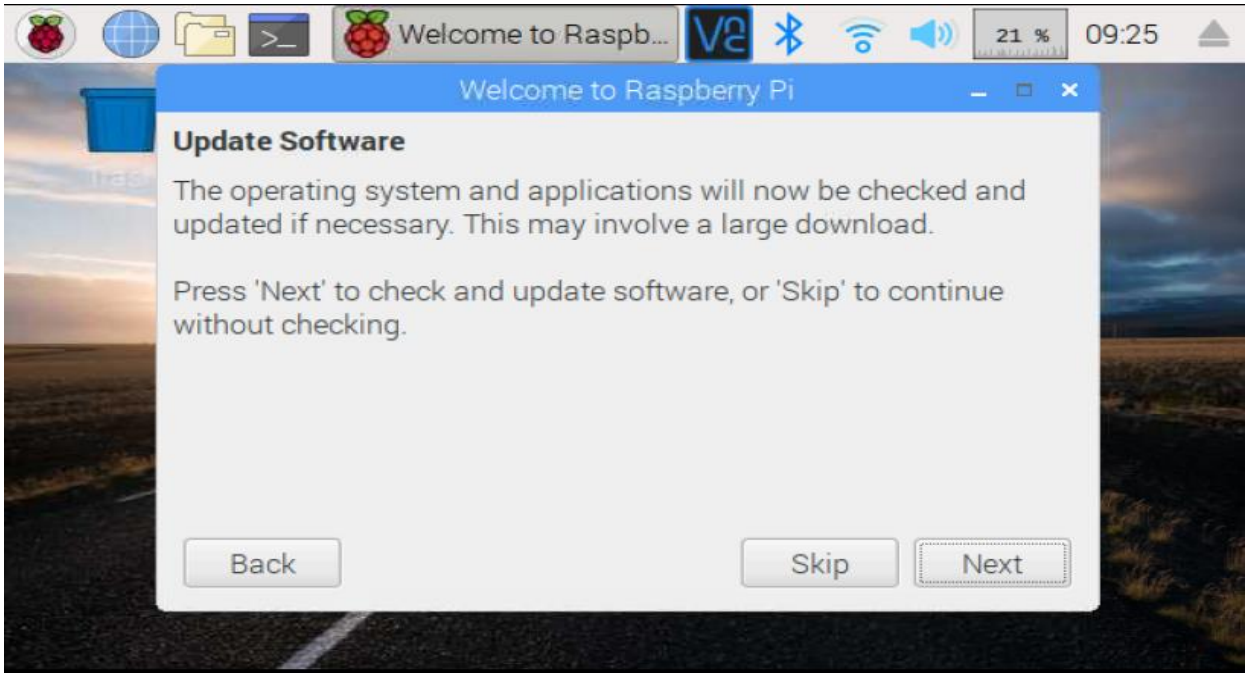
Configuring Raspberry Pi OS

If you are prompted for a username and password, the default username is "pi", and the password is "raspberry". If you're concerned about security, you'll want to change these. On first boot, you will probably be given a "Welcome to the Raspberry Pi" dialog box, which takes you through the process of choosing important settings.

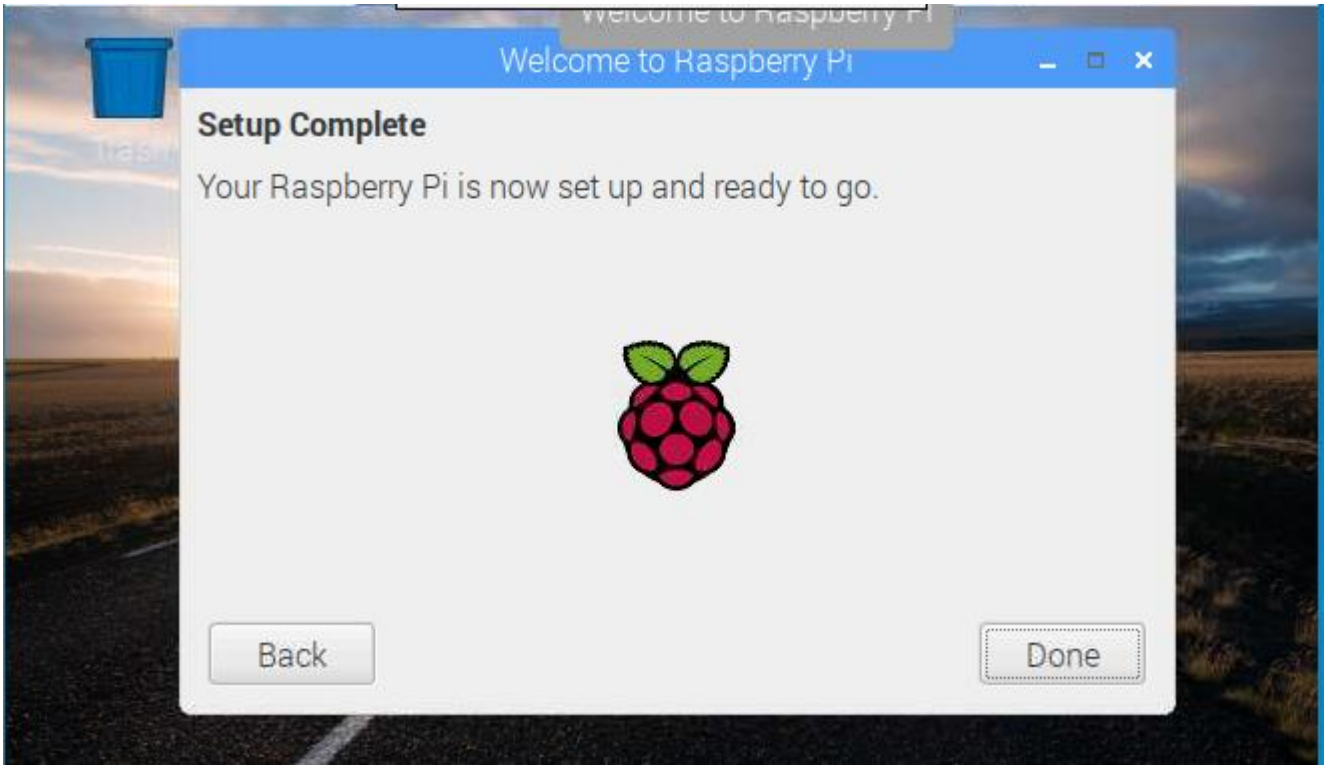
1. Click Next on the dialog box and then select your country, language and keyboard type.

2. Change your default password on the next screen or leave it blank for it remain as "raspberry."

4. Click Next when prompted to Update Software. This will only work when you are connected to the Internet, and it can take several minutes. If you are not connected to the Internet, click Skip.



5. Click Done or Reboot (if prompted to reboot).



Result

Raspberry Pi OS installation completed

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 15 Date: D D - M M - Y Y

Interface Raspberry Pi IO ports for input and output.

Objective: Blink LED connected to GPIO at 1 sec interval

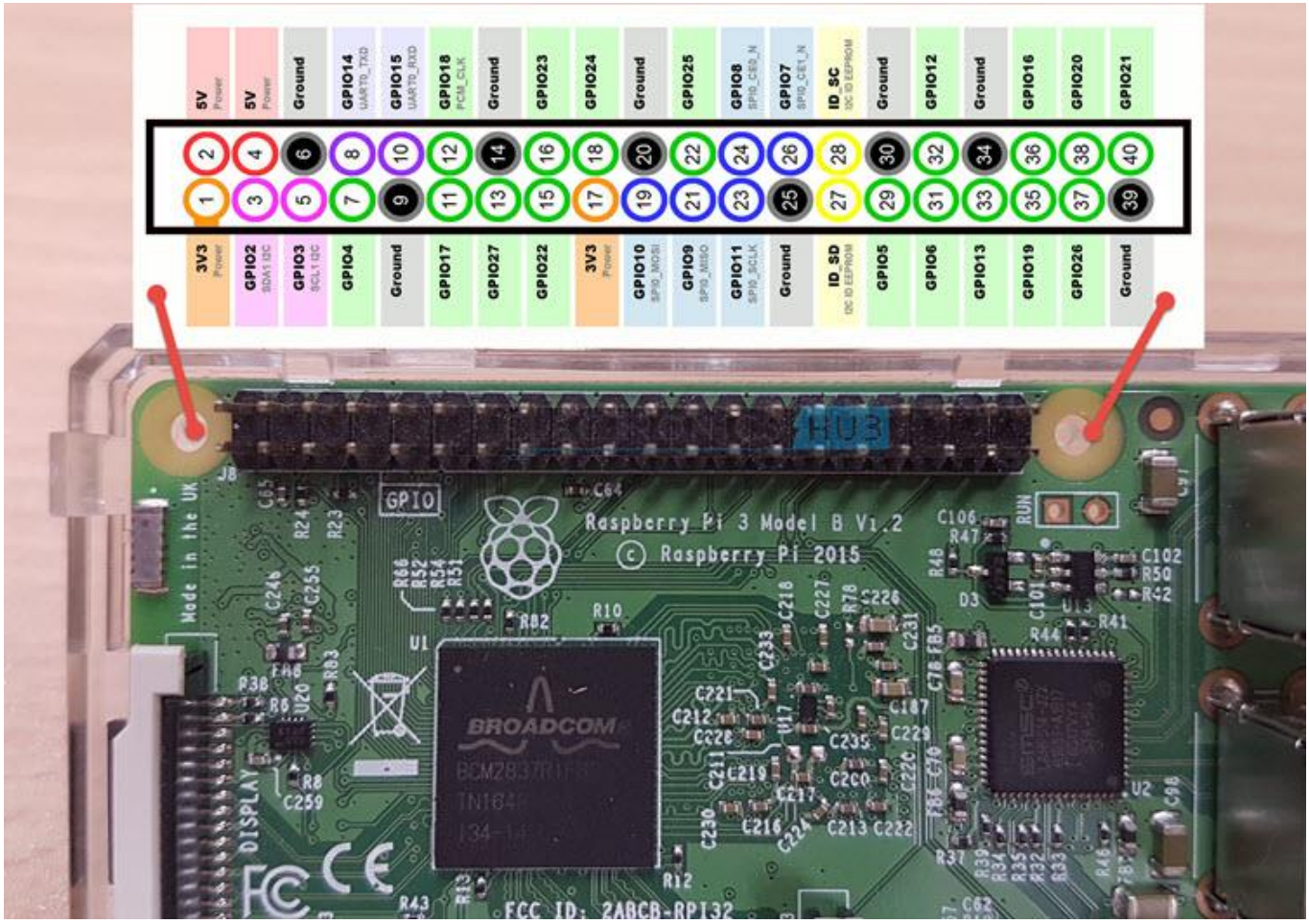
GPIO (General Purpose Input/Output) Pins of Raspberry Pi

Raspberry Pi has a powerful feature in the form of the General-Purpose Input / Output or simply GPIO Pins. GPIO Pins form the physical interface between the real world and the Raspberry Pi.

Different external components like LEDs, Motors, Sensors, Display, etc. are connected to the Raspberry Pi through these GPIO Pins. In our project, we are going to blink an LED using Raspberry Pi and hence the knowledge of all the GPIO Pins is important.

The latest revision of the Raspberry Pi series i.e. Raspberry Pi 4 Model B has 40 GPIO Pins. Out of these 40 GPIO pins few are power pins i.e. 3.3V Pins (2), 5V Pins (2) and GND (8). In the rest of the 28 pins, few are truly general purpose GPIO Pins while few pins have a dual function.

The following image shows the GPIO Pins of the Raspberry Pi 3 Model B board.



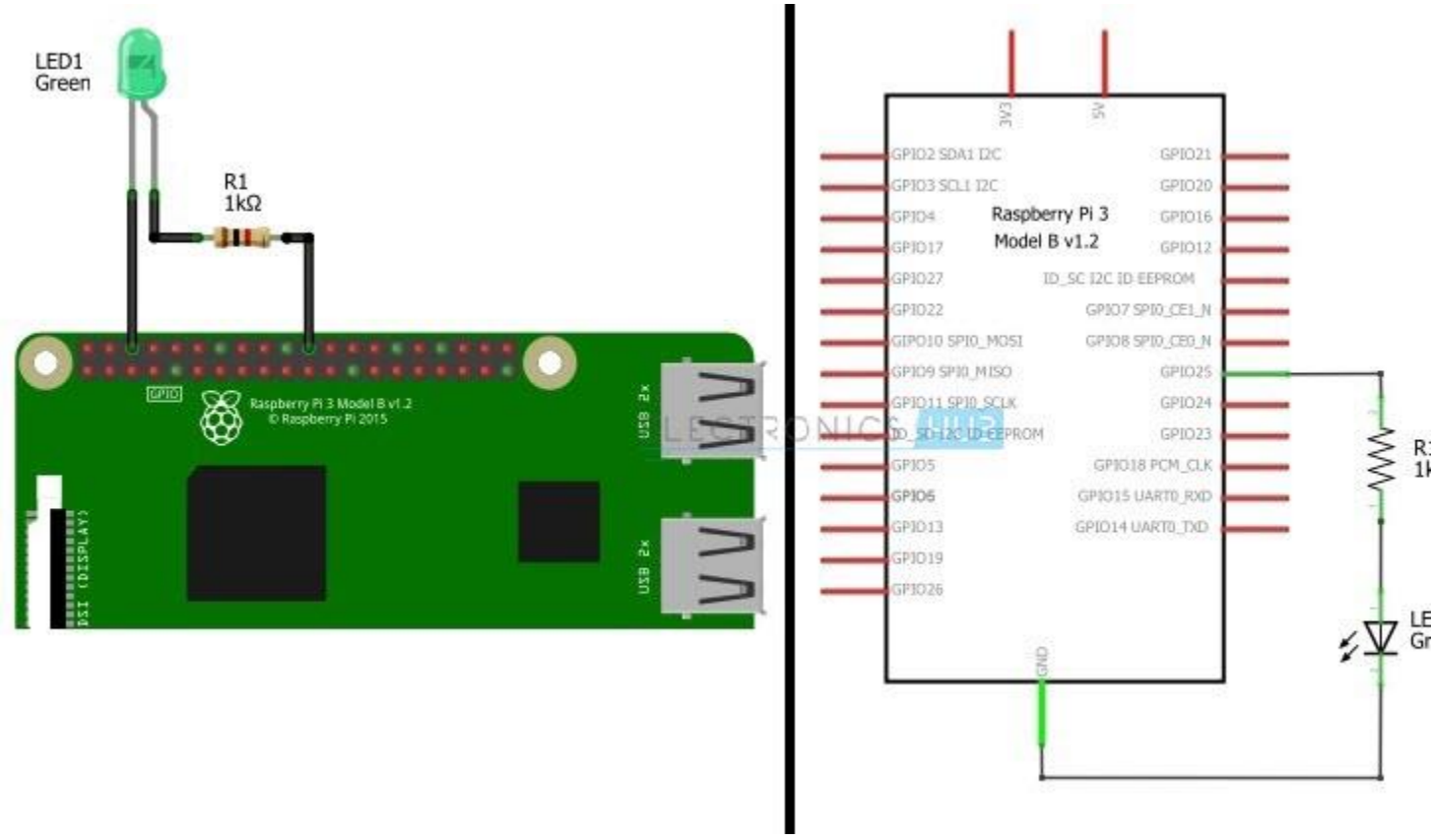
The numbers in the center (that are circled) are the Physical Pins of the Raspberry Pi. They are also called as Board Pins or Numbers. The GPIO Numbers (like Physical Pin 3 is GPIO2) are those which are seen by the Processor. This numbering is called as GPIO Numbering or BCM Numbering.

Circuit Diagram of Blinking LED with Raspberry Pi

In order to Blink an LED using Raspberry Pi, we need to first connect the LED to the Raspberry Pi. There are two ways you can connect your LED to the Raspberry Pi. I'll show both ways of connecting the LED.

Circuit 1

In the first circuit, the anode of the LED is connected to GPIO25 (Physical Pin 22) through a 1KΩ current limiting resistor. The cathode of the LED is connected to the GND Pin.



Code

```
import RPi.GPIO as GPIO      # Import Raspberry Pi GPIO library
from time import sleep       # Import the sleep function from the time module
GPIO.setwarnings(False)      # Ignore warning for now
GPIO.setmode(GPIO.BOARD)     # Use physical pin numbering
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW)
                             # Set pin 8 to be an output pin
                             # and set initial value to low (off)
while True:                  # Run forever
  GPIO.output(8, GPIO.HIGH)  # Turn on
  sleep(1)                   # Sleep for 1 second
  GPIO.output(8, GPIO.LOW)   # Turn off
  sleep(1)                   # Sleep for 1 second
```

Result

LED blink with Raspberry Pi is done.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

ANNEXURE

Soldering Practice - tools

Soldron soldering iron

Soldron Soldering iron stand

Nipper

Wire Stripper

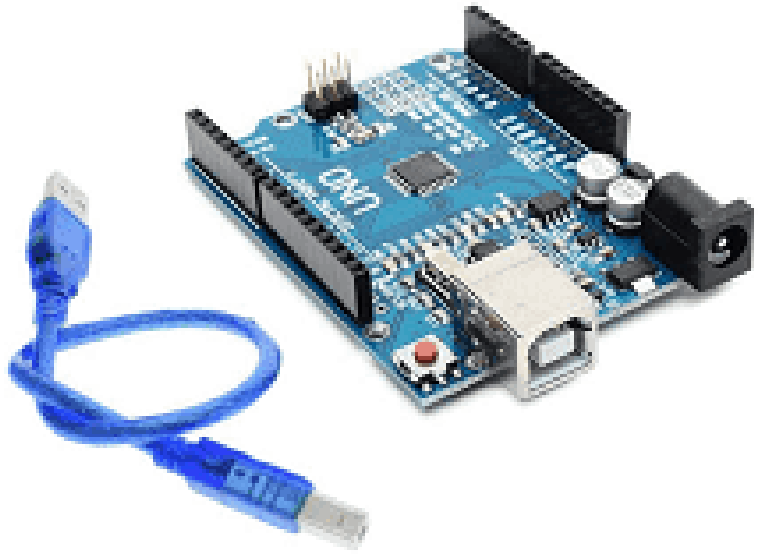
SS needle nose plier

Soldron de soldering pump

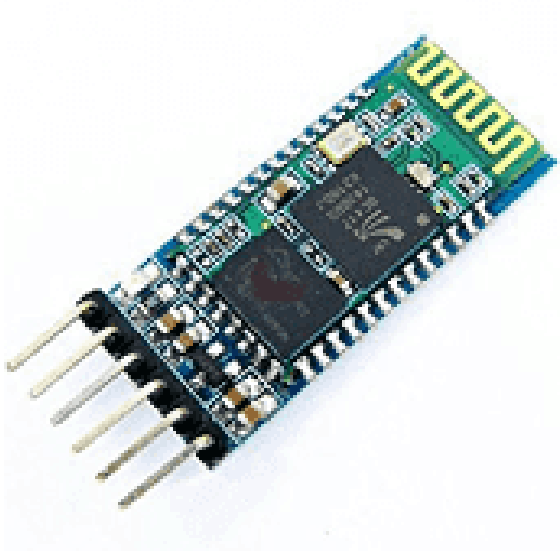

D-SOL-WIK
1 M.
2.5 mm
CK
KAPOOR

Arduino and compatibles

Arduino uno

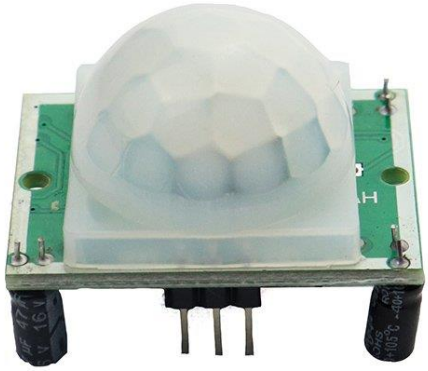


Blue tooth module (HC05)



Jumper wires (male – female)

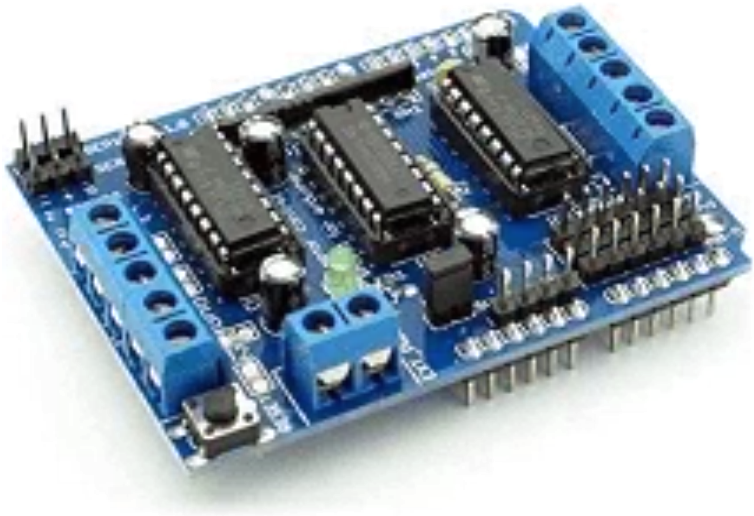
PIR Sensor



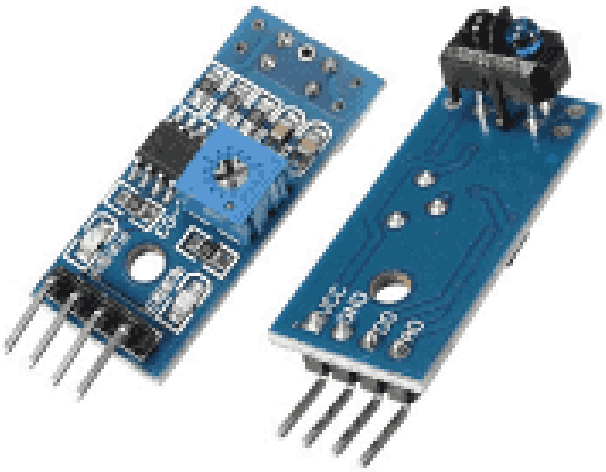
Relay Module

LCD Shield

Motor shield



Line follower sensor



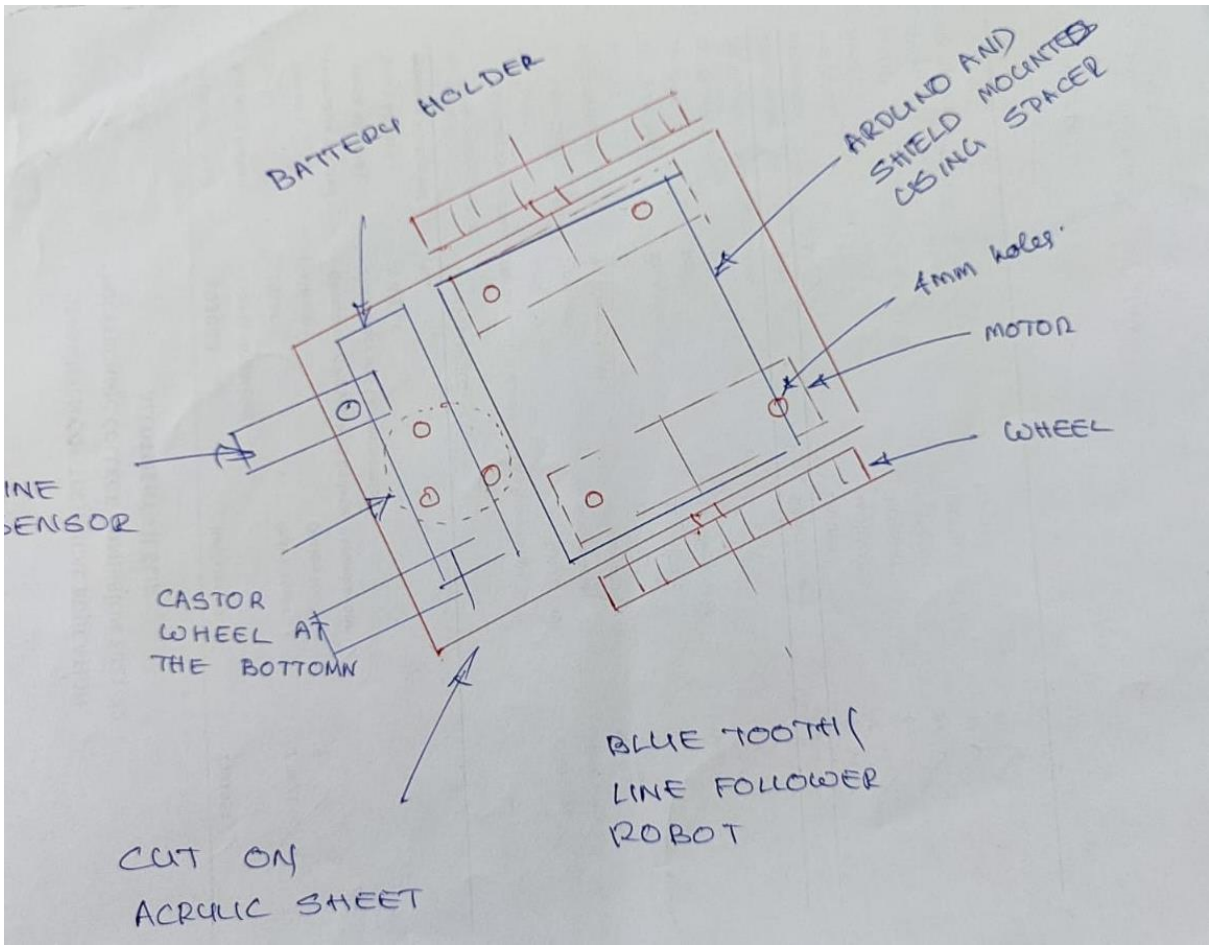
ROBO wheel

Castor wheel

Battery Holder (6V) for 6V motor

Spacer, screws, nut

Acrylic base for robot



Raspberry Pi and Compatibles

Raspberry pi 4

Micro SD card 32GB

Raspberry Pi case (Buy the same model)

HDMI to VGA convertor

Tools required and price as on March 2022

Those institutions which are provided with Robotics kit need not purchase any thing except HDMI to VGA convertor.

Sl	Name	Rate (Rs)
1	Soldron soldering iron	300
2	Soldron soldering iron stand	300
3	Nipper	100
4	Wire stripper	100
6	Stainless Steel needle nose plier	150
7	Soldron de soldering pump	200
8	60-40 solder, flux, solder wick	150

Arduino and compatibles

1.	Arduino Uno with USB cable	500
2.	HC05 blue tooth module	250
3.	Jumper wire (male – male, male - female, female – female)	2 each
4.	PIR sensor	100
5.	Relay module (single)	75
6.	LCD key shield	250
7.	For robot	
a.	Motor shield	150
b.	Line follower sensor (2 for one robot)	50 each
c.	Geared dc motor 3 - 6V 100 rpm (2Nos)	75 each
d.	Robo wheel (2Nos)	30 each
e.	Battery holder	50
f.	Hardware (M3 screws, nut, spacer)	4 spacer, 10 screws
g.	Acrylic base for robot	150

(Can be fabricated locally or use facility available at FabLab established at polytechnic colleges by ASAP. Use laser cutter for the purpose)

Raspberry Pi

1. Raspberry pi 4 1GB RAM or above	4300
2. Micro SD card 32GB	400
3. Adaptor (mobile charger)	use available – 2A or above
4. Raspberry Pi case	200
5. HDMI to VGA convertor	250

NOTES

This page is intentionally kept blank to preserve page format.

GOVERNMENT POLYTECHNIC COLLEGE, PERUMBAVOOR
Koovappady P O, Perumbavoor, Ernakulam – 683544
www.gptcperumbavoor.ac.in