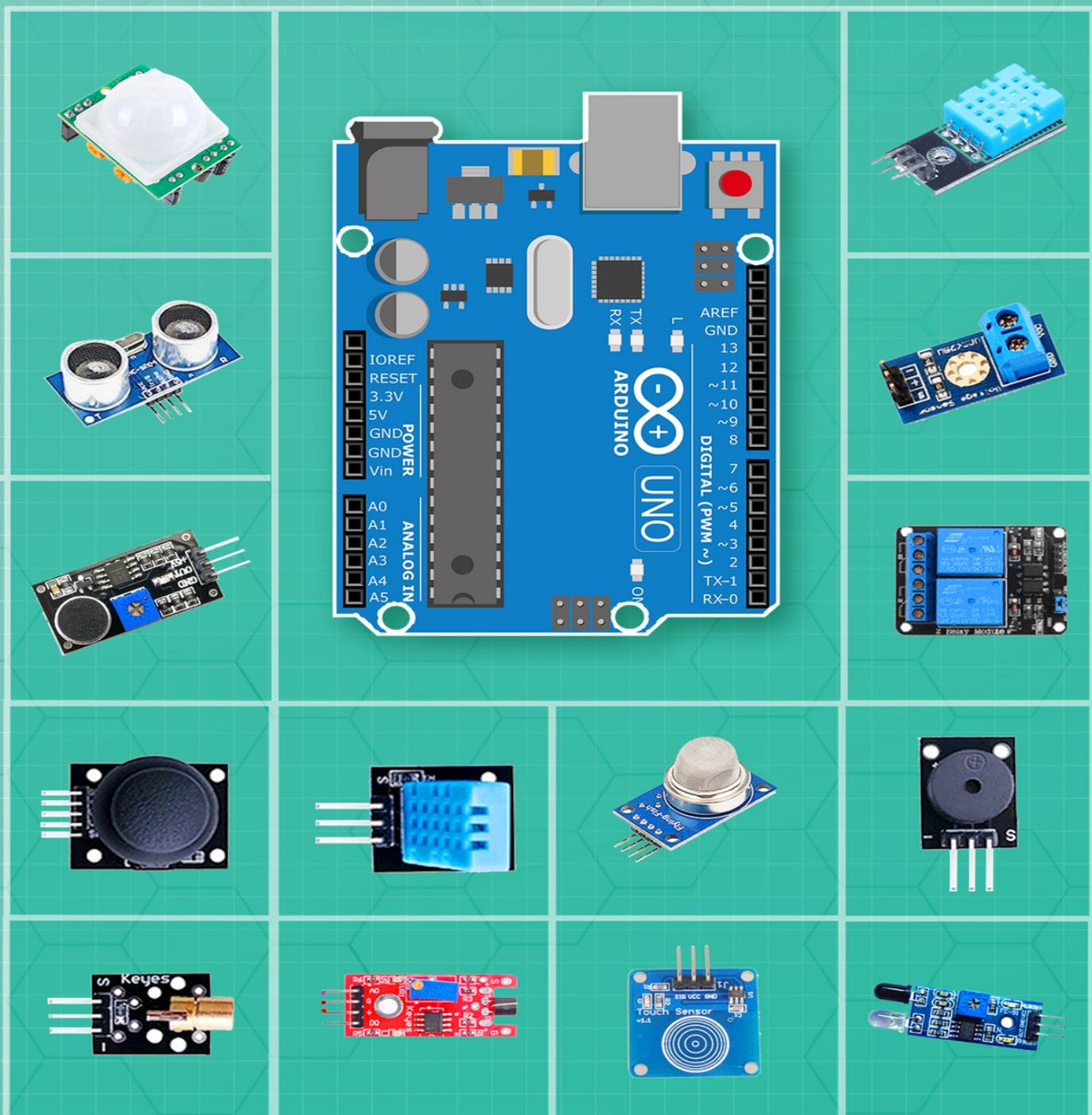


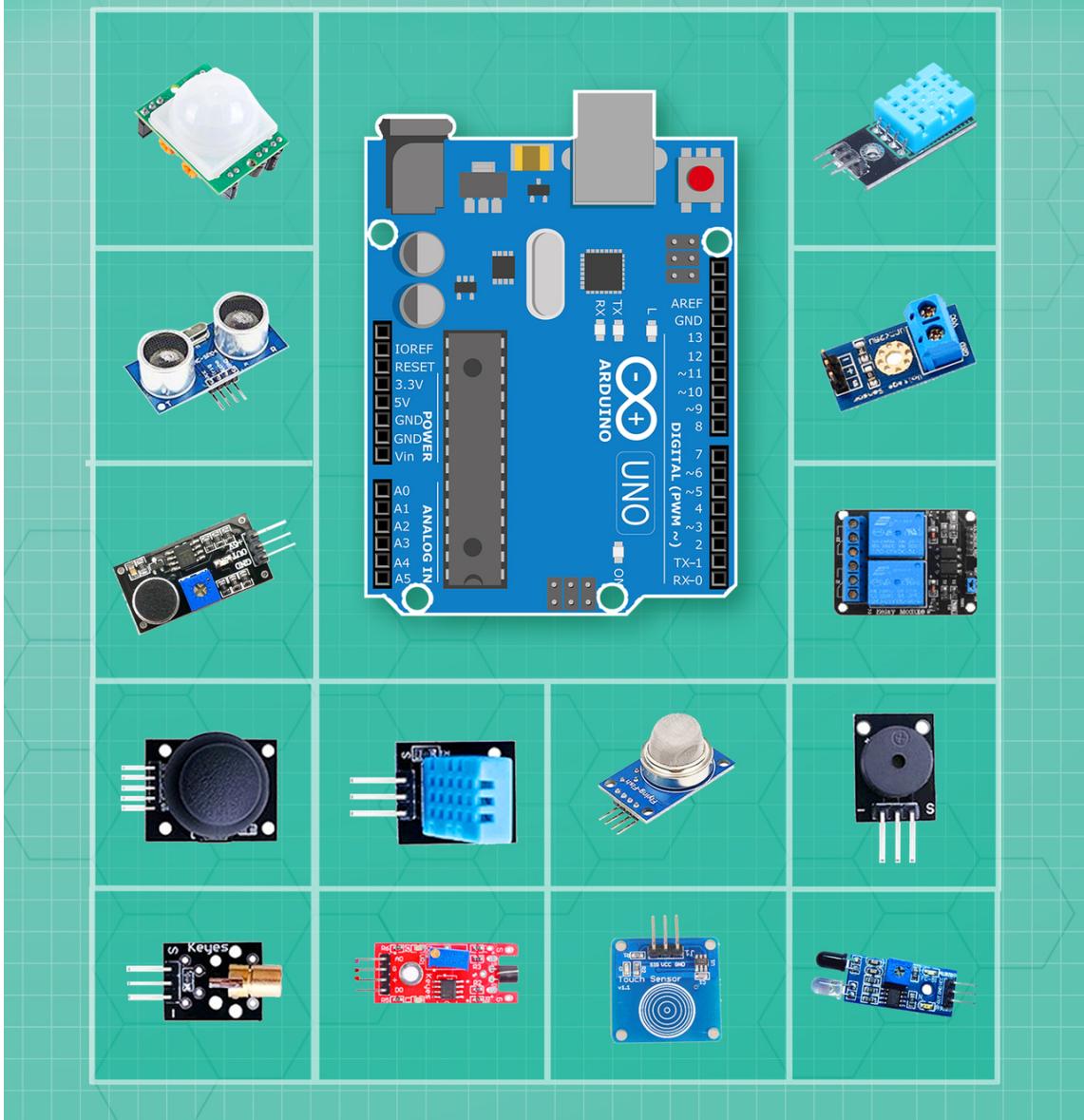
# LEARN ARDUINO SENSORS COMPLETE HAND GUIDE BEGINNER TO CORE ADVANCE

Example Sensor Code, Specification, Dimensions,  
Connecting method



# LEARN ARDUINO SENSORS COMPLETE HAND GUIDE BEGINNER TO CORE ADVANCE

Example Sensor Code, Specification, Dimensions,  
Connecting method



# **LEARN ARDUINO SENSORS**

## **COMPLETE HAND GUIDE**

## **BEGINNER TO CORE ADVANCE**

Example Sensor Code, Specification, Dimensions,  
Connecting method

BY  
JANANI SATHISH

## **CONTENTS**

[WHAT IS ARDUINO](#)

[THE HOLE ON BOARD](#)

[WHAT IS A MICROCONTROLLER](#)

[WHAT IS ARDUINO RASPBERRY PI AND PIC  
MICROCONTROLLER](#)

[KY-001 TEMPERATURE SENSOR MODULE](#)

[KY-002 VIBRATION SWITCH MODULE](#)

[KY-003 HALL MAGNETIC SENSOR MODULE](#)  
[KY-004 KEY SWITCH MODULE](#)  
[KY-005 INFRARED EMISSION SENSOR MODULE](#)  
[KY-006 SMALL PASSIVE BUZZER MODULE](#)  
[KY-008 LASER SENSOR MODULE](#)  
[KY-009 3-COLOUR FULL-COLOUR LED SMD MODULES](#)  
[KY- 010 OPTICAL BROKEN MODULE](#)  
[KY-011 2-COLOUR LED MODULE](#)  
[KY-012 ACTIVE BUZZER MODULE](#)  
[KY-013 TEMPERATURE SENSOR MODULE](#)  
[KY-015 TEMPERATURE AND HUMIDITY SENSOR MODULE](#)  
[KY-016 3-COLOUR LED MODULE](#)  
[KY-017 MERCURY OPEN OPTICAL MODULE](#)  
[KY-018 PHOTO RESISTOR MODULE](#)  
[KY-019 5V RELAY MODULE](#)  
[KY-020 TILT SWITCH MODULE](#)  
[KY-021 MINI MAGNETIC REED MODULES](#)  
[KY-022 INFRARED SENSOR RECEIVER MODULE](#)  
[KY-023 XY-AXIS JOYSTICK MODULE](#)  
[KY-024 LINEAR MAGNETIC HALL SENSORS](#)  
[KY-025 REED MODULE](#)  
[KY-026 FLAME SENSOR MODULE](#)  
[KY-027 MAGIC LIGHT CUP MODULE](#)  
[KY-028 TEMPERATURE SENSOR MODULE](#)  
[KY-029 YIN YI 2-COLOUR LED MODULE 3MM](#)  
[KY-031 KNOCK SENSOR MODULE](#)  
[KY-032 OBSTACLE AVOIDANCE SENSOR MODULE](#)  
[KY-033 HUNT SENSOR MODULE](#)  
[KY-0347 COLOUR LED FLASH-MODULE](#)  
[KY-035 BIHOR MAGNETIC SENSOR MODULE](#)

[KY-036 METAL-TOUCH SENSOR MODULE](#)  
[KY-037 MICROPHONE SENSOR MODULE](#)  
[KY-038 MICROPHONE SOUND SENSOR MODULE](#)  
[KY-039 HEARTBEAT SENSOR MODULE](#)  
[KY-040 ROTARY ENCODER](#)  
[RASPBERRY PI CONNECTIONS](#)  
[RASPBERRY PI CONFIGURATION AND CODE](#)  
[ARDUINO AND RASPBERRY PI CONNECTION](#)  
[HOW ARDUINO THINKS](#)  
[ARDUINO SYNTAX WITH EXAMPLE](#)  
[ASSEMBLING A CIRCUIT](#)  
[ARDUINO SOFTWARE FOR THE FIRST TIMERS](#)  
[ARDUINO SYNTAX](#)  
[FINISHING CONFIGURATION](#)  
[THE BREADBOARD](#)  
[IF A CONDITIONAL STATEMENT](#)  
[FOR LOOP](#)  
[WHILE LOOP](#)  
[USE AN ARRAYS](#)  
[CONTRAST BETWEEN INPUT AND OUTPUT](#)  
[LED MULTIPLE](#)  
[BUTTON INPUT AND LED](#)  
[INPUTS BUTTONS](#)  
[IR SENSOR INPUT ANALOG](#)  
[POTENTIOMETER INPUT ANALOG](#)  
[ANALOGUE INPUT WITH EXAMPLE](#)  
[LIQUID CRYSTAL DISPLAY](#)  
[QUICK SUMMARY OF THE MAIN DIFFERENCES](#)  
[HARDWARE POWER AND](#)  
[CONNECTIVITY](#)

COMPARISON OF SOFTWARE  
EXPAND THE NEXT CAPABLITIES  
BASIC PARTS REQUIREMENTS  
THE PART NEEDED TO COMPLETE THE WORK  
POWER SUPPLY WIRE  
ARDUINO IS READY  
YOUR BOOTLOADING CHIPS OPTIONAL  
COMPLETE AND BURN BOOTLOADER  
ATMEL ATMEGA 328 DETAILS  
DIGITAL AND SERIAL PORT  
MOTORS TYPES  
SIMPLE DC MOTOR  
SERVO MOTOR  
STEPPER MOTOR  
RUNNING A DC MOTOR  
RUNNING A SERVO MOTOR  
ARDUINO BLUETOOTH  
INTRODUCTION AND HISTORY OF WIRELESS POWER  
HOW WIRELESS POWER WORKS  
ADVANTAGES AND DISADVANTAGES  
SCOPE OF WITRICITY AND WITRICITY APPLICATIONS  
HOW DOES IT WORK  
ANDROID APPLICATION  
SCHEMATIC  
CODE  
TESTING CODE  
EARN WITH ARDUINO

# WHAT IS ARDUINO

You will get my thoughts regarding Arduino nuts and bolts. Furthermore, in the second case of how programming should be possible with Arduino. Before we go further, I need to help you to remember a portion of the things that Arduino is famous with today, it is utilized in four distinctive IoT executions all through the world. Arduino gadgets are modest, they decrease utilization and which is the reason they are extremely well known for use in the web usage of things. In the primary module, we have seen various things, we have perceived the idea of the Internet of Things, the fundamental idea, the general component of the pilot merchandise. We have seen that there are different sorts of sensors, different kinds of sensors that might be different sorts of actuators, the standards behind different kinds of feeling different kinds of activation. We have seen that there are different sorts of organizations that are feasible for us. Creation for use in IoT different kinds of specialized gadgets can be utilized to convey in the web things. Getting it, how might we utilize these ideas to fabricate genuine web in savvy home situations at home to improve me from them, you know, the everyday errands we do at home or a keen home, brilliant city situation, as In shrewd medical clinics, keen urban communities, you know, keen transportation, vehicles associated, we need to take it diverse IoT gadget help. What's more, one of the mainstream is Arduino. We know whether we need to utilize the development of the web. You need to purchase this which is very modest.



And afterwards, you need to program this, and this is the thing that I will show you in this specific course. With me, I have Mr The not fixed Mukherjee will dominate and Mukherjee will take you through your hands - Arduino programming to from the start of the rudiments for it will be very modern the idea of Arduino programming. I need to show you first how the Arduino gadget looks. This is Arduino Uno. Arduino Uno, Arduino has an alternate variation, they all have the distinctions you know, contrasts in Andon determinations. This is Arduino Uno. Furthermore, this gadget should be modified. It should be customized, as you can see here, the size is little and can be extremely incorporated with this web. You know, when you attempt to apply the Internet of Things. It very well may be actualized previously. This gadget is really, you know the diverse SERR that you know learning in this specific course. This diverse SENR can be introduced on this gadget, various actuators can be introduced on this gadget and SENRS, distinctive SENRs and different actuators after introducing information got from a similar depiction. It tends to be shipped off the correspondence unit for this situation that will be examined by the Rupee and how this information can be scattered and can be sent for additional investigation for Andon stockpiling. I will take you through Arduino's fundamental highlights in this talk. First and foremost, as you have heard, Arduino has become

extremely famous today. Most importantly, the fundamental Rean, this is an open programmable program board

## Features of Arduino

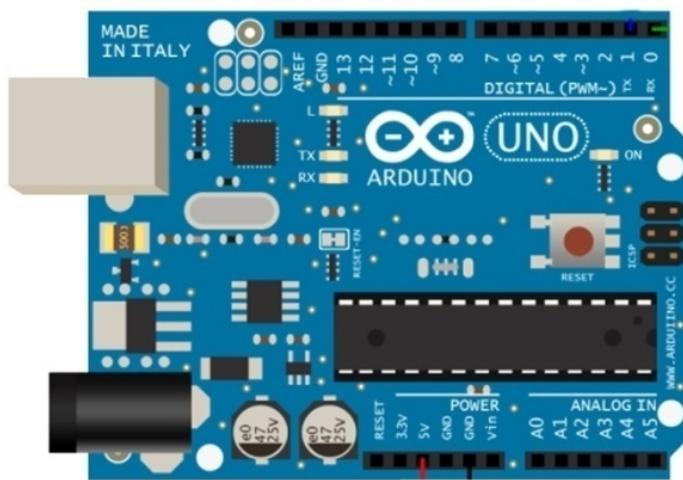
- Open source based electronic programmable board (micro controller)and software(IDE)
- Accepts analog and digital signals as input and gives desired output
- No extra hardware required to load a program into the controller board

With a default microcontroller, and this product ID and ID programming will help you change the microcontroller conduct as per your requirements. It gets simple and computerized signals that can be given as information and it will give chiefly advanced yield. No extra equipment is expected to stack the program to the regulator choice. Individuals who have worked with the 8051 arrangement, microcontrollers, 8085 chip, certainly recollect that you need extra software engineers to program processor sheets and there are numerous ICS interfacing and every one of these things are not required with Arduino-based frameworks.

## Types of Arduino Board

- Arduino boards based on ATMEGA328 microcontroller
- Arduino boards based on ATMEGA32u4 microcontroller
- Arduino boards based on ATMEGA2560 microcontroller
- Arduino boards based on AT91SAM3X8E microcontroller

To begin, there are a few fundamental varieties of Arduino sheets. They have an Atmega 328-based microcontroller, they have an Atmega 32 arrangement microcontroller, they have an Atmega 2560 arrangement microcontroller and afterwards there is an Atmega 91 Sam, three x 80 arrangement microcontrollers.

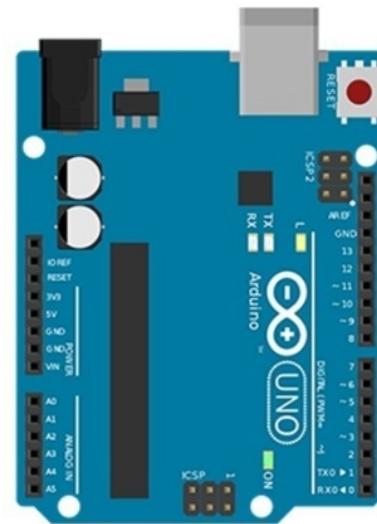


This is me from the centre microcontroller because you can check whether you centre around this IC this Arduino board, we utilize the UNO board to give a showing. This is the IC chip and fundamentally,

all the others are voltage converters or IC interfacing required for the info yield work with this Atmega arrangement chip.

## Arduino UNO

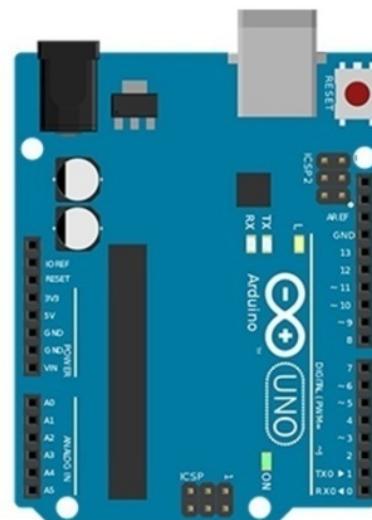
Feature	Value
Operating Voltage	5V
Clock Speed	16MHz
Digital I/O	14
Analog Input	6
PWM	6
UART	1
Interface	USB via ATMega16U2



This is me from the centre microcontroller because you can check whether you centre around this IC this Arduino board, we utilize the UNO board to give an exhibit. This is the IC chip and essentially, all the others are voltage converters or IC interfacing required for the information yield work with this Atmega arrangement chip.

## Board Details

- Power Supply: USB or power barrel jack
- Voltage Regulator
- LED Power Indicator
- Tx-Rx LED Indicator
- Output power, Ground
- Analog Input Pins
- Digital I/O Pins



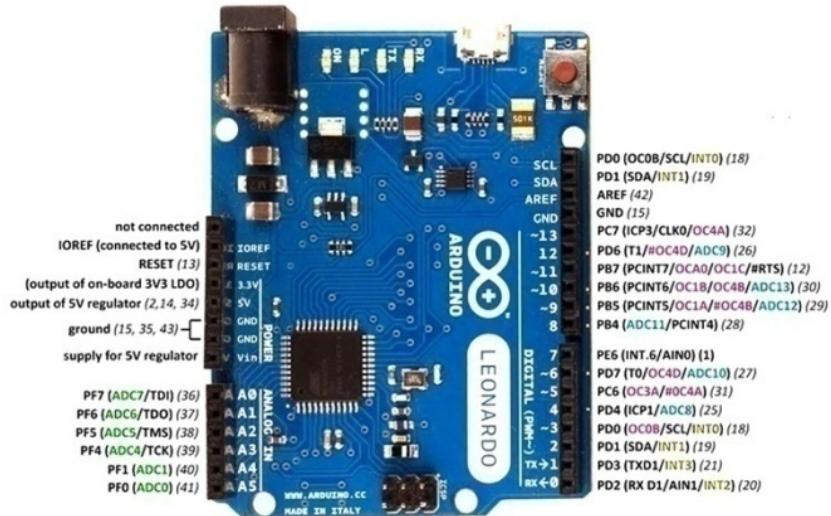
The more subtleties as should be obvious, from this picture, you have a USB connector that you associate with your Atmega boat to your PC. The best thing about this is the interfacing is simple. You can interface your Arduino framework based mode to a Windows-based PC or Macintosh or a Windows or Linux-based framework. This is a USB connector. At that point, you have a force connector to control the gadget in free mode. If not, on the off chance that you associate it to a PC, it draws power from the actual PC. Furthermore, this is a simple reference pin. You have 14 computerized pins, which can be utilized as information and yield. As you can see beginning from zero to 13. It is 14 information yield pins. What's more, here you have six simple pins, zero to five that can get simple information. Furthermore, this is me from the force connector. You have a 3. 3 volt five-volt association and Andon soil. Just me from the fundamental Arduino part.

## Arduino IDE

- Arduino IDE is an open source software that is used to program the Arduino controller board
- Based on variations of the C and C++ programming language
- It can be downloaded from Arduino's official website and installed into PC

Presently, the Arduino ID is SOFTWARE Open-URCE. The Arduino framework itself is an open URCE framework accessible with equipment details. You can really on the off chance that you have manufacture offices, you can make your own Arduino gadget. This ID no ID is an open URCE when it is utilized to program Arduino sheets, in light of varieties of C and C ++ programming dialects, and can be downloaded uninhibitedly from Arduino's true site.

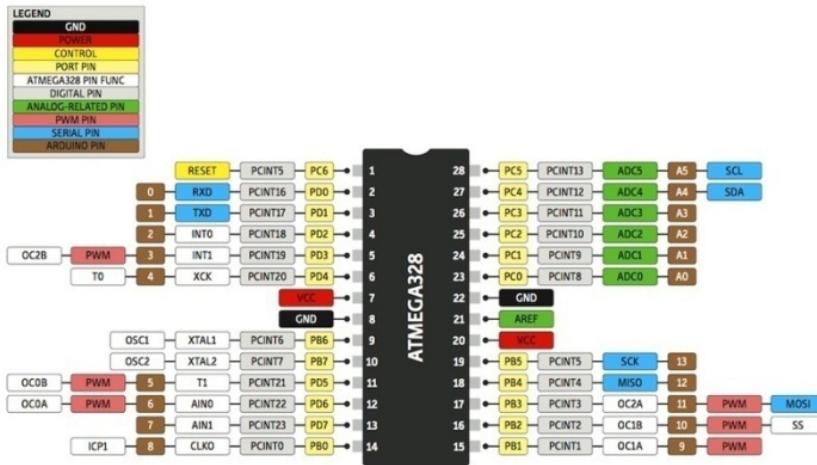
# THE HOLE ON BOARD



We will discuss openings in the board seeing in the past picture opening lines are running here and there on the two sides of the board. You will stick things in the openings. There are two different openings that you may finish and that is both before the board. Since you can see two spots where you can connect things as a rule you will utilize the left to append a USB attachment and in this way to your PC and you will utilize the correct one. To control your board when a long way from your PC utilizing a divider attachment or battery, the left fitting will be utilized to transfer code to the slate. This can be utilized to control whatever equipment you join to a little opening on the left and right that is Arduino. Much obliged to you for watching this exercise. If you have questions, if it's not too much trouble, ask Q and on the boat. Presently you understand what Arduino is. Furthermore, what is the following stage to show you how it occurred? Since there is a creating improvement local area for Arduino on the Internet. You could discover the code and circuit outline for whatever project you need to make. Nonetheless, it expects you to have the option to peruse the code and comprehend it to be examined later in this course. Much thanks to you for

watching this exercise. If you have questions, if it's not too much trouble, ask Q and on the boat.

## **WHAT IS A MICROCONTROLLER**



Miniature supernatural occurrence about the microcontroller. Above all else, we need to comprehend the contrast between microcontrollers and microchips. Well if we say the microchip as a PC cerebrum, you can say that the microcontroller capacities as the mind and muscles for PCs. As such, the microcontroller is a framework on the chip. This implies that the chip has four kilobytes of smash. It has a hard circle, there are 32 kilobytes, and the processor comprises eight megahertz. Indeed, this framework and detail sound extremely low, yet they can run practically all projects required in numerous thighs simply little memory thighs. Great about microcontrollers in their design. In three to eight, we have 28 pins. You can see here, thighs are the actual number of pins and in pink, you can see that the number allocated in Harding, you can see that the pin four, four on an actual pin is a two pin in Arduino. On the whole parts, we will manage Arduino Pin in Arduino. OK, presently you can see here that the eight microcontrollers, what will be 10 will be set as a memory design or engineering Z there are different sorts of microcontrollers who have somewhat 16-cycle engineering and 64-digit, yet here we will work with eight-Bit.

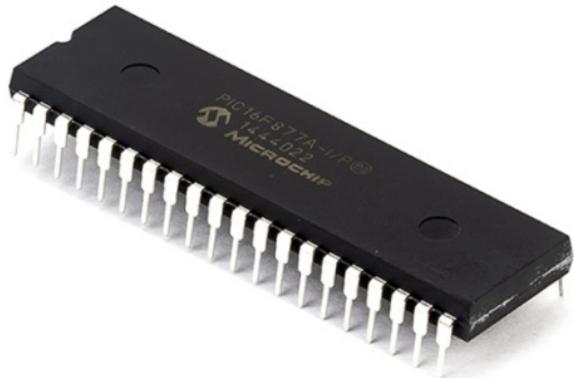
## WHAT IS ARDUINO RASPBERRY PI AND PIC MICROCONTROLLER

We will talk about art definition, we know Raspberry By and Big Mike Room for

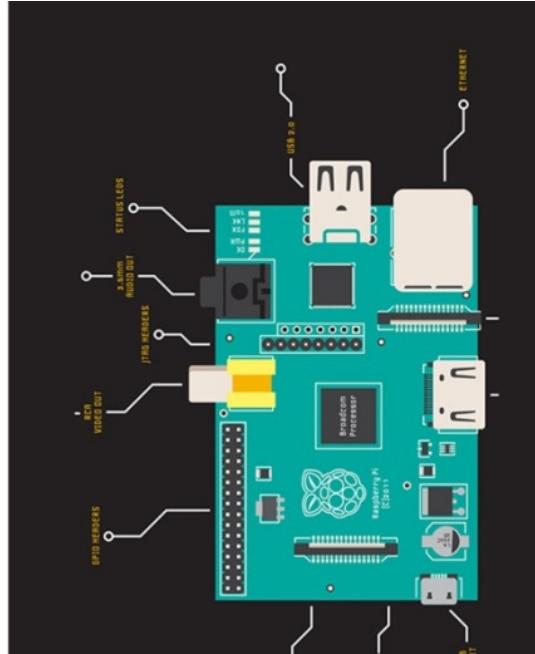
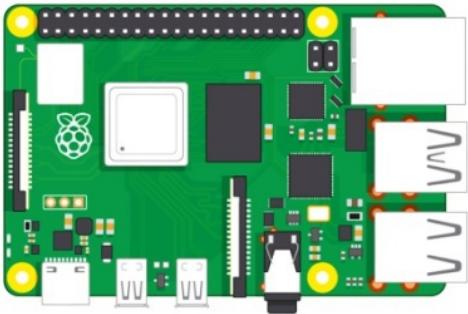


many.

We will see this third quick regulator. So how about we start with Arduino. What is a workmanship we know and we know is a bunch of improvement sheets that accompanied equipment libraries and imaginative programming? This implies you can purchase Arduino sheets and begin building up your undertaking and moment sheets worked around the AVR microcontroller are ribose fundamental programming on the board and accessible with the expectation of complimentary coding for Arduino programming for administrator investigates my control. The lone contrast is that the program for Arduino was written in its language called Arduino programming language. This language is equivalent to any C programming language except that everything is prepared to utilize the program to peruse and compose and. A little memory has been composed by somebody. We just need to call some unacceptable capacities to attempt to add the location and incentive as you find in this line. Sambo if you need to blend with you is a sequential bar. You don't have to introduce it on a register or board, simply call the sequential. At that point attempt word hi so you can print it in the sequential port. This is an illustration of Arduino's board. It's omega monstrous as should be obvious. It has the best processor. So how about we proceed. What is my extraordinary control that is typically spoken as a photograph is a microcontroller family made by CPU innovation that comes from 16 50 beginning on the floor by instrument model Microelectronics Division Model VIC has memory or space or field of perusing or even on the program stockpiling? Some with a wide vision for eliminating memory.

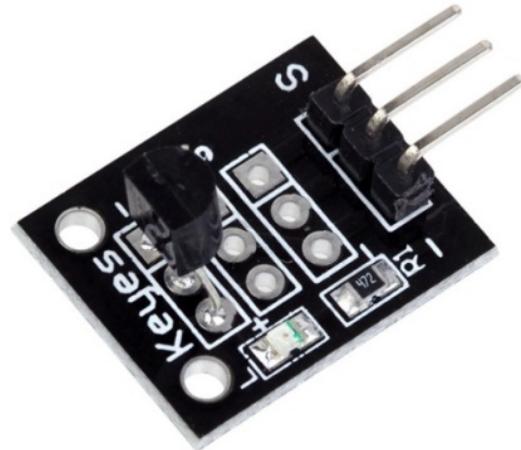


All models presently utilize streak memory for the program. Also, a more up to date manual permits it to get back to the actual program. Program memory and isolated. One is 8 pieces of 16 pieces and many. It is wide 32-digit program guidelines and pieces tally a great deal of back Pfeiffer and perhaps 1214 16 or 24 pieces. The guidance set additionally differs by a model with a more grounded chip that adds directions for the Digital PIN 0 144 sign preparing capacity that has discrete Input-yield Benn ADC and DSC which means transforming it into different changes and your great correspondence. I see jars and even you become honorable varieties and fast there for some sorts. Producers of PC programming for the advancement known as rout. This isn't the most celebrated and most renowned called Crecy and accessible free of charge online constructing agents and compiler C++. Outsiders and a few open-source gadgets are additionally accessible. This is how Mac Big. I see Loke. It doesn't rise like here. If we talk about Arduino Scott like Arduino board it's simply I see as you can see it's simply decent and you need to associate it somewhere around understanding what garbage bin be associated or with a stockpile of wire. We will talk about this in the forthcoming exercises. Presently we should discuss the past r raspberry as you see this is the soul of the card-sized PC initially intended for instruction enlivened by the nineteenth 81 BBC Micro Creator Ibn Uptown.

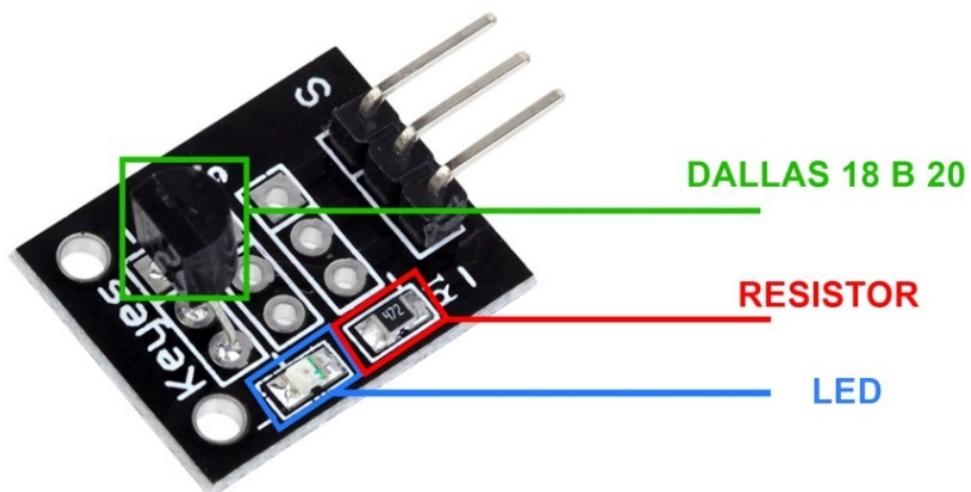


The point is to make minimal effort gadgets that will improve programming abilities and help with college level arrangement. In any case, on account of its little size and the value that can be gotten to immediately embraced by tinkles making vehicles and electronic fans for projects that need more than essential math, for example, Arduino or the distant regulator that I accept is more slow than present day PCs or work areas yet completely Linux and can Providing all normal capacities that suggest at low degrees of utilization. So this is a quick presentation. It is a quick prologue to Big Mike. Regardless of whether we are not raspberry pi. On the off chance that you don't have the foggiest idea about this. Regulator or minicomputer. Then, we will speak more about the contrast between the three. However, we need to discuss what every one of them so on the off chance that you just come out somewhat about or to my control, you will understand what resembles what the particulars are. So it's for this exercise. Much obliged to you for watching. This is instructive training designing group.

## KY-001 TEMPERATURE SENSOR MODULE

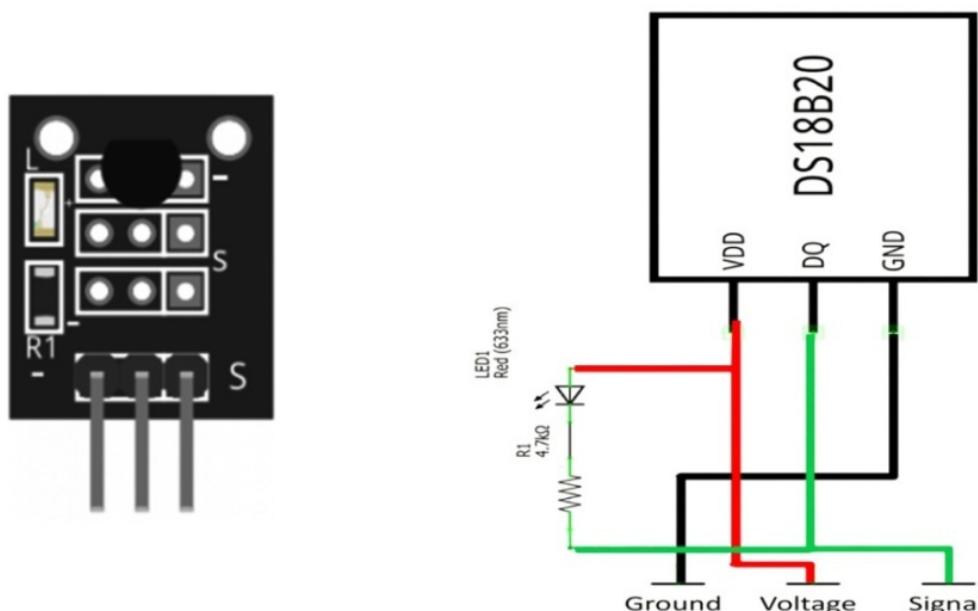


We will discuss the temperature sensor known as KY 001. This is a module itself So we should begin. This module has incorporated information from a temperature sensor called Dallas 18 B 20. With single bass innovation, a resistor and a drove the resistor utilized in this module is 4.7-kilo ohm and the fundamental purpose behind utilizing their resistor is to low mid current circling inside the module.



In different words to keep the current from consuming our module that LED illuminates to show is the module is working appropriately or

not. Presently, I will show how these parts are associated together. There we have the actual sensor and obviously, the LED light and there it is you Sir r1 on the third tube To perceive how the pins of the modules are associated with this board. Thus, we have web the progression of the voltage in dark with the progression of ground. What's more, in green the progression of the indication of this module it is yours to quantify the temperature with a sensor like Dallas d s 18 b 20, which has a temperature range from less than 55 entryways Celsius to in addition to 125 check Celsius or less 67 evaluations Fahrenheit to 257.



Extraordinary reasonable evening and a blunder size of 0.5 doors look through which isn't a lot of an issue expecting that you won't go into us for exact estimation of the temperature. Dallas d s 18 b 20 sensors can interface with a lot of other sensor modules. The article To be 20 in any event, utilizing a similar advanced port, as a result of the single transport innovation and the extraordinary chronic number of each dsat Mb to any sensor. The base in these modules is three. So we have the ground torment with the short sign, the voltage pin in the centre and have been of the sign up with s sign. So we should discuss the information exchange. This module gives a computerized information exchange. So we can associate the agonies of the underwriter with any computerized some portion of the diverse microcontrollers, like Arduino or Raspberry Pi.

## APPLICATIONS:

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

### Specifications

Operating Voltage	3.0V to 5.5V
Temperature Measurement Range	-55°C to 125°C [-57°F to 257°F]
Measurement Accuracy Range	±0.5°C
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

### Connection Diagram

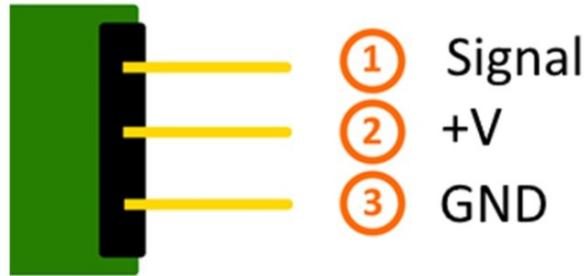
KY-001	Arduino
S	Pin 2
middle	+5V
-	GND

## CONNECTIONS ARDUINO:

Sensor Signal = [Pin 4]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]



## INPUT:

### EXAMPLE SENSOR CODE:

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 2

// Setup a oneWire instance to communicate with any OneWire
// devices (not just Maxim/Dallas temperature ICs)
OneWireoneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperaturesensors(&oneWire);

void setup(void)
{
// start serial port
Serial.begin(9600);
Serial.println("Dallas Temperature IC Control Library Demo");
// Start up the library
```

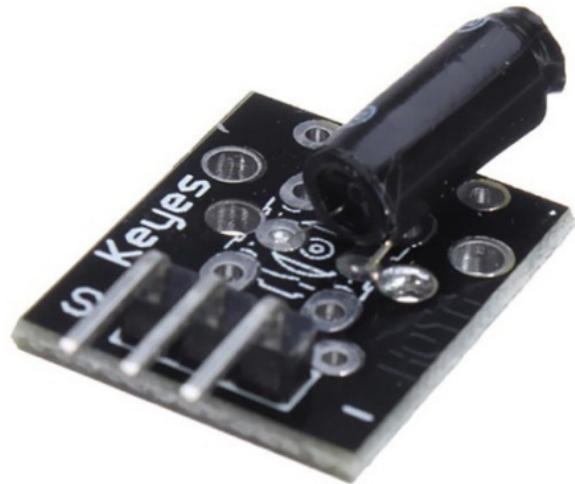
```
sensors.begin(); // IC Default 9 bit. If you have troubles consider
upping it 12. Ups the delay giving the IC more time to process the
temperature measurement

}

void loop(void)
{
// call sensors.requestTemperatures() to issue a global temperature
// request to all devices on the bus
Serial.print("Requesting temperatures...");
sensors.requestTemperatures(); // Send the command to get
temperatures
Serial.println("DONE");

Serial.print("Temperature for Device 1 is: ");
Serial.print(sensors.getTempCByIndex(0)); // Why "byIndex"? You
can have more than one IC on the same bus. 0 refers to the first IC
on the wire
```

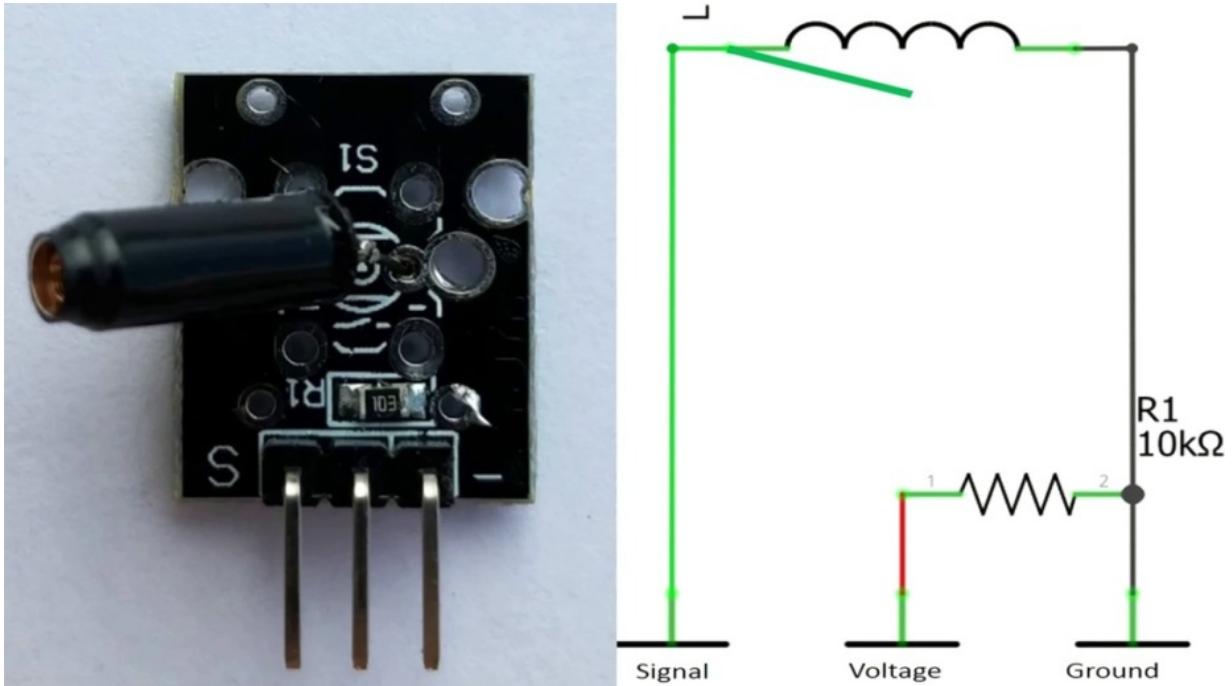
## KY-002 VIBRATION SWITCH MODULE



The vibration switch sensor is known as KY 002. This is simply the module. So how about we begin. This module has a coordinated inboard, a vibration sensor called SW 180 10 P, and, a resistor, the resistor in this modules Thank you for an arm and the primary explanation behind utilizing the resistor is to learn to meet current circling inside the module. As such, to keep flows from consuming our module. Presently I will show how these parts are associated together. Thus, there we have the actual sensor and obviously, the resistor r1 on the correct you can perceive how these pins in this module are associated in this book. Thus, we have in that the progression of the voltage we have in dark, the progression of the ground.

Furthermore, we have in green the progression of the sign the location of this particular law of stun and vibration and truly respond to stun and vibration by shutting the mystery the vibrations switch that they are in this module it's regularly there is in state and will be on while accomplishing an appropriate vibration by an outer power contacting it. The aftereffect of development inside the vibration module coming about changes in electric property. After giving the data that the sensor has detected the stun the module promptly will

Reset its typical state to hang tight for different stuns. As such, the Sasa will be constantly sitting tight for stuns and vibration, to awaken the sensor the beans in this module Aarthi. Along these lines, we



have the ground torment with the less significant, the voltage pin in the centre, and obviously, the agony of the sign with an S sign. So we should discuss the indication of this module gives an advanced sign. So we can interface the pins of the endorser with any advanced port or the distinctive microcontrollers, like Arduino or Raspberry Pi.

## Specifications

Operating Voltage	5v
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

## Connection Diagram

Connect the Power line (middle) and ground (-) to +5 and GND respectively. Connect signal (S) to pin 3 on the Arduino.

## Connections Arduino:

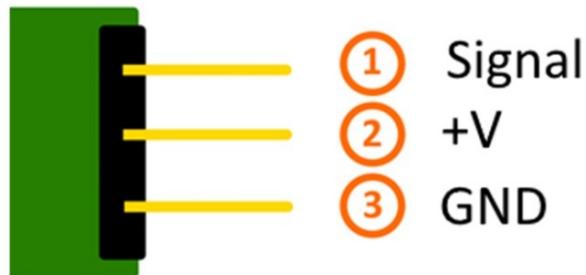
LED + = [Pin 13]

LED - = [Pin GND]

Sensor Signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]



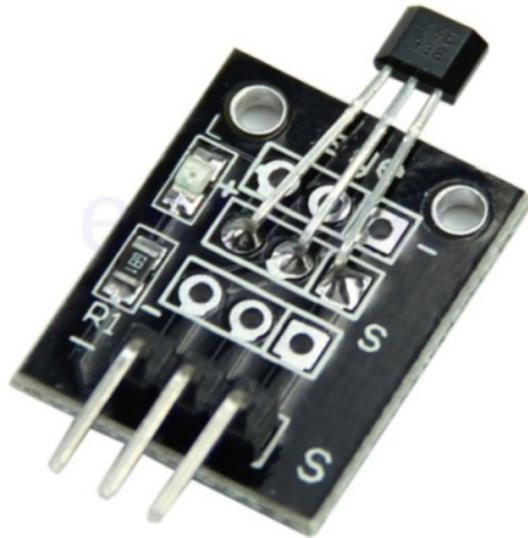
## INPUT:

## EXAMPLE SENSOR CODE:

```
int Led = 13; // define the LED Pin
```

```
int shock = 3 // define the sensor Pin  
intval; // define a numeric variable val  
  
void setup () {  
    pinMode (Led, OUTPUT); // LED pin as output  
    pinMode (shock, INPUT); // input from KY-002 sensor  
}  
  
void loop () {  
    val = digitalRead (shock); // read the value from KY-002  
    if (val == HIGH ) { // when sensor detects shock, LED  
flashes  
        digitalWrite(Led, LOW);  
    } else {  
        digitalWrite (Led, HIGH);
```

## KY-003 HALL MAGNETIC SENSOR MODULE



The entire attractive sensor is known as KY 003. This is a module itself. So how about we begin. This module has incorporated on board an attractive sensor called a 3141. One is sr, and one LED. The resistor utilized in this module is six sounded at arm and the primary purpose behind utilizing the resistor is to restrict current circling inside the module. As such to keep the current from consuming our module. The LED illuminates to show if the module is working appropriately or not. Presently I will show how these parts are associated together. There we have the actual sensor and drove as one other resistor r1 on the correct you can perceive how the pins of the module are associated with all these boards. Along these lines, we have web the progression of the voltage, we have in the dark of a progression of the ground. Also, we have in green the progression of the sign, this module is utilized to identify the attractive field. On the off chance that there is no attractive field present, the yield signs of the module will give it will be high. On the off chance that there is an attractive field that recognized the yield signal that the module will give it will be low and simultaneously the LED in the module will turn on. Thus, the LED will turn on just when an attractive field has been recognized by the attractive sensor. This module has three pins The piece in this module are the ground pin with a less significant, the voltage pin in the centre and the agony of

the sign with an S sign. So we should discuss the sign. This module gives an advanced sign. So we can interface the pin of the sign with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

#### Specification

Operating Voltage	4.5V to 24V
Operating Temperature Range	-40°C to 85°C [-x°F to x°F]
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

#### Connection Diagram

KY-003	Arduino
S	Pin 3
middle	+5V
-	GND

## CONNECTIONS ARDUINO:

LED + = [Pin 13]

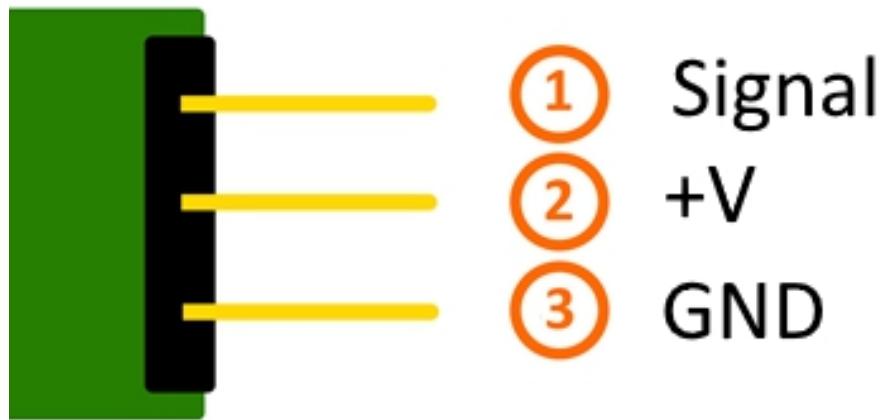
LED - = [Pin GND]

Sensor Signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]

## INPUT:

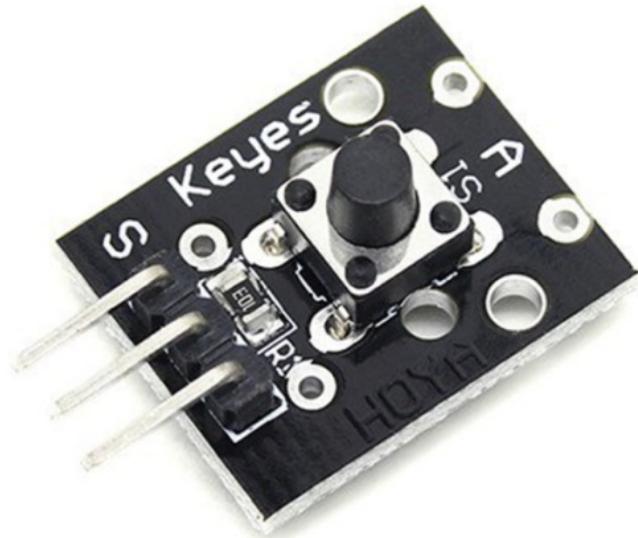


## EXAMPLE SENSOR CODE:

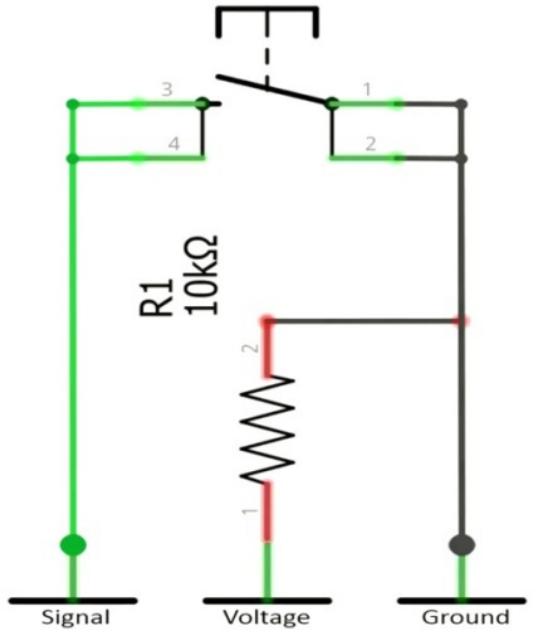
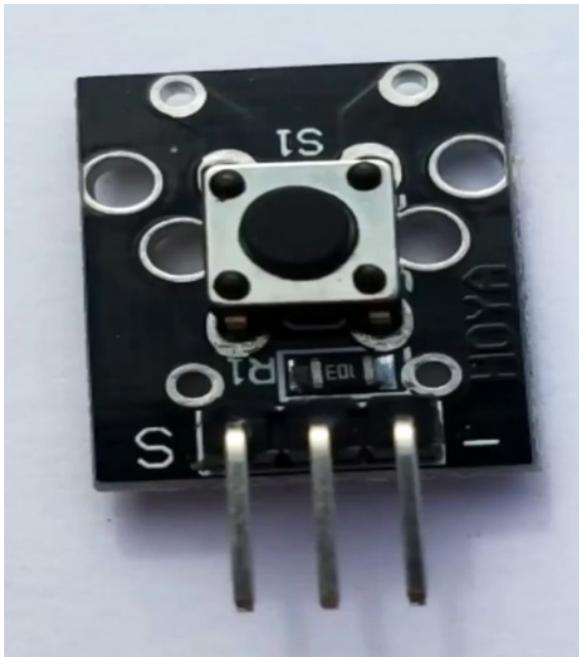
```
int led = 13; //Define the LED pin  
intbuttonpin = 3; //Define the push button pin  
intval; //Define a numeric variable  
  
void setup()  
{  
    pinMode(led,OUTPUT);  
    pinMode(buttonpin,INPUT);  
}  
  
void loop()  
{  
    val = digitalRead(buttonpin); // check the state of the button  
    if(val==HIGH) // if button is pressed, turn LED on  
    {
```

```
digitalWrite(led,HIGH);
}
else
{
digitalWrite(led,LOW);
```

## **KY-004 KEY SWITCH MODULE**



The base module is known as KY 004. This is a module itself. So how about we get started. This module has incorporated on-board a catch called f set 1713 and one resistor. The resistor utilized in this module is thank you on and the primary explanation behind utilizing the resistor is to restrict current circling inside the module. As such to keep the current from consuming our module now I will show how these segments are associated together. There we have the actual sensor and obviously, there is Easter r1alright you can perceive how the piece of the module are associated on this board, we have web the progression of the voltage, we have in dark, the progression of the ground.



What's more, obviously we have in green, the progression of the sign. The catch Episode 70 and 13 has two states squeezed or undressed which are his ordinary state and have the existing pattern of a hundred thousand ticks. At the point when this catch is in his typical express, the voltage will remain in the initial two legs or pins of the catch and the yield signal that the module will give in an ordinary state will be high. At the point when we press the catch, the voltage will pass the opposite side, and as result, changing the yield sign to low. At the point when you discharge the catch, it will go out consequently in his typical state due to the spring inside, who power the shirt. The power that is expected to press a catch is a hundred 80 to 230 gam. This module has three pins. The pins in this module are the ground pin with a less significant, the voltage pin in the centre and the agony of the sign with an S sign. So we should discuss the sign. This module gives a computerized signal so we can associate the agony of the sign with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi

#### Specifications

Rating	50mA 12VC
Environment temperature	-25°C to 105°C [-13°F to 221°F]
Electrically Life	100,000 cycles
Operating Force	180/230(±20gf)
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

KY-004 Connection Diagram

KY-004	Arduino
S	Pin 3
middle	+5V
-	GND

## CONNECTIONS ARDUINO:

LED + = [Pin 13]

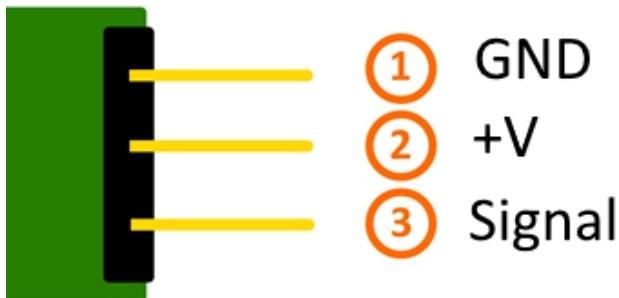
LED - = [Pin GND]

Sensor Signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]

## INPUT:



## **EXAMPLE SENSOR CODE:**

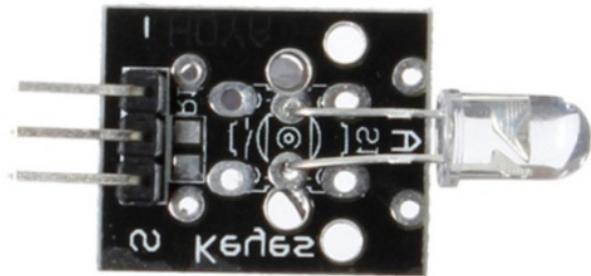
```
int led = 13; //Define the LED pin
intbuttonpin = 3; //Define the push button pin
intval; //Define a numeric variable

void setup()
{
    pinMode(led,OUTPUT);
    pinMode(buttonpin,INPUT);

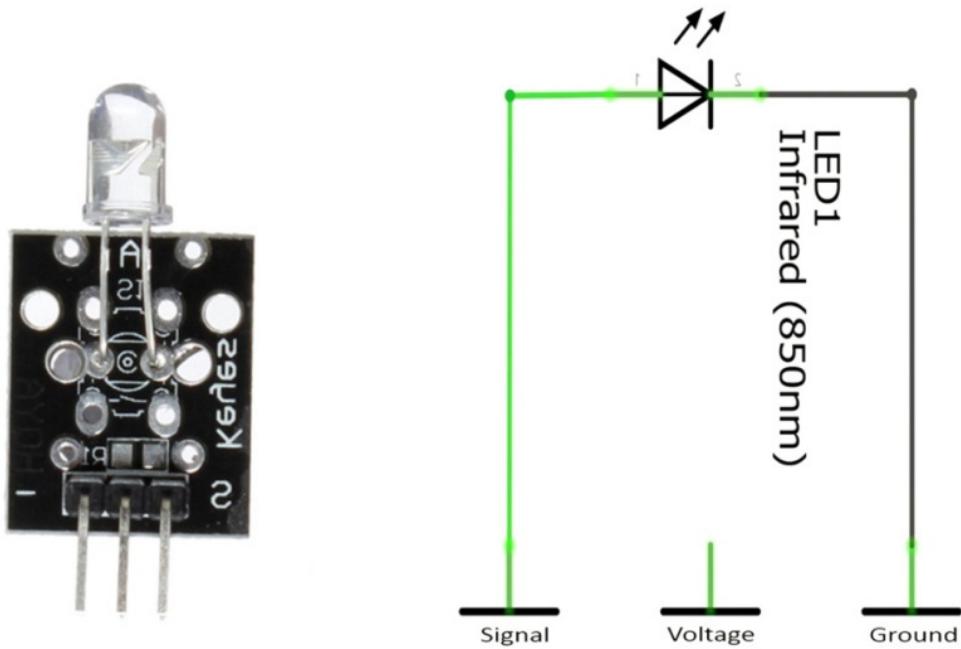
}

void loop()
{
    val = digitalRead(buttonpin); // check the state of the button
    if(val==HIGH) // if button is pressed, turn LED on
    {
        digitalWrite(led,HIGH);
    }
    else
    {
        digitalWrite(led,LOW)
```

## **KY-005 INFRARED EMISSION SENSOR MODULE**



The infrared transmitter module is known as KY 005. This is a module itself. So we should begin. So folks, as should be obvious, this module has just the infrared transmitter and that's it. Presently I will show how the infrared transmitter is associated with this board. There we have the infrared transmitter itself. What's more, presently I will show the progression of the ground in dark. Also, I will show in green, the progression of the sign. The infrared transmitter is comparative by the look with a LED, however, is this made by alternate semiconductor materials. The infrared transmitter is otherwise called infrared producing diode, the measurement of the transmitter is five millimetres. The infrared transmitter will change over the power into infrared lights, the sign that This module will communicate will have a recurrence of 38 kilohertz. This module is utilized along with KY 0 22 known as the infrared beneficiary module.



So we should discuss the pins. The pins in this module are three, we have the ground pin with the less significant, the voltage pin is in the centre. Furthermore, we have the pin of the sign with an S sign. So we should discuss the sign. This module takes a computerized signal, so we can interface the pin of the sign with any advanced piece of various microcontroller sheets like Arduino or Raspberry Pi.

#### Specifications

Operating Voltage	5V
Forward Current	30 ~ 60 mA
Power Consumption	90mW
Operating Temperature	-25°C to 80°C [-13°F to 176°F]
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

#### KY-005 Connection Diagram

KY-005	Arduino UNO
S	Pin 3
middle	+5V
-	GND

## CONNECTIONS ARDUINO:

KY-005

Signal = [Pin 3 (Arduino Uno) | Pin 9 (Arduino Mega)]

GND+resistor = [Pin GND\*]

GND = [Pin GND]

## INPUT:



## EXAMPLE SENSOR CODE:

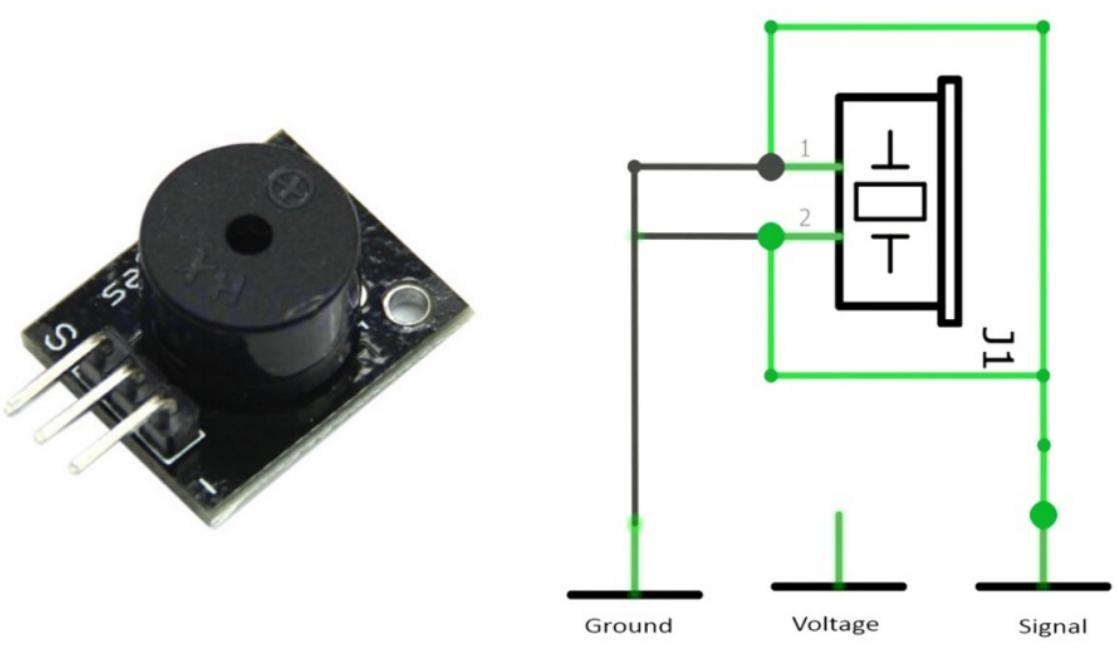
```
#include <IRremote.h>  
IRsend irsend;  
void setup()  
{  
    Serial.begin(9600);  
}  
  
void loop()  
{  
    for (int i = 0; i < 50; i++) {
```

```
irsend.sendSony(0xa90, 12); // Sony TV power code  
delay(40);
```

## KY-006 SMALL PASSIVE BUZZER MODULE



This is a module itself. So how about we begin. As should be obvious, this module has just the ringer inside and that's it. Presently I will show how the signal is associated on this board. There we have the actual signal. On the correct, you can perceive how the piece of the module symbol noted on this board. So we have in dark, the progression of the ground. What's more, obviously we have in green, the progression of the sign. This module can't deliver sound without anyone else. So we need to compose a program With postponements and we need to realize that the bell will give a sound just before the defer time and not during the deferral, the ringer will give a sound sign of 1.5 to 2.5 kilohertz. So we should discuss the pins. The pins in this module are three, we have the ground pin with the less significant, the voltage pin is in the centre.



Also, we have the torment of the sign with an S sign. So we should discuss the sign. This module takes an advanced sign so we can associate the agony of the sign with any computerized part of various microcontroller sheets like Arduino or Raspberry Pi.

#### Specifications

Operating Voltage	1.5 ~ 15V DC
Tone Generation Range	1.5 ~ 2.5kHz
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

#### KY-006 Connection Diagram

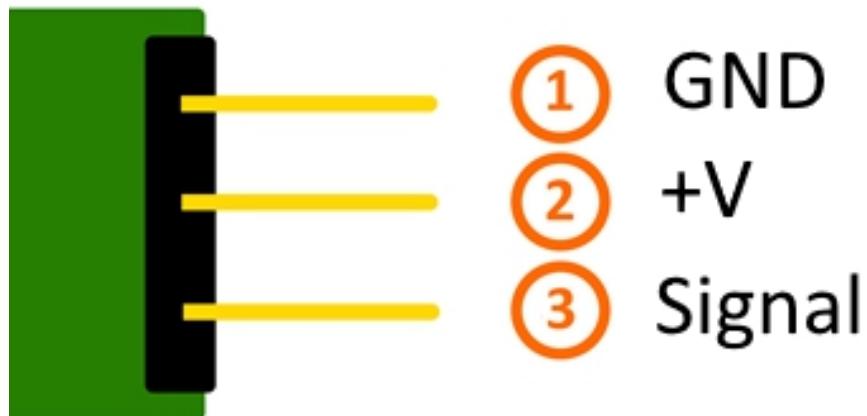
Connect signal (S) to pin 8 on the Arduino and ground (-) to GND. The middle pin is not used.

## CONNECTIONS ARDUINO:

Sensor signal = [Pin 8]

Sensor - = [Pin GND]

## INPUT:



## EXAMPLE SENSOR CODE:

```
int buzzer = 8; // set the buzzer control digital IO pin

void setup() {
    pinMode(buzzer, OUTPUT); // set pin 8 as output
}

void loop() {
    for (int i = 0; i < 80; i++) { // make a sound
        digitalWrite(buzzer, HIGH); // send high signal to buzzer
        delay(1); // delay 1ms
        digitalWrite(buzzer, LOW); // send low signal to buzzer
        delay(1);
```

```
}

delay(50);

for (int j = 0; j < 100; j++) { //make another sound
digitalWrite(buzzer, HIGH);

delay(2); // delay 2ms

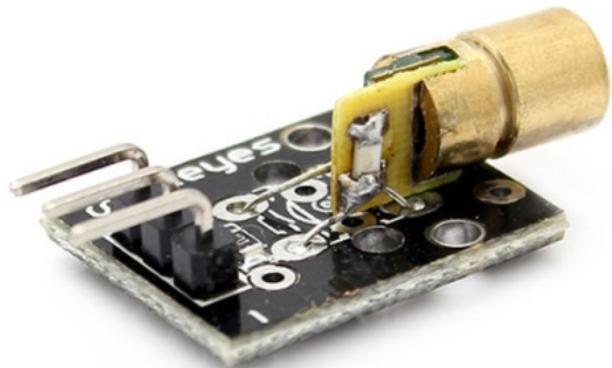
digitalWrite(buzzer, LOW);

delay(2);

}

delay(100);
```

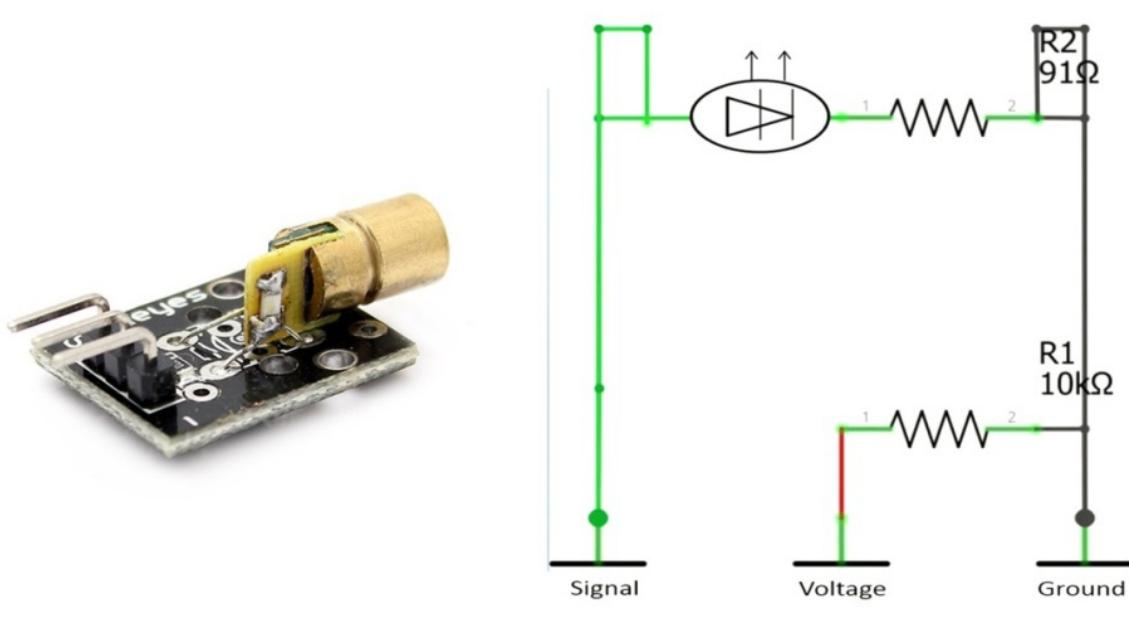
## KY-008 LASER SENSOR MODULE



The laser transmitter module is known as KY 008. This is a module itself. So how about we begin. This module has incorporated on board laser transmitter and two resistors resistor r1 and the resistor R to the resistor r1 utilized in this module. It's Thank you long and there is this there are two perspectives in this module. It's 91 on and the fundamental purpose behind utilizing the resistor is to restrict money-gathering inside the module. As such to keep the current from consuming our module. Presently, I will show how these segments are associated together. There we have the actual laser and obviously, the resistor r1 and the resistor are to the right again perceive how the pins of the module are associated on this board. Thus, we have web the progression of the voltage in dark, the progression of the ground and obviously, we have in green the progression of the signal this module comprises of 650-nanometer red laser diode head.

This module has two fundamental employments. It very well may be utilized to include sightseer wire distinguishing and can be utilized as a pointer laser are the extraordinary method to draw distances and cutoff points as they shot straightforwardly, frequently at significant distances, combined with a photoresistor, or specific laser recognizing module can be utilized to identify the way of the laser. In

principle, lasers can go until they hit something or in other words, laser light will disperse in the climate and keeping in mind that experiencing the focal point, which causes the dissipate and increment dab size. Be cautious and don't immediate the light of the laser in your eyes since it's risky. So we should discuss the pins.



The pins in this module are three. We have the ground pin with the short sign, the voltage pin is in the centre. Also, obviously, we have a circling of the sign with an S sign. So we should discuss the sign. This module takes an advanced sign. So we can interface the pin of the sign with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

#### Specifications

Operating Voltage	5V
Output Power	5mW
Wavelength	650nm
Operating Current	less than 40mA
Working Temperature	-10°C ~ 40°C [14°F to 104°F]
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

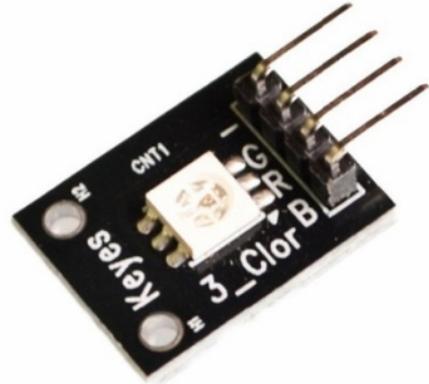
#### KY-008 Connection Diagram

Connect signal (S) to pin 13 on the Arduino and ground (-) to GND. Middle pin is not used.

## EXAMPLE SENSOR CODE:

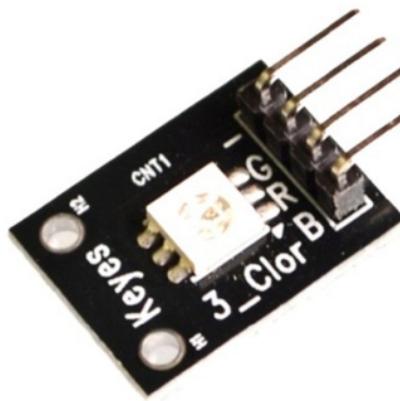
```
intlaserPin = 13;  
void setup() {  
    pinMode(laserPin, OUTPUT); // Define the digital output  
    // interface pin 13  
}  
void loop() {  
    digitalWrite(laserPin, HIGH); // Open the laser head  
    delay(1000); // Delay one second  
    digitalWrite(laserPin, LOW); // Close the laser head  
    delay(1000);
```

## KY-009 3-COLOUR FULL-COLOUR LED SMD MODULES



The RGB LED SMD module known as KY 009. This is a module itself. So how about we begin. So as should be obvious, this module has just the LED SMD module and that's it. Presently I will show how the LED SMD is associated with this board. There we have the LED SMD itself. On the correct, you can perceive how the pins of the module are associated on this board. We have in dark, the progression of the ground and we have in green the progression of the three pins for the LEDs. As should be obvious, in this module, it is utilized to deliver light.

This module can deliver light in three tones because inside the chip, our three LEDs are blue, red, and green. With this module, we can illuminate each LED Individually, or we can make a blended shading utilizing the mix of red, blue, and green. We should discuss the paints. The paints in this module are four, we have a VA ground pin with a short sign.



B      G      R      GND

Furthermore, we have three sign pins named R, G, and B. So we should discuss the sign. This module takes three computerized signals, so we can associate the pin servo sign with any advanced ports of various microcontroller sheets, like Arduino or Raspberry Pi.

#### KY-009 Specifications

Operating Voltage	5V max Red 1.8V ~2.4V Green 2.8V ~ 3.6V Blue 2.8V ~ 3.6V
Forward Current	20mA ~ 30mA
Operating Temperature	-25°C to 85°C [-13°F ~ 185°F]
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

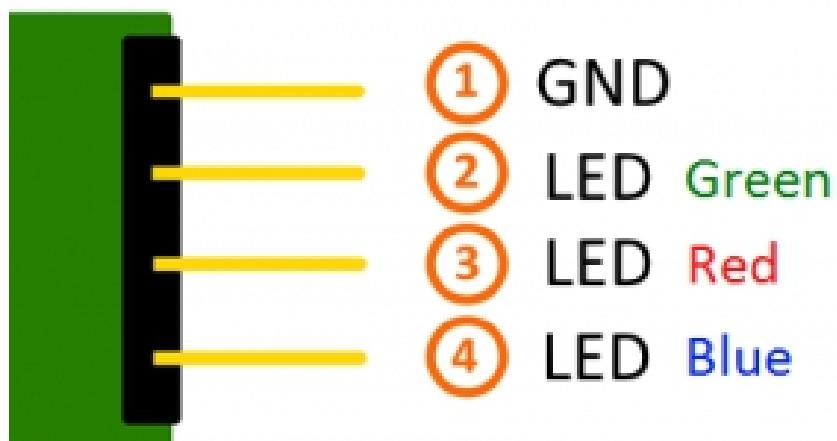
#### KY-009 Connection Diagram

KY-009	Breadboard	Arduino
R	180Ω resistor	Pin 9
G	110Ω resistor	Pin 10
B	110Ω resistor	Pin 11
-	GND	GND

## CONNECTIONS ARDUINO:

LED Red = [Pin 10]  
LED Green = [Pin 11]  
LED Blue = [Pin 12]  
Sensor GND = [Pin GND]

## INPUT:

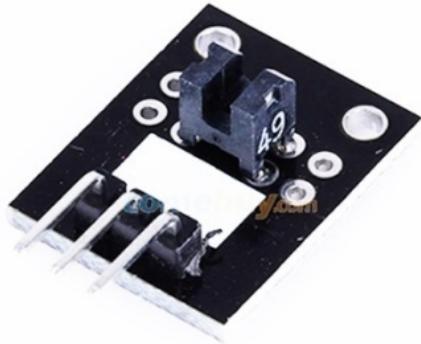


## EXAMPLE SENSOR CODE:

```
intredpin = 11; //select the pin for the red LED  
intbluepin = 10; // select the pin for the blue LED  
intgreenpin = 9;// select the pin for the green LED  
  
intval;  
  
void setup() {  
pinMode(redpin, OUTPUT);  
pinMode(bluepin, OUTPUT);
```

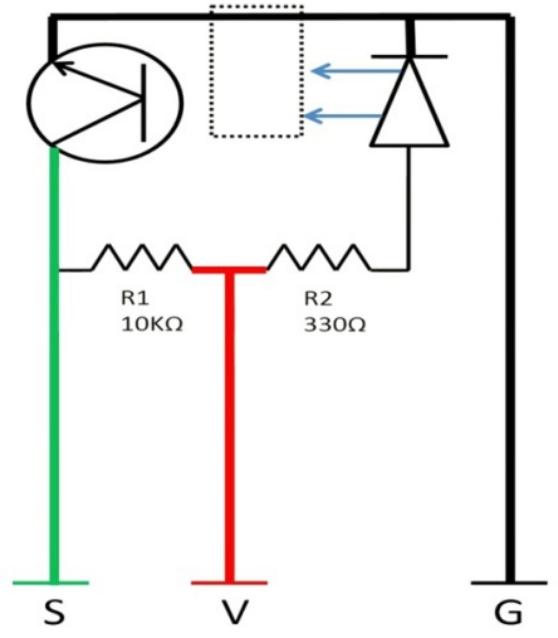
```
pinMode(greenpin, OUTPUT);
Serial.begin(9600);
}
void loop()
{
for(val = 255; val> 0; val--)
{
analogueWrite(redpin, val); //set PWM value for red
analogueWrite(bluepin, 255 - val); //set PWM value for blue
analogueWrite(greenpin, 128 - val); //set PWM value for green
Serial.println(val); //print current value
delay(1);
}
for(val = 0; val< 255; val++)
{
analogueWrite(redpin, val);
analogueWrite(bluepin, 255 - val);
analogueWrite(greenpin, 128 - val);
Serial.println(val);
delay(1);
}
```

## KY- 010 OPTICAL BROKEN MODULE



The photograph interrupter module is known as KY 010. This is a module itself. So how about we begin. This module has coordinated on board a photograph interrupter, or optical broken and two resistors, resistor r1 and resistor or to the resistor r1. utilized in this module. It's Thank you for an arm and the resistor are two utilized in this module. It's 330 ohms. Also, the fundamental purpose behind utilizing the resistor is to restrict money ordering inside the module. At the end of the day to save Correct for consuming our module. Presently I will show how these segments are associated together. There we have the actual sensor and obviously, the resistor r1 and the resistor R two. On the correct, you can perceive how the pins of the module are associated on this board. In this way, we have web the progression of the voltage in dark the progression of the ground. Furthermore, we have in green the progression of the sign. This module has onboard the photograph intrude on that is a photo sensor that coordinates light emanating components and light getting components into a solitary bundle. This gadget will turn the mystery optically on or off. It is comprised of a plastic part that has a state of you where the infrared diode is located. to one side and before it is a photograph semiconductor by a gathering a light emission light starting with one end then onto the next. This identifier can distinguish when an article passes between the sides, breaking the

correspondence between them. An article that will go through the space between the photograph semiconductor and the infrared eating regimen will intrude on the infrared light and the sign that This module will give will change from low to high. This gadget is extremely quick and ideal for checking and affectability. The beans in



this module is three. So we have the ground pin with a less sign. The voltage pin in the centre and obviously, the pin of the sign With an S sign. So we should discuss the sign. This module gives an advanced sign. So we can interface the pins of the sign with any computerized ports of various microcontroller sheets like Arduino or Raspberry Pi.

## Specifications

Operating Voltage	3.3 ~ 5V
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

## KY-010 Connection Diagram

KY-010	Arduino
- (left)	GND
middle	+5V
S (right)	Pin 3

## CONNECTIONS ARDUINO:

LED + = [Pin 13]

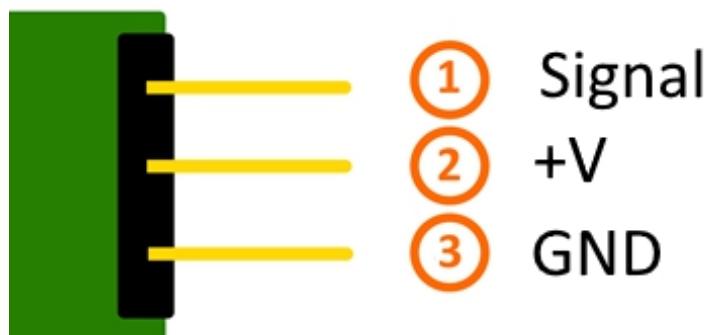
LED - = [Pin GND]

Sensor Signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]

## INPUT:



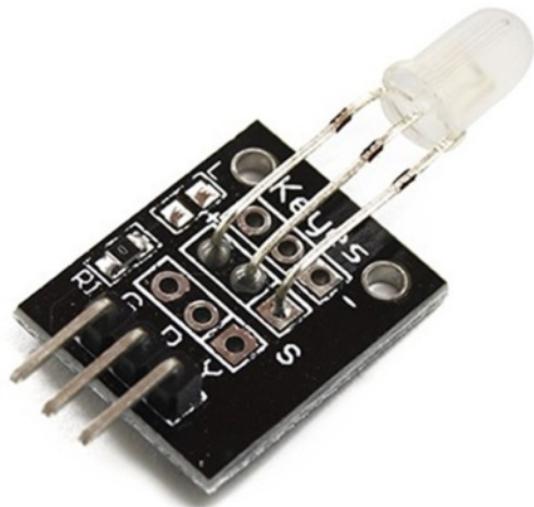
## **EXAMPLE SENSOR CODE:**

```
int Led = 13; // define LED pin
intbuttonpin = 3; // define photo interrupter signal pin
intval; //define a numeric variable

void setup()
{
    pinMode(Led, OUTPUT); // LED pin as output
    pinMode(buttonpin, INPUT); //photo interrupter pin as input
}

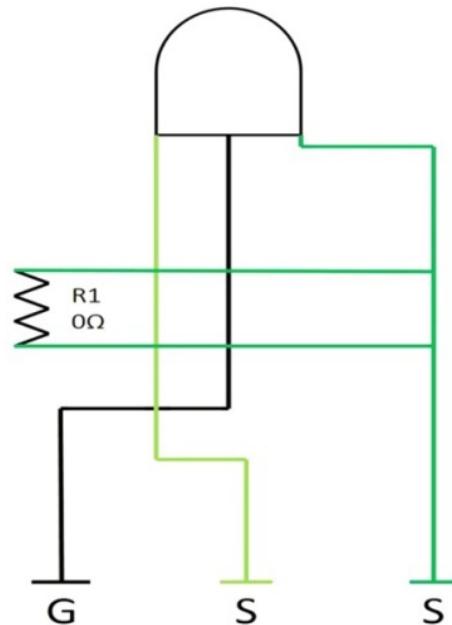
void loop()
{
    val=digitalRead(buttonpin); //read the value of the sensor
    if(val == HIGH) // turn on LED when sensor is blocked
    {
        digitalWrite(Led,HIGH);
    }
    else
    {
        digitalWrite(Led,LOW);
    }
}
```

## **KY-011 2-COLOUR LED MODULE**



The two shading LCD is known as KY 0 11. This is simply the module so how about we begin. This module is coordinated ready, one LED and one resistor, the resistor utilized in this module. It's 0 ohms. Furthermore, I don't know the motivation behind why we utilize a 0-ohm resistor in this model. Presently I will show how these parts are associated together.

There we have the actual LED, and obviously, the resistor r1. On the correct, you can perceive how the piece of the module is associated on this board. We have in dark, the progression of the ground, we have in green and in light green, the progression of the sign additionally of the LEDIn this module can deliver light in two tones in red, and in green. With this module, we can illuminate each tone exclusively. Or then again we can make a blended shading utilizing the mix of the red tone and the green tone. So we should discuss the pins.



The pins in this module are three, we have the ground pin with a less sign. What's more, obviously we have two different pins which are the pins of the signal. So we should discuss the sign. This module takes to advanced sign. So we can interface the pins of the sign with any computerized port of various microcontroller sheets like Arduino or Raspberry Pi.

## **CONNECTIONS ARDUINO:**

LED Green = [Pin 10]

LED Red = [Pin 11]

Sensor GND = [Pin GND]

## **INPUT:**



## EXAMPLE SENSOR CODE:

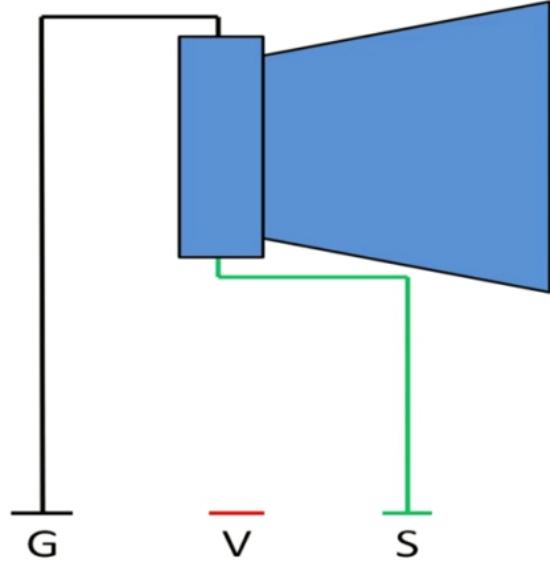
```
intredpin = 11; // pin for red signal  
intgreenpin = 10; // pin for green signal  
intval;  
  
void setup() {  
    pinMode(redpin, OUTPUT);  
    pinMode(greenpin, OUTPUT);  
}  
  
void loop() {  
    for(val = 255; val> 0; val--) {  
        analogueWrite(redpin, val); //dim red  
        analogueWrite(greenpin, 255 - val); // brighten green  
        delay(15);  
    }  
    for(val = 0; val< 255; val++) {  
        analogueWrite(redpin, val); //brighten red
```

```
analogWrite(greenpin, 255 - val); //dim green  
delay(15);
```

## **KY-012 ACTIVE BUZZER MODULE**



KY 012 This is a module itself. So we should begin. As should be obvious, this module has just a signal inside and that's it. Presently I will show how the bell is associated on this board. There we have the actual ringer. On the correct, you can perceive how the pins of the measured composed on this board. So we have in the dark the progression of the ground. Also, we have in green, the progression of the sign. This sort of bells can deliver sound without anyone else. All they require is a force from any source. Furthermore, on the off chance that we like to change the sound, we can do that by utilizing any sort of microcontroller.



This module gives us a sound recurrence of 2.5 kilohertz with a mistake size of in addition to short 300 hertz So we should discuss the pins. The pins in this module are three, we have the ground pin with the less significant, the voltage pin is in the centre. Furthermore, we have a pin of the sign with an S sign. So we should discuss the sign. This module takes a computerized signal, so we can interface the agony of the sign with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

## **CONNECTIONS ARDUINO:**

Sensor Signal = [Pin 13]

Sensor [N.C] =

Sensor GND = [Pin GND]

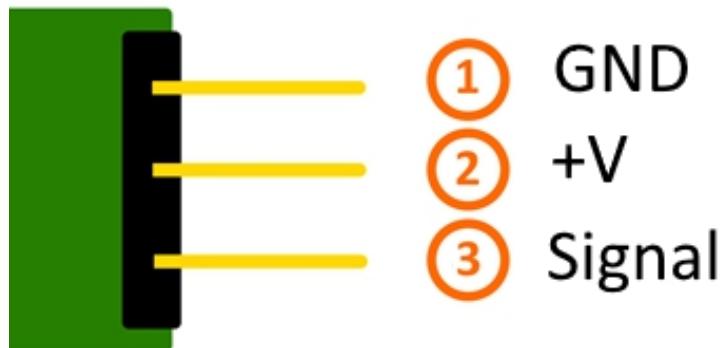
## KY-012 Specifications

Operating Voltage	3.5V ~ 5.5V
Maximum Current	30mA / 5VDC
Resonance Frequency	2500Hz ± 300Hz
Minimum Sound Output	85Db @ 10cm
Working Temperature	-20°C ~ 70°C [-4°F ~ 158°F]
Storage Temperature	-30°C ~ 105°C [-22°F ~ 221°F]
Dimensions	18.5mm x 15mm [0.728in x 0.591in]

## KY-012 Connection Diagram

KY-012	Arduino
S	Pin 8
-	GND

## INPUT:



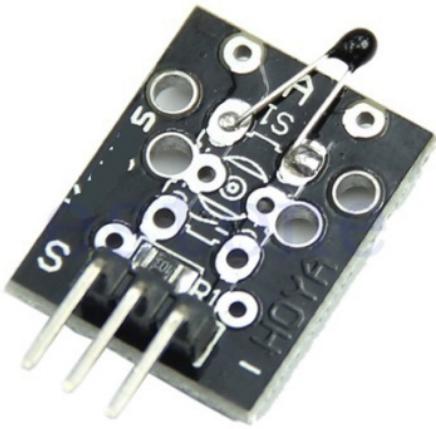
## EXAMPLE SENSOR CODE:

```
intbuzzerPin = 8;  
void setup ()  
{
```

```
pinMode (buzzerPin, OUTPUT);
}

void loop ()
{
digitalWrite (buzzerPin, HIGH);
delay (500);
digitalWrite (buzzerPin, LOW);
delay (500);
```

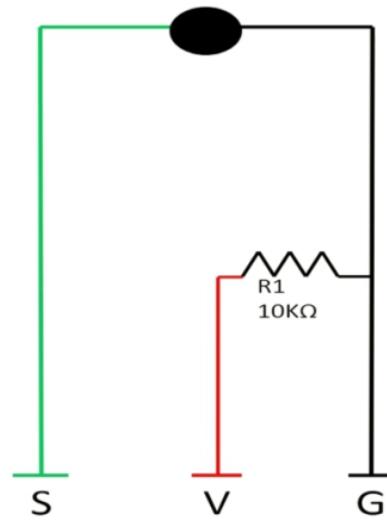
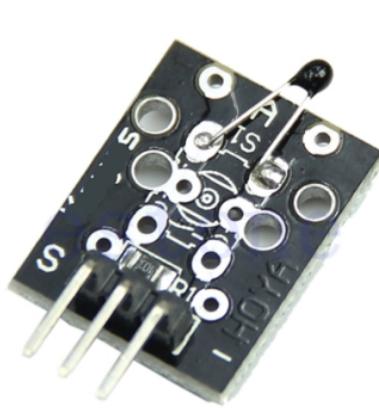
## KY-013 TEMPERATURE SENSOR MODULE



KY 013. This is simply the module. So we should begin. This module has coordinated ready, a temperature sensor and our resistor. The resistor utilized in this module. It's Thank you for an arm and the principle explanation behind utilizing the resistor is to restrict money quality inside the module. As such to keep the current from consuming our module. Presently I will show how these segments are associated together. There we have the actual sensor. What's more, there is Easter r1on the correct you can perceive how the speed of the module is associated on this board, we have web the progression of the voltage, we have in dark, the progression of the ground. What's more, we have in green, the progression of the sign thermistors is the sort of components that can feel the temperature since they are worked with semiconductor materials that are appeared to display significant obstruction changes about little temperature changes. This obstruction can be estimated utilizing a little AC or DC going through the thermistor to gauge the yield of the voltage drop. The temperature sensor is discovered to be other temperature touchy electrical protections here comes a name an unmistakable condition nation of words warm and obstruction. The module depends on crafted by a thermistor. We should increment or reduction the opposition when encompassing temperature changes. This sort of thermistors can gauge the temperature from less than 55 entryways Celsius to 125 doors Celsius or less 67. Incredible

extremely pleasant to 257 doors for a night with a blunder size of in addition to less point five entryway Celsius.

So we should discuss the pins. The pins in this module are three we have the ground pin with the less significant, the voltage pin is in the centre. What's more, obviously we have the pin of the sign with an S



sign.

So we should discuss the sign. This module gives a simple sign So we can interface the agony of the sign with any simple port of various microcontroller sheets like Arduino or Raspberry Pi.

#### KY-013 Specifications

Operating Voltage	5V
Temperature measurement range	-55°C to 125°C [-67°F to 257°F]
Measurement Accuracy	±0.5°C

#### KY-013 Connection Diagram

KY-013	Arduino
S	A0
middle	5V
-	GND

## **CONNECTIONS ARDUINO:**

Sensor Signal = [Pin 13]

Sensor [N.C] =

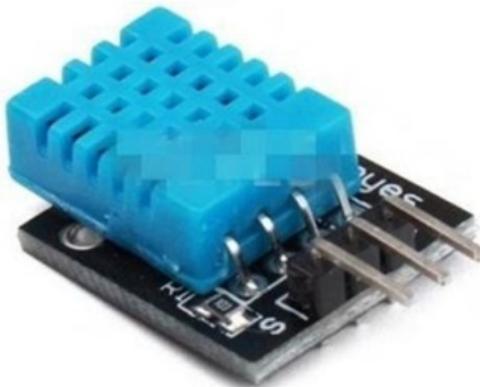
Sensor GND = [Pin GND]

## **EXAMPLE SENSOR CODE:**

```
int ThermistorPin = A0;  
int Vo;  
float R1 = 10000; // value of R1 on board  
float logR2, R2, T;  
float c1 = 0.001129148, c2 = 0.000234125, c3 = 0.0000000876741;  
//steinhart-hart coeficients for thermistor  
  
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Vo = analogueRead(ThermistorPin);  
    R2 = R1 * (1023.0 / (float)Vo - 1.0); //calculate resistance on  
    thermistor  
    logR2 = log(R2);  
    T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2)); // temperature  
    in Kelvin  
    T = T - 273.15; //convert Kelvin to Celcius  
    // T = (T * 9.0)/ 5.0 + 32.0; //convert Celcius to Farenheit  
  
    Serial.print("Temperature: ");
```

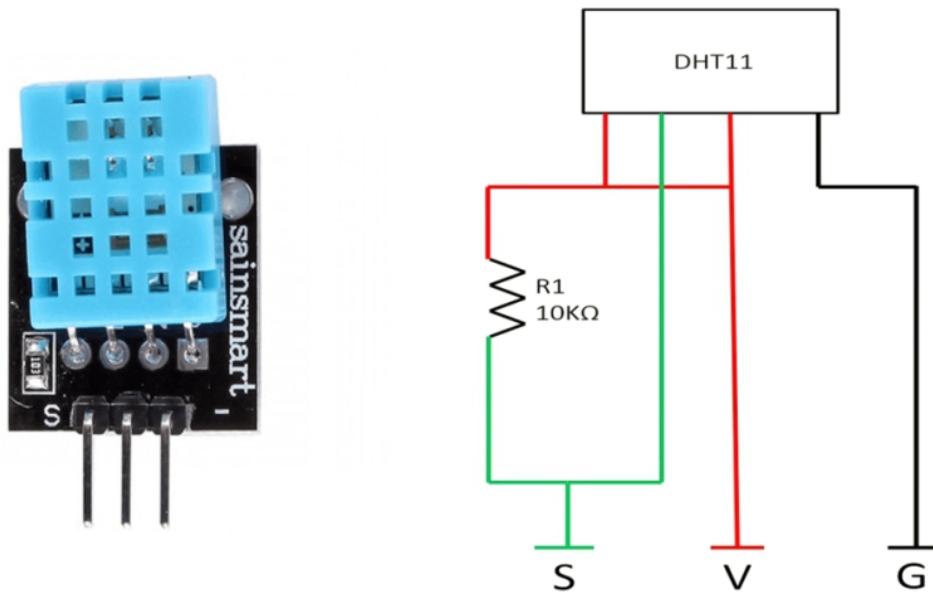
```
Serial.print(T);
Serial.println(" C");
delay(500);
}
```

## KY-015 TEMPERATURE AND HUMIDITY SENSOR MODULE



KY 015. This is simply the module. So we should begin. This module has coordinated on board a temperature and dampness sensor called the HDx 11. What's more, obviously, a resistor, the resistor utilized in this module, it's Thank you long and the fundamental purpose behind utilizing the resistor is to restrict flows circling inside the module. As such, to keep the current from consuming our model. Presently, I will show how these parts are associated together. There we have the actual sensor. Furthermore, there is an Easter Erwin. Overnight you can perceive how the speed of the module is associated with this board. We have web the progression of the voltage, we have in dark the progression of the ground, and

obviously, we have in green, the progression of the sign this module can gauge the temperatureFrom0 grades Celsius to 50 evaluation Celsius with a mistake size of in addition to short two evaluation Celsius and a mugginess from 20% to 90%. With a blunder size of in addition to short five people, it tends to be associated with any sort of microcontroller sheets, yet the wire should not surpass a most extreme length of 20 meters. This sensor is a solitary wire sequential interface that permits a quick and simple framework mix. little size, low force utilization signal transmission removes up to 20 meters settle on it great decision anyplace even in the extreme applications.



The sensor the HDL stove comprises of opposition estimation adversary and an NTC temperature estimation part DHD 11 uses an interior thermistor and a dampness sensor to decide to encompass conditions. An inner chip is answerable for changing over readings into an advanced sign. This sensor can screen temperature and dampness by giving an exceptionally adjusted computerized signal for these qualities. So we should discuss the pins. The BNC in this module are three, we have the ground pin with the short sign, the voltage pin is in the centre. Also, we have the pin of the sign with an S sign. So we should discuss the sign. This module gives a profound Don't flag so we can associate the pins of the sign with any

advanced ports of various microcontroller sheets like Arduino or Raspberry Pi.

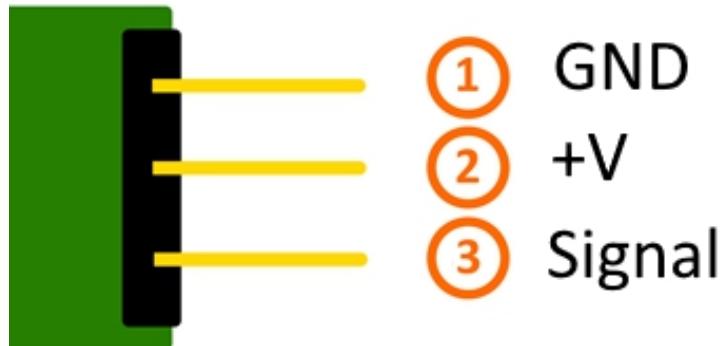
#### KY-015 Specifications

Operating Voltage	3.3V to 5.5V
Humidity measurement range	20% to 90% RH
Humidity measurement accuracy	±5% RH
Humidity measurement resolution	1% RH
Temperature measurement range	0°C to 50°C [32°F to 122°F]
Temperature measurement accuracy	±2°C
Temperature measurement resolution	1°C
Signal transmission range	20m

#### KY-015 Connection Diagram

KY-015	Arduino
S	Pin 8
middle	+5V
-	GND

## INPUT:



## CONNECTIONS ARDUINO:

GND = [Pin GND]

+V = [Pin 5V]

Signal = [Pin D2]

## EXAMPLE SENSOR CODE:

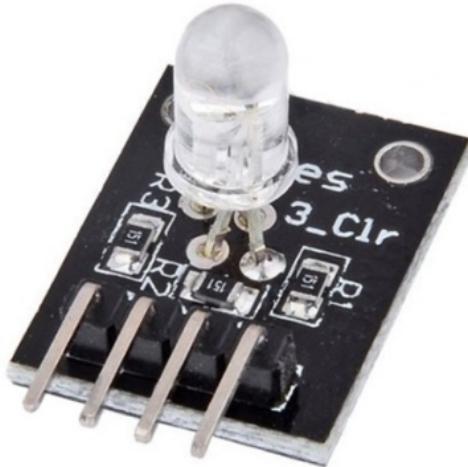
```
intDHpin = 8; // input/output pin
bytedat[5];
byteread_data()
{
bytei = 0;
byte result = 0;
for (i = 0; i< 8; i++) {
while (digitalRead(DHpin) == LOW); // wait 50us
delayMicroseconds(30); //The duration of the high level is judged to
determine whether the data is '0' or '1'
if (digitalRead(DHpin) == HIGH)
result |= (1 << (8 - i)); //High in the former, low in the post
while (digitalRead(DHpin) == HIGH); //Data '1', waiting for the next
bit of reception
}
return result;
}
voidstart_test()
{
digitalWrite(DHpin, LOW); //Pull down the bus to send the start
signal
delay(30); //The delay is greater than 18 ms so that DHT 11 can
detect the start signal
digitalWrite(DHpin, HIGH);
delayMicroseconds(40); //Wait for DHT11 to respond
```

```
pinMode(DHpin, INPUT);
while(digitalRead(DHpin) == HIGH);
delayMicroseconds(80); //The DHT11 responds by pulling the bus
low for 80us;
if(digitalRead(DHpin) == LOW)
delayMicroseconds(80); //DHT11 pulled up after the bus 80us to start
sending data;
for(inti = 0; i< 5; i++) //Receiving temperature and humidity data,
check bits are not considered;
dat[i] = read_data();
pinMode(DHpin, OUTPUT);
digitalWrite(DHpin, HIGH); //After the completion of a release of data
bus, waiting for the host to start the next signal
}
void setup()
{
Serial.begin(9600);
pinMode(DHpin, OUTPUT);
}
void loop()
{
start_test();
Serial.print("Humdity = ");
Serial.print(dat[0], DEC); //Displays the integer bits of humidity;
Serial.print('.');
Serial.print(dat[1], DEC); //Displays the decimal places of the
humidity;
```

```
Serial.println('%');
Serial.print("Temperature = ");
Serial.print(dat[2], DEC); //Displays the integer bits of temperature;
Serial.print('.');
Serial.print(dat[3], DEC); //Displays the decimal places of the
temperature;
Serial.println('C');
byte checksum = dat[0] + dat[1] + dat[2] + dat[3];
if (dat[4] != checksum)
Serial.println("-- Checksum Error!");
else
Serial.println("-- OK");

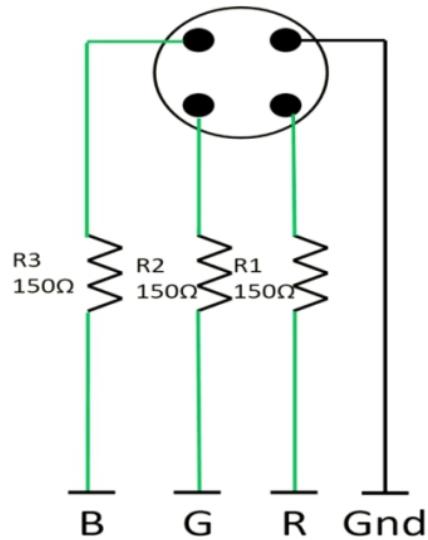
delay(1000);
```

## KY-016 3-COLOUR LED MODULE



KY 0 16. This is a module itself. So we should begin. This module has coordinated on board a LED and three resistors resistor r1, the resistor R two and obviously, these are our three, the resistor utilizing this module r 150. On each and the fundamental explanation behind utilizing the resistor is to restrict money grouping inside the module, in another word to keep current from consuming our model.

Presently I will show how these parts are associated together. There we have the actual LED and obviously resistor R one, r two and r three of the correct you can perceive how the pins of the module are associated on this board. Thus, we have in dark, the progression of the ground. Furthermore, we have in green, the progression of the red, green and blue signs. This module can deliver light blending in three tones, red, blue, and green. This happens because this LED takes three advanced signs, so one sign for each tone. With this module, we can illuminate each tone independently or we can blend them to make another tone.



With the assistance of heartbeat width regulation. The tones that we can make will be a blend of reading green and blue tone. We should discuss the pins. The paints in this module are four, we have the ground pin with the short sign. What's more we have three sign pins named R, G, and B. So we should discuss the sign. This module takes three computerized signals, so we can associate the pins of the sign with any advanced ports of various microcontroller sheets like Arduino or Raspberry Pi.

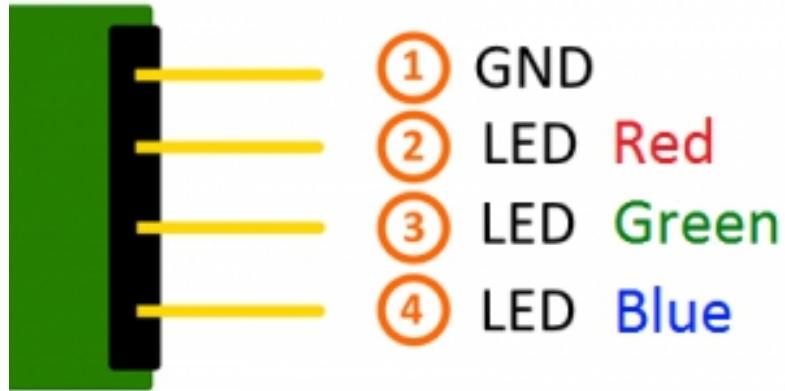
#### Specifications

Operating Voltage	5V
LED drive mode	Common cathode driver
LED diameter	5 mm

#### KY-016 Connection Diagram

KY-016	Arduino
R	Pin 11
B	Pin 10
G	Pin 9
-	GND

## **INPUT:**



## **CONNECTIONS ARDUINO:**

LED Red = [Pin 10]  
LED Green = [Pin 11]  
LED Blue = [Pin 12]  
Sensor GND = [Pin GND]

## **EXAMPLE SENSOR CODE:**

```
intredpin = 11; // select the pin for the red LED  
intbluepin =10; // select the pin for the blue LED  
intgreenpin =9; // select the pin for the green LED  
intval;  
void setup() {  
pinMode(redpin, OUTPUT);  
pinMode(bluepin, OUTPUT);  
pinMode(greenpin, OUTPUT);  
Serial.begin(9600);
```

```
}

void loop() {
    for(val = 255; val> 0; val--)
    {
        analogueWrite(11, val);
        analogueWrite(10, 255 - val);
        analogueWrite(9, 128 - val);
        Serial.println(val, DEC);
        delay(5);
    }
    for(val = 0; val< 255; val++)
    {
        analogueWrite(11, val);
        analogueWrite(10, 255 - val);
        analogueWrite(9, 128 - val);

        Serial.println(val, DEC);
        delay(5);
    }
}
```

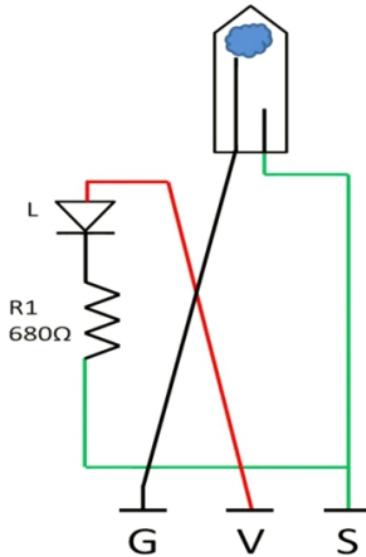
## KY-017 MERCURY OPEN OPTICAL MODULE



KY 017. This is a module itself. So how about we begin. This module has incorporated onboard on mercury switch, one resistor and one drove the resistor utilized in this mode It's 680 ohms and the fundamental explanation behind utilizing the resistor is to restrict cash relating inside the module. As such, to keep the current from consuming our module, the LED illuminates when the mercury switch shorts are associated with the pin that is inside the sensor.

Presently, I will show how these parts are associated together. There we have the actual sensor and obviously, the resistor r1 and the LED on there as you can perceive how the pins of the module are associated on this board. Thus, we have web the progression of the voltage we have in dark, the progression of the ground. Furthermore, we have in green the progression of the sign.

The mercury switch model has the knowledge to conduct and mercury fluid in a typical express, the module will give a sign that will be high. In any case, when the irregular fluid shorts, or interfaces the two sheets together, we should flag that the module will give will be low, making the LED in the module light up. So the LED will be on just when our mercury fluids contact the two contacts. So we should discuss the pins. The pins in this module are three we have the ground pin with the less significant, the voltage pin is in the centre.



Furthermore, we have a pin of the sign with an S sign. So we should discuss the sign. This module gives a digital signal, so we can associate the pins of the sign with any advanced ports of various microcontroller sheets like Arduino or Raspberry Pi.

#### Specifications

Operating Voltage	3.3V to 5.5V
-------------------	--------------

#### KY-017 Connection Diagram

KY-017	Arduino
S	Pin 3
middle	+5V
-	GND

## CONNECTIONS ARDUINO:

LED +        =        [Pin 13]

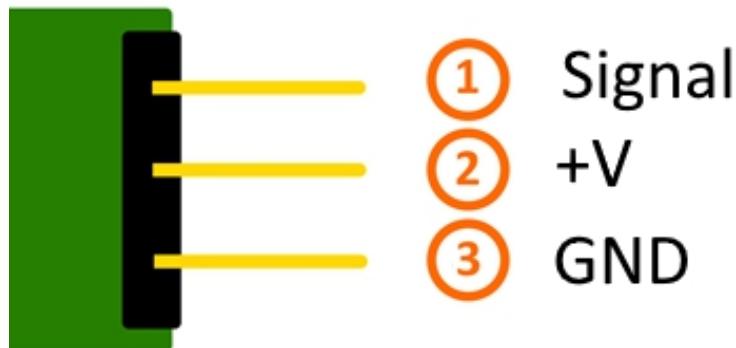
LED - = [Pin GND]

Sensor Signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]

## INPUT:



## EXAMPLE SENSOR CODE:

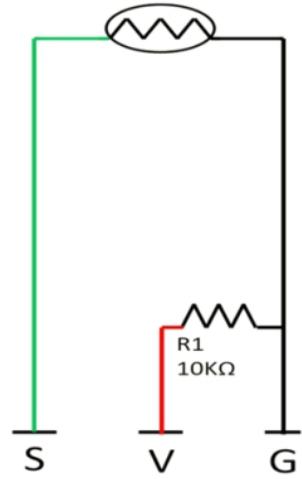
```
intled_pin = 13; // Define the LED interface
intswitch_pin = 3; // Definition of mercury tilt switch sensor interface
intval; // Defines a numeric variable
void setup()
{
    pinMode(led_pin, OUTPUT);
    pinMode(switch_pin, INPUT);
}
void loop()
{
```

```
val = digitalRead(switch_pin); // check mercury switch state
    if(val == HIGH)
    {
        digitalWrite(led_pin, HIGH);
    }
    else
    {
        digitalWrite(led_pin, LOW);
    }
```

## KY-018 PHOTO RESISTOR MODULE



The photograph resistor module is known as KY 0 18. This is simply the module. So we should begin. This module has coordinated on board photograph resistor into one resistor. The resistor utilized in this module it's 10-kilo ohm. What's more, the fundamental purpose behind utilizing the resistor is to restrict cash gathering inside the module. At the end of the day, to keep current from consuming our module. Presently I will show how these parts are associated together. There we have the actual sensor, and obviously, the resistor r1. On the right, you can perceive how the So, the module is associated on this board. Along these lines, we have channelled the progression of the voltage, we have in dark, the progression of a ground. Furthermore, we have in-game the progression of the sign is Easter's are otherwise called light ward resistors. Furthermore, they are made by semiconductor materials, their obstruction will diminish when we increment the light power and the inverse, their opposition will increment when we decline the light force.



In obscurity, a photograph resister can have opposition as high as a few super ohms while in the light of photograph resister can have an obstruction as low as barely any hundred ohms we should discuss the pins. The pins in this module are three, we have the ground pin with a short sign, the voltage pin is in the centre. What's more, obviously we have the pin of the sign with an S sign. So we should discuss the sign. This module gives an analogue signal. So we can associate the pin of the sign with any analogue port of various microcontroller sheets like Arduino or Raspberry Pi.

#### Specifications

Operating Voltage	3.3V to 5.5V
-------------------	--------------

#### KY-017 Connection Diagram

KY-017	Arduino
S	Pin 3
middle	+5V
-	GND

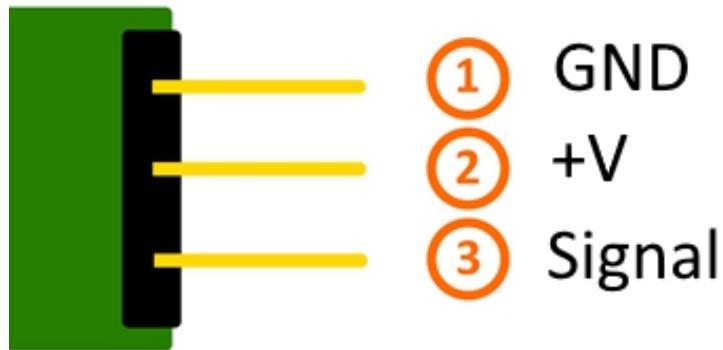
## CONNECTIONS ARDUINO:

Sensor GND = [Pin GND]

Sensor +V = [Pin 5V]

Sensor Signal = [Pin A5]

## INPUT:



## EXAMPLE SENSOR CODE:

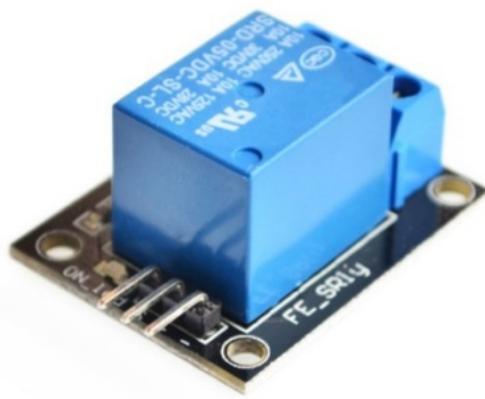
```
intsensorPin = 2; //define analogue pin 2
int value = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    value = analogueRead(sensorPin);
    Serial.println(value, DEC); // light intensity
    // high values for bright environment
```

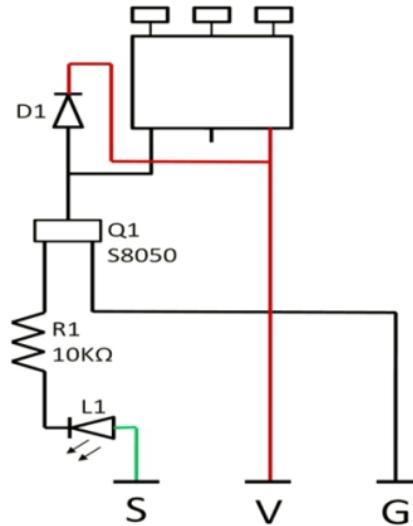
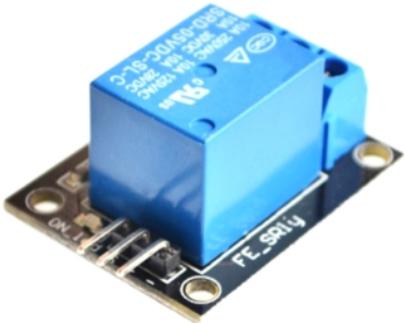
```
// low values for dark environment  
delay(100);
```

# KY-019 5V RELAY MODULE



The transfer module is known as KY 019. This is a module itself. So we should begin. This module is incorporated ready. One drove one resistor One eating regimen and one semiconductor. The resistor utilized in this module is 150 M. Also, the fundamental explanation behind utilizing the resistor is to restrict cash examining inside the module. The LED illuminates to show if the module is working or not. Presently, I will show how these segments are associated together. There we have the hand-off itself and we can see the LED lone, the resistor r1, the semiconductor q1 and obviously, the diodes The one on the correct you can perceive how the piece of the module I associated on this board, we have bay the progression of the voltage we have in dark, the progression of the ground and we have in green the progression of a signal relays are switches that open and close insider facts electro precisely or electronically. The transfers control

one electrical circuit by opening and shutting contacts in another circuit. As handed-off outlines show when a hand-off contact is typically open, there is an open contact proprietor transfer it's not invigorated. At the point when the hand-off contact is regularly shut, there is a shut contact when you lay it's not invigorated. Regardless, applying electrical flow to the contact will change their state. transfers are for the most part utilized



to switch more modest nurseries in the support to stop and don't normally control power burning-through gadgets. Presently the fewer transfers can handle bigger voltages and sovereigns by having an intensifying impact as a result of some voltage concerned her ribbon curl case out in a huge voltage being exchanged by the contacts. We should discuss the pins. The pins in this module are three, we have the ground pin with a short sign, the voltage pin is in the centre. Furthermore, we have the pin of the sign with the S sign. So we should discuss the sign. This module takes a computerized signal so we can associate the torment of the sign with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

## Specifications

TTL Control Signal	5VDC to 12VDC (some boards may work with 3.3)
Maximum AC	10A 250VAC
Maximum DC	10A 30VDC
Contact Type	NC and NO
Dimensions	27mm x 34mm [1.063in x 1.338in]

## KY-019 Connection Diagram

For the DC part of the circuit connect S (signal) to pin 10 on the Arduino, also connect the Power line (+) and ground (-) to +5 and GND respectively.

On the AC side connect your feed to Common (middle contact) and use NC or NO according to your needs.

NO (Normally Open) will get power when (S) is high, NC (Normally Closed) gets disconnected when (S) is high.

KY-019	Arduino	AC Device
S	Pin 10	
+	+5V	
-	GND	
NC		
Common		Feed In
NO		Feed Out

## CONNECTIONS ARDUINO:

Sensor - = [Pin GND]

Sensor + = [Pin 5V]

Sensor Signal = [Pin 10]

## INPUT:



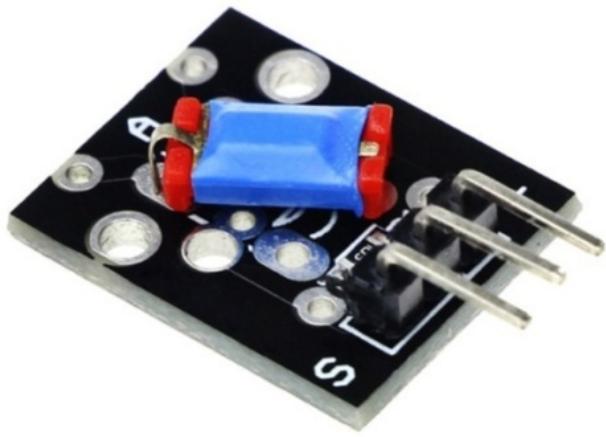
## EXAMPLE SENSOR CODE:

```
int relay = 10; //Pin 10
```

```
void setup()
{
    pinMode(relay,OUTPUT);      // Define the port attribute
as output
}
void loop()
{
    digitalWrite(relay,HIGH);    // turn the relay ON
    // [NO] is connected to feed
    // [NC] is not connected to feed
    delay(1000);

    digitalWrite(relay,LOW);   // turn the relay OFF
    // [NO] is not connected to feed
    // [NC] is connected to feed
    delay(1000);
```

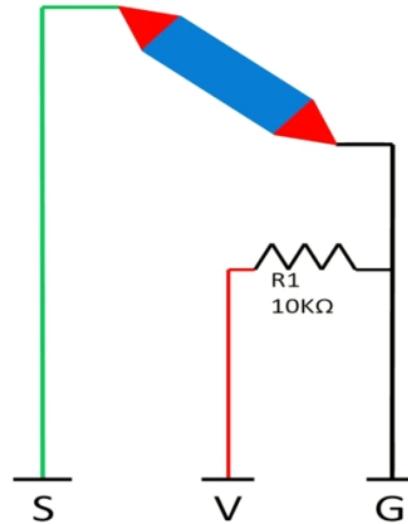
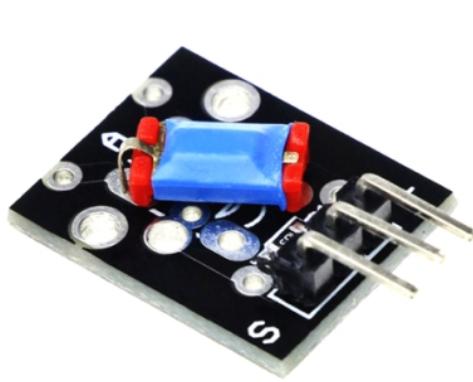
## KY-020 TILT SWITCH MODULE



The slant switch module is known as KY 0 20. This is a module itself. So how about we begin. This module has coordinated ready, a slant switch, and one resistor, the resistor utilizing this module it's on and the fundamental explanation behind utilizing the resistor is to restrict money-gathering inside the module.

As such to keep current from consuming our modular. Presently, I will show how these parts are associated together. There we have the actual sensor, and obviously, the resistor r1. On the right, you can perceive how the piece of the module is associated with this board. So we have gulf the progression of the voltage we have in dark, the progression of the ground, and obviously, we have in-game the progression of the sign. This sensor has inside a metal egg bowl that opens or shuts the holy advance. gasping.

The point of the slant the sensor's regularly have, yet to under sensor, it's shifted in that way that the ball shorts to direct the sensor turns on. Indeed, even that the sensor distinguishes changes when it's shifted, it is in reality extremely essential and can't quantify the point of the slant. We should discuss the pins. The pins in this module



are three, we have the ground pin with a minus sign, the voltage being is in the middle. And of course, we have the pin of the signal with the S sign. So let's talk about the signal. This module gives a digital signal so we can connect the pins of the signal with any digital ports of different microcontroller boards, like Arduino or Raspberry

#### Specifications

Operating Voltage	3.3V to 5v
Output Type	Digital

#### KY-020 Connection Diagram

KY-020	Arduino
S	2
middle	+5V
-	GND

Pi.

## **CONNECTIONS ARDUINO:**

LED + = [Pin 13]

LED - = [Pin GND]

Sensor Signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]

## **INPUT:**

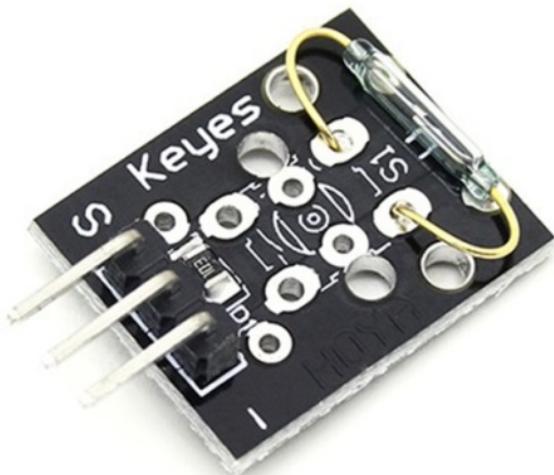


## **EXAMPLE SENSOR CODE:**

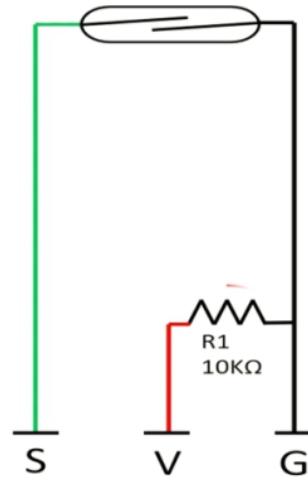
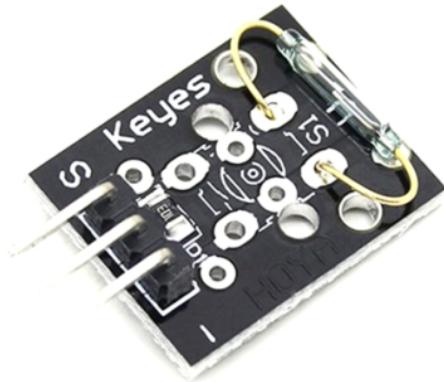
```
inttiltPin = 2; // pin number for tilt switch signal  
intledPin = 13; // pin number of LED  
inttiltState = 0; // variable for reading the tilt switch status  
void setup() {  
    pinMode(ledPin, OUTPUT); // set the LED pin as output  
    pinMode(tiltPin, INPUT); // set the tilt switch pin as input  
}  
void loop(){
```

```
// get the tilt switch state  
tiltState = digitalRead(tiltPin);  
// check if tilt switch is tilted.  
if (tiltState == HIGH) {  
    digitalWrite(ledPin, HIGH);  
}  
else {  
    digitalWrite(ledPin, LOW);
```

## KY-021 MINI MAGNETIC REED MODULES



The attractive read switch is known as KY 0 21. This is a module itself so we should begin. the module has a coordinated ready or understood switch and one resistor, the resistor utilized in this module is the tangka arm and the principle explanation behind utilizing the resistor is to restrict current circling inside the module. As such, to keep current from consuming our model. Presently, I will show how these parts are associated together. There we have the actual sensor and obviously, the resistor r1. On the correct, you can perceive how the piece of the module is associated on this board. In this way, we have gulf the progression of the voltage we have in dark the progression of ground And obviously, we have England the progression of the sign. I read switch needs to lead that is produced using ferromagnetic materials and seal within the slim glass envelope loaded up with a lifeless gas to keep them liberated from residue and earth. Now and again the glass has a special raised area projecting of plastic for considerably more prominent insurance. In the ordinary condition of contact doesn't contact one another. So the sign that the module will give will be high, however, when an attractive field will come nearer to the sensor, this will constrain the conductor to contact each other on account of their government



attractive materials which can get handily polarized under an attractive field to this module will give a sign that will below, for the most part, the two agreements more Make a level Barrel zone of behaviours with one other because that needs to expand the life and dependability of the switch. We should discuss the pins. The pins in this module are three, we have the ground pin with a less significant, the voltage pin is in the centre. Furthermore, we have the pin of the sign with the S sign. So we should discuss the sign. This module gives an advanced sign. So we can interface the pins of the sign with any advanced ports of various microcontroller sheets like Arduino or Raspberry Pi.

## Specifications

Operating Voltage	3.3V to 5v
Output Type	Digital

KY-021 Connection Diagram

KY-021	Arduino
S	2
middle	+5V
-	GND

## CONNECTIONS ARDUINO:

LED + = [Pin 13]

LED - = [Pin GND]

Sensor Signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]

## INPUT:

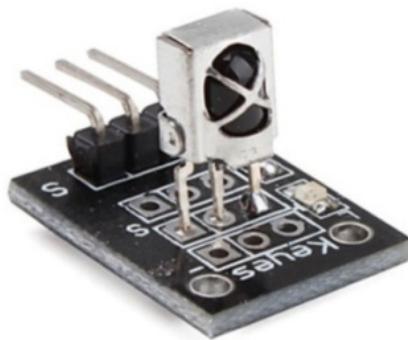


## EXAMPLE SENSOR CODE:

```
int led = 13; // LED pin
int reelSwitch = 2; // magnetic sensorpin
int switchState; // variable to store reel switch value
void setup()
{
    pinMode (led, OUTPUT);
    pinMode (reelSwitch, INPUT);
}
void loop()
{
    switchState = digitalRead(reelSwitch); // read the value of digital
    interface 2 and assign it to switchState

    if (switchState == HIGH) // when the magnetic sensor detect a
    signal, LED is flashing
    {
        digitalWrite(led, HIGH);
    }
    else
    {
        digitalWrite(led, LOW);
    }
}
```

# KY-022 INFRARED SENSOR RECEIVER MODULE



This module is utilized along with KY 0 22 known as the infrared recipient module. So we should discuss the pins. The pins in this module are three, we have the ground pin with the less significant, the voltage pin is in the centre. What's more, obviously we have the pin of the sign with an S sign. So we should discuss the sign. This module takes an advanced sign, so we can interface the pin of the sign with any computerized part of various microcontroller sheets like Arduino or Raspberry Pi.

### KY-022 Specifications

Operating Voltage	2.7 to 5.5V
Operating Current	0.4 to 1.5mA
Reception Distance	18m
Reception Angle	$\pm 45^\circ$
Carrier Frequency	38KHz
Low Level Voltage	0.4V
High Level Voltage	4.5V
Ambient Light Filter	up to 500LUX

### KY-022 Connection Diagram

KY-012	Arduino
S	Pin 11
middle	+5V
-	GND

## CONNECTIONS ARDUINO[RECEIVER]:

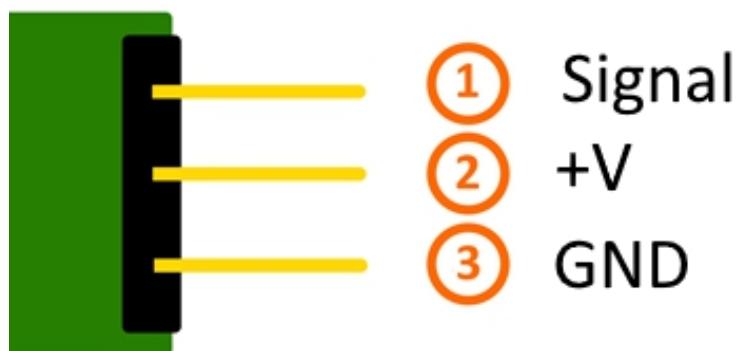
### KY-022

Signal = [Pin 11]

+V = [Pin 5V]

GND = [Pin GND]

### INPUT:



## **EXAMPLE SENSOR CODE:**

```
#include <IRremote.h>

int RECV_PIN = 11; // define input pin on Arduino
IRrecv irrecv(RECV_PIN);
decode_results results; // decode_results class is defined in
IRremote.h

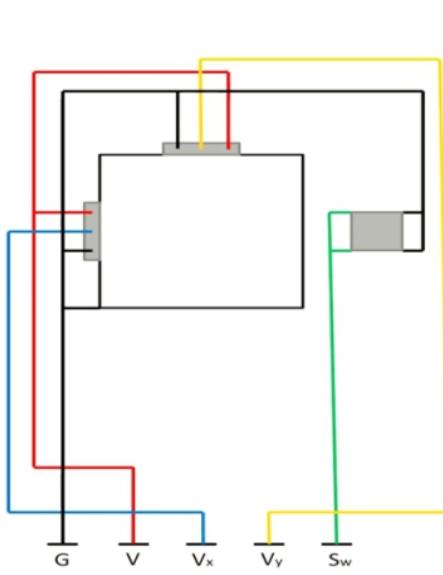
void setup() {
    Serial.begin(9600);
    irrecv.enableIRIn(); // Start the receiver
}

void loop() {
    if (irrecv.decode(&results)) {
        Serial.println(results.value, HEX);
        irrecv.resume(); // Receive the next value
    }
    delay (100); // small delay to prevent reading errors
}
```

## **KY-023 XY-AXIS JOYSTICK MODULE**



The double access module is known as KY 0 23. This is a module it's So, we should begin. This module is incorporated ready, just a single joystick and one catch and that's it. Presently I will show how these parts are associated together. There we have the actual joystick. What's more, we can see the catch of the module. On the correct, you can perceive how the beans of the module are associated on this board. We have in dark, the progression of the ground, we have associate, the progression of the voltage, we have in blue, the progression of the simple being of x hub, we have in orange, the simple pin of Y hub. Also, we have occupied the progression of the sign pin. The PS two style joystick it's a ranch worked gadget that when put into consideration to sees offers a helpful method of getting administrator inputs. It's generally comprised of two focuses so meters The press-button switch, the two potentiometers demonstrate which heading the potentiometer is being pushed in the non-worked mode, the potentiometer is in the centre, so the two resistors are equivalent and that the voltage will be similarly part to the two resistors. If one of the tomahawks changes like the x hub, for instance, the estimation of the resistor will change. For instance, estimation of resistor one will exhibits, along these lines, will affect the tumble to the resistor or the estimation of the resistor



one will fall and the estimation of the resistor two will raise as per the division of the resistor esteems. You can quantify a particular voltage esteem bit resistors and find the situation of the pivot the switch imparts a low sign when a joystick handle is squeezed. We should discuss the pins. The pins in this module are fine. We have the ground pin with G and D sign the voltage pin with in addition to five sign to simple pins with a V r x and v y sign. Furthermore, the computerized torment named as W. So we should discuss the sign. This module provides for a simple sign and one advanced sign. So we can interface the simple pin of the sign with any simple board and the advanced paint of the seat With any computerized part of various microcontroller sheets like Arduino or Raspberry Pi

#### Specifications

Operating Voltage	3.3V to 5V
Board Dimensions	2.6cm x 3.4cm [1.02in x 1.22in]

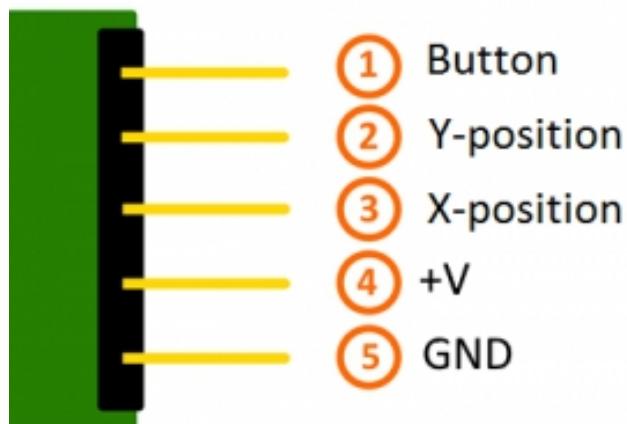
Arduino KY-023 Connection Diagram

KY-023	Arduino
GND	GND
+5V	5V
VRx	A0
VRy	A1
SW	7

## CONNECTIONS ARDUINO:

- Button = [Pin 3]  
Y-Position = [Pin A1]  
X-Position = [Pin A0]  
Sensor +V = [Pin 5V]  
Sensor GND = [Pin GND]

## INPUT:



## **EXAMPLE SENSOR CODE:**

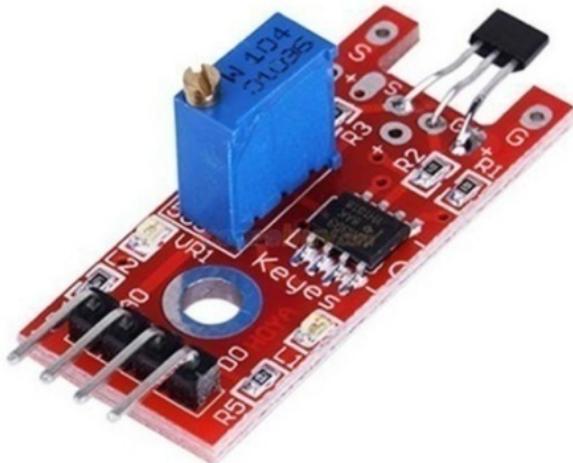
```
int value = 0;

void setup() {
    //pinMode(A0, INPUT);
    //pinMode(A1, INPUT);
    pinMode(7, INPUT_PULLUP); //set pin 7 as an input and
enable the internal pull-up resistor
    Serial.begin(9600);
}

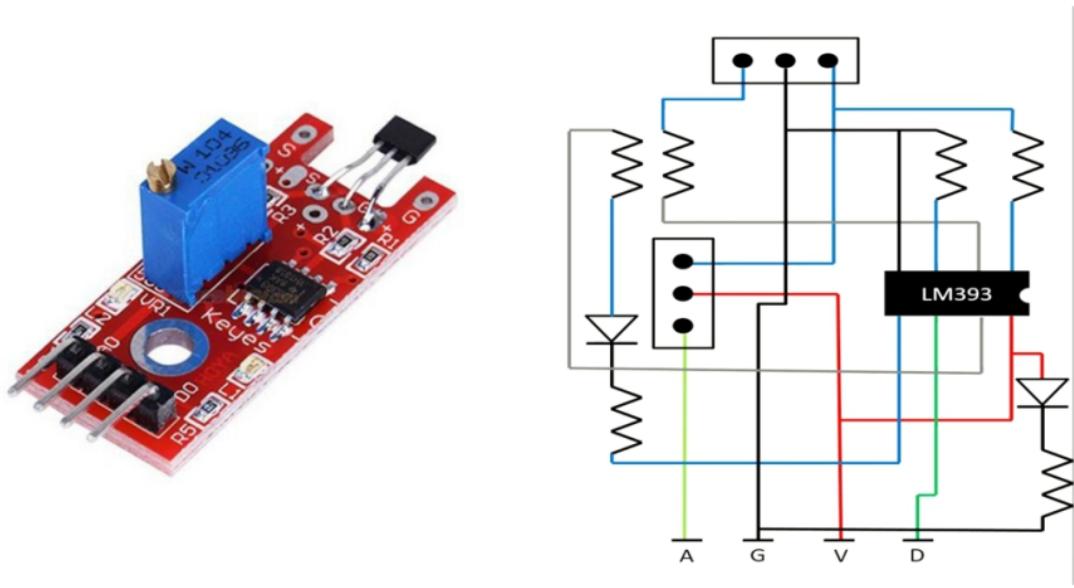
void loop() {
    value = analogueRead(A0);      // read X axis value [0..1023]
    Serial.print("X:");
    Serial.print(value, DEC);
    value = analogueRead(A1);      // read Y axis value [0..1023]
    Serial.print(" | Y:");
    Serial.print(value, DEC);
    value = digitalRead(7);      // read Button state [0,1]
    Serial.print(" | Button:");
    Serial.println(value, DEC);

    delay(100);
}
```

**KY-024 LINEAR MAGNETIC HALL  
SENSORS**



Learn our entire attractive sensor known as KY 0 24. This is a module itself so how about we begin. This module is coordinated onboard one attractive hold sensor, one potentiometer one central processor, six resistors and two LED is the resistor r1. Utilizing this module, it's Thank you arms, the resistor R two, it's 100-kilo ohms. The resistor R three is 150 ohms. Their resistor R four's won KY on the resistor R five, it's won KY on and the resistor R six is 100-kilo ohm. Also, the primary explanation behind utilizing the resistor is to restrict current circling inside the module. All in all, to squeeze Current from consuming our module the LED lone lights up to show if the module it's working or not at any point drove I two lights up just a single the sensor has identified the attractive field. Presently I will show how these parts are associated together. There we have the actual sensor and obviously, the six is Easter's and the two LVDS abrogate. You can perceive how the piece of the module symbol noted on this board, we have in dark, the progression of the ground, we have in the red, the progression of the voltage, we have in a light green, the progression of the simple sign and we have in green, the progression of the computerized signal. This sensor is utilized to label the attractive field It can respond within the sight of the attractive field. It has a potentiometer to change the affectability of the sensor, and



it gives both simple and computerized yields. The advanced yield goes about as a switch that will turn on or off when the magnet is close. Then again, the simple yield can quantify the extremity and the general strength of the attractive field. At the point when an attractive field is identified by the sensor, the LED I two will illuminate. We should discuss the pins. The pins in this module are four. We have the ground pin with G sign. We have the voltage pin with the in addition to sign.

We have the simple pin with A0 sine. What's more, obviously we have the computerized pin with the 0 sign. So we should discuss the sign. This module gives two signs, we can interface the pin of the simple sign with any simple board and the agony of the advanced sign with any computerized port of various microcontroller sheets like Arduino or Raspberry Pi.

## Specifications

Operating Voltage	2.7V to 6.5V
Sensitivity	1.0 mV/G min., 1.4 mV/G typ., 1.75 mV/G max.
Board Dimensions	1.5cm x 3.6cm [0.6in x 1.4in]

Arduino KY-024 Connection Diagram

KY-024	Arduino
A0	A0
G	GND
+	5V
D0	3

## CONNECTIONS ARDUINO:

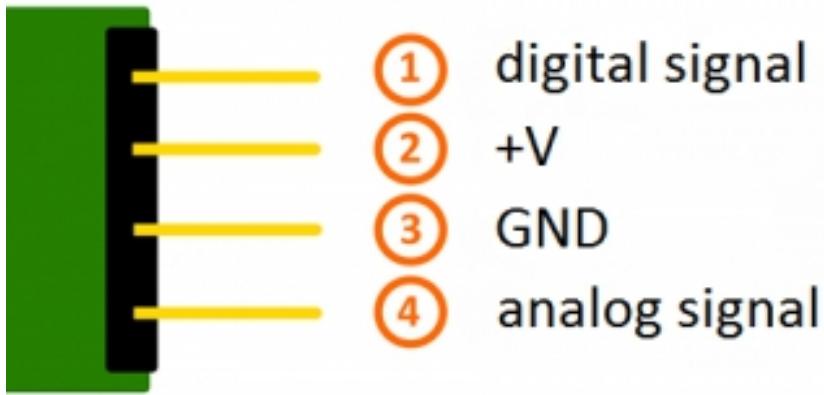
digital signal = [Pin 3]

+V = [Pin 5V]

GND = [Pin GND]

analogue signal = [Pin 0]

## INPUT:



## EXAMPLE SENSOR CODE:

```

int led = 13 ; // LED on arduino
int digitalPin = 3; // linear Hall magnetic sensor digital interface
int analoguePin = A0; // linear Hall magnetic sensor analogue
interface
int digitalVal ; // digital readings
int analogueVal; // analogue readings

void setup ()
{
pinMode (led, OUTPUT);
pinMode (digitalPin, INPUT);
//pinMode(analoguePin, INPUT);
Serial.begin(9600);
}

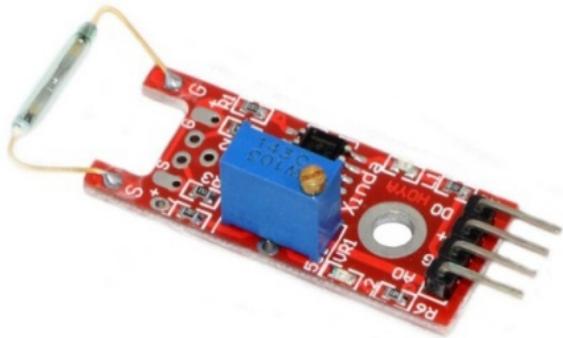
void loop ()
{

```

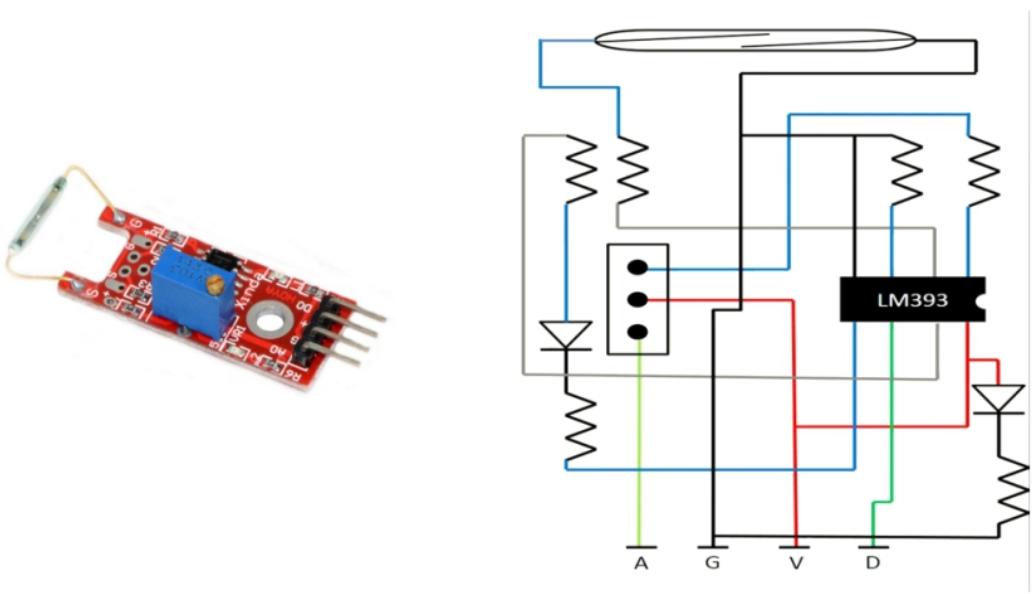
```
// Read the digital interface
digitalVal = digitalRead(digitalPin) ;
if (digitalVal == HIGH) // When magnetic field is present, Arduino
LED is on
{
digitalWrite (led, HIGH);
}
else
{
digitalWrite (led, LOW);
}
// Read the analogue interface
analogueVal = analogueRead(analoguePin);
Serial.println(analogueVal); // print analogue value

delay(100);
```

## KY-025 REED MODULE



The read switch module is known as KY 0 25. This is a module itself so how about we begin. This module is incorporated onboard one eight switches, one potentiometer one microprocessor six resistors and two LVDS. The resistor r1 utilized in this module is thank you um, there is this there is two use in this module. It's 100-kilo ohms vertices there are three, it's 150 ohms. There is this there are four, it's the one-kilo ohm, there is this there are five, it's the one-kilo ohm, at that point the resistor R six is 100-kilo ohm. Also, the principle explanation behind utilizing the resistors is to restrict current coursing inside the module. All in all, to keep current from consuming our module, the LED lone lights up to show if the module is working appropriately or not. Also, drove I to use in this module illuminates just when an attractive field has drawn closer to the sensor. We will show how these segments are associated together. There we have the actual sensor. Also, six is Easter's and to LVDS have the correct you can perceive how the piece of the module is no more. Did on this board. Along these lines, we have in dark the progression of the ground, we have bay the progression of the voltage, we



have in light green, the progression of the simple sign. Also, we have in green the progression of the computerized signal. Peruse switch needs to direct that are produced using ferromagnetic materials and seal within the slight glass envelope loaded up with lifeless gas to keep them liberated from residue and earth. Once in a while, the glass has an adjust projecting of plastic for much more prominent assurance. In the ordinary condition of contact doesn't contact one another. Along these lines, the sign that the module will give will be high, yet when an attractive field will come nearer to the sensor, this will drive the conductor to contact each other in light of ferromagnetic materials which can get effectively charged under an attractive field, this module will give a sign that will below. By and large, the two behaviours move and make a level equal region of behaviours with one other because that needs to broaden the life and unwavering quality of the switch. We should discuss the beans

. The beans in this module are four, we have the ground being with a G sign, the voltage pin with an or more sign, the simple pin with an o sign and the computerized pin with the Do sign. So we should discuss the sign. This module gives two signs, we can interface the agony of the simple sign with any analogue port The torment of the advanced sign with any computerized port of various microcontroller sheets like Arduino or Raspberry Pi.

## **CONNECTIONS ARDUINO:**

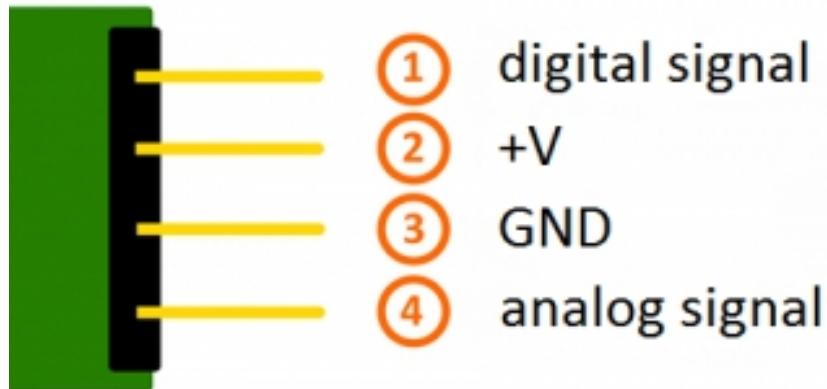
digital signal = [Pin 3]

+V = [Pin 5V]

GND = [Pin GND]

analogue signal = [Pin 0]

## **INPUT:**



## **EXAMPLE SENSOR CODE:**

```
int led = 13; // define the LED pin  
int digitalPin = 3; // KY-025 digital interface  
int analoguePin = A0; // KY-025 analogue interface  
int digitalVal; // digital readings  
int analogueVal; //analogue readings
```

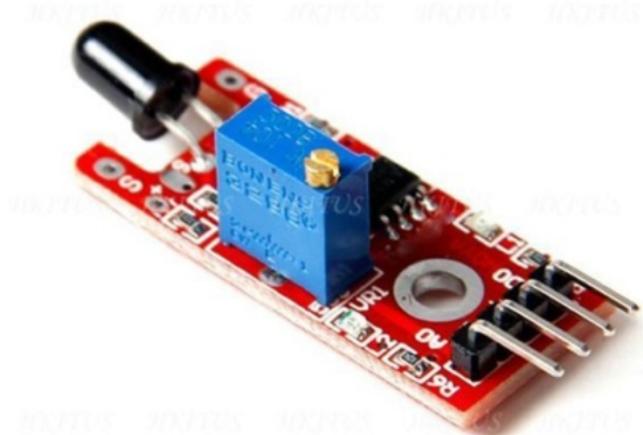
```
void setup()
{
pinMode(led, OUTPUT);
pinMode(digitalPin, INPUT);
//pinMode(analoguePin, OUTPUT);
Serial.begin(9600);
}

void loop()
{
// Read the digital interface
digitalVal = digitalRead(digitalPin);
if(digitalVal == HIGH) // if magnetic field is detected
{
digitalWrite(led, HIGH); // turn ON Arduino's LED
}
else
{
digitalWrite(led, LOW); // turn OFF Arduino's LED
}

// Read the analogue interface
analogueVal = analogueRead(analoguePin);
Serial.println(analogueVal); // print analogue value to serial

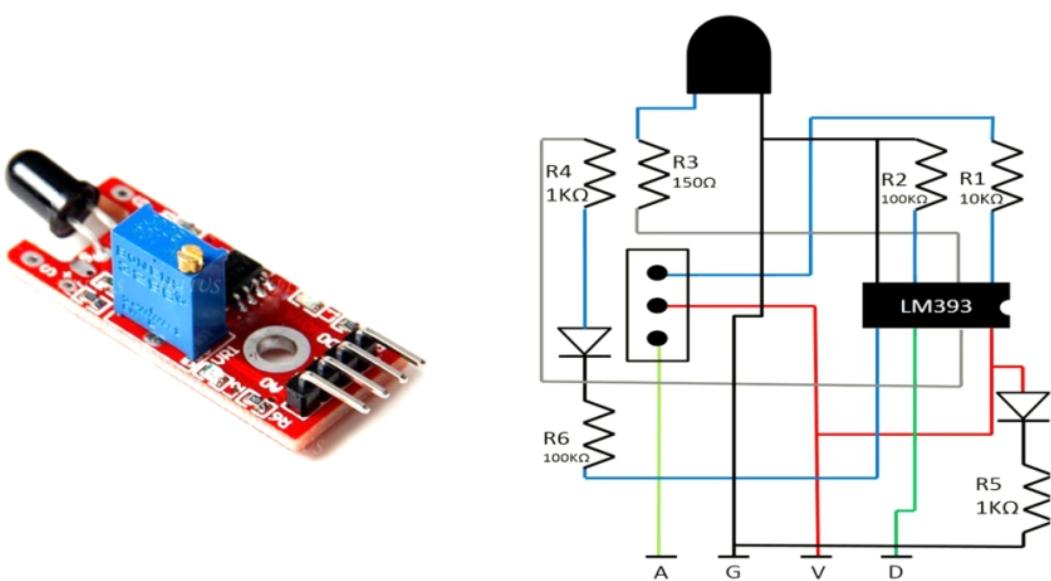
delay(100);
```

# KY-026 FLAME SENSOR MODULE



The fire sensor is known as KY 0 26. This is simply the module. So we should begin.

This module has coordinated onboard AI fire sensor, one potentiometer one CPU 60 resistors and two drove the resistor r1 utilized in this module is thank you ohms the resistor R two is 100-kilo ohms. There is there are three is 150 ohms, there is Easter r four is one kilo on their resistor R five is Wankel's arm and the resistor R six is 100-kilo ohms



The primary purpose behind utilizing the resistor is to restrict the current flowing inside the module. As such, to keep the current from consuming our module, the LED lone lights up to show if the module is working appropriately or not. Presently, I will show how these segments are associated together. There we have the actual sensor and obviously, the six resistors and the two LEDs on the correct you can see all the piece of the module symbol on this board. We have in dark, the progression of the ground, we have in the red, the progression of the voltage, we have in light green, the progression of the analogue signal and we have in green, the progression of the computerized signal the fire sensor is utilized to identify fires or other light sources that have a frequency of 760 nanometres to thousand hundred nanometres.

Close to infrared beam fire indicator, otherwise called a visual fire finder utilize fire acknowledgement innovation to affirm fire by investigating close to infrared radiation utilizing a charge-coupled gadget. A close to the infrared sensor is particularly ready to screen fire phenol Mia without an excessive amount of he depletes from water or water fume. Viral electric sensor working at this frequency can be generally modest. Different channel or huge cell exhibit sensor checking flares in the close to the infrared band is ostensibly the most dependable technology available for the discovery of

flames, light outflow from a fire structure a picture of the fire at a specific moment advanced picture preparing can be used to perceive fire through breaks down of a video made from the close to infrared pictures. This module can identify the fire, however can likewise take the standard light.

The affectability of this sensor is customizable and has a steady presentation. These sensors are utilized for more limited reach fire identification also. The sensor starts to distinguish fire a good ways off of 0.8 meters. If the firepower is high, the location trouble will increase lets discussion about the pins. The pins in this module are four, we have the ground pin with G sign, we have the voltage pin with the in addition to sign, we have the simple pin with a 0 sign. What's more, obviously we have the computerized pin with the 0 sign. So we should discuss the sign. This module gives two signs, we can interface the pin of the simple sign with any simple port and the pin of the computerized signal with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

#### Specifications

Operating Voltage	3.3V to 5.5V
Infrared Wavelength Detection	760 nm to 1100 nm
Sensor Detection Angle	60°
Board Dimensions	1.5cm x 3.6cm [0.6in x 1.4in]

#### Arduino KY-026 Connection Diagram

KY-026	Arduino
A0	A0
G	GND
+	5V
D0	2

## CONNECTIONS ARDUINO:

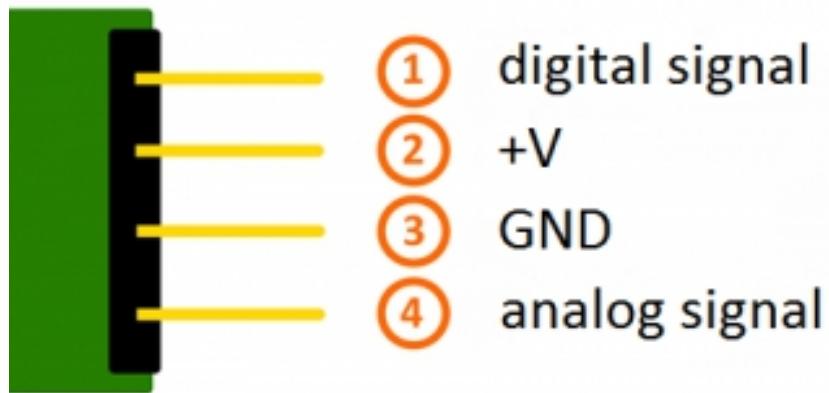
digital signal = [Pin 3]

+V = [Pin 5V]

GND = [Pin GND]

analogue signal = [Pin 0]

## INPUT:



## EXAMPLE SENSOR CODE:

```
int led = 13; // define the LED pin
int digitalPin = 2; // KY-026 digital interface
int analoguePin = A0; // KY-026 analogue interface
int digitalVal; // digital readings
int analogueVal; //analogue readings

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(digitalPin, INPUT);
```

```
//pinMode(analogPin, OUTPUT);
Serial.begin(9600);
}

void loop()
{
// Read the digital interface
digitalVal = digitalRead(digitalPin);
if(digitalVal == HIGH) // if flame is detected
{
digitalWrite(led, HIGH); // turn ON Arduino's LED
}
else
{
digitalWrite(led, LOW); // turn OFF Arduino's LED
}

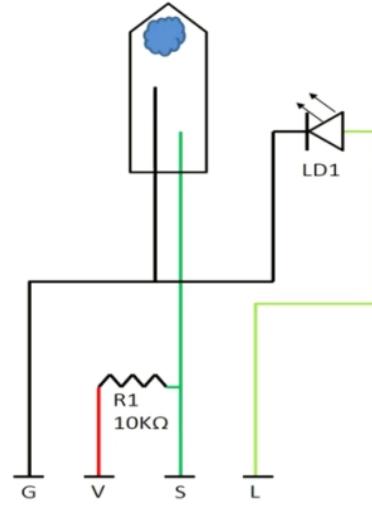
// Read the analogue interface
analogueVal = analogueRead(analoguePin);
Serial.println(analogueVal); // print analogue value to serial
delay(100);
```

## KY-027 MAGIC LIGHT CUP MODULE



The enchantment light crab module is known as KY0 27. This is simply the module. So we should begin. This module has coordinated onboard one mercury switch, one is Easter and one drove Easter is utilizing this module is thank you so be it The fundamental purpose behind utilizing the resistor is to restrict current coursing inside the module. At the end of the day, to keep current from consuming our model, the LED we can turn on or off as we wish. Presently I will show every one of these segments are associated together there we have the mercury switch itself and obviously, there is these there are one and the LED LD one on assortment you can see all the pins of the module symbol on this board. We have in dark, the progression of the gong, we have in red, the progression of the voltage we have in green, the progression of the mercury switch sign and we have in a light increase the progression of the LED Signal Mercurius which has inside two contacts and mercurial fluid. In typical express, the sign that This module will give will be high. Be that as it may, when the Mercurial fluid interface or short two pins together, the

the sign will change excessively low. On other hand, the LED of the module we can turn on



or then again off as we wish, because the LED takes a sign that comes straightforwardly from any microcontroller. We should discuss the pins. The pins in this module are four. We have the ground pin with a G sign, the voltage pin with an or more sign. The mercury switch signal has the S sign and the Led signal has the L sign. So we should discuss the sign. This module gives one computerized sign and takes one advanced sign. So we can associate the pins of the signs with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

## Specifications

Operating Voltage	3.3V to 5.5V
Board Dimensions	1.5cm x 3.6cm [0.6in x 1.4in]

## Arduino KY-027 Connection Diagram

KY-027 (A)	Arduino
G	GND
+	5v
S	8
L	9

KY-027 (B)	Arduino
G	GND
+	5v
S	7
L	6

## CONNECTIONS ARDUINO:

LED + = [Pin 13]

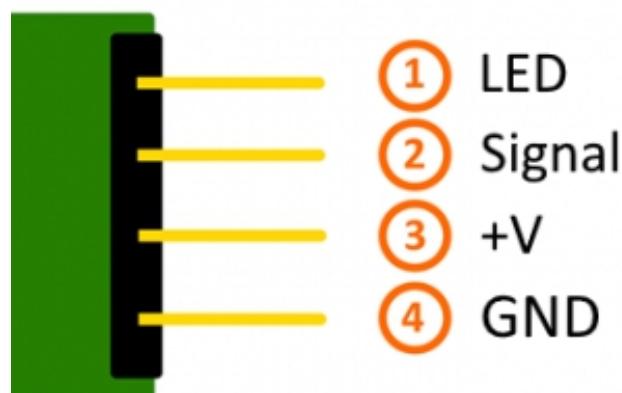
LED - = [Pin GND]

Sensor signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]

## INPUT:

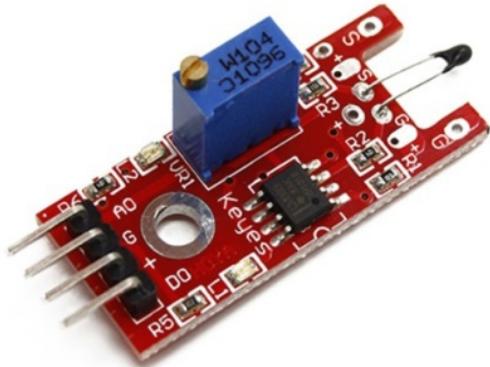


## EXAMPLE SENSOR CODE:

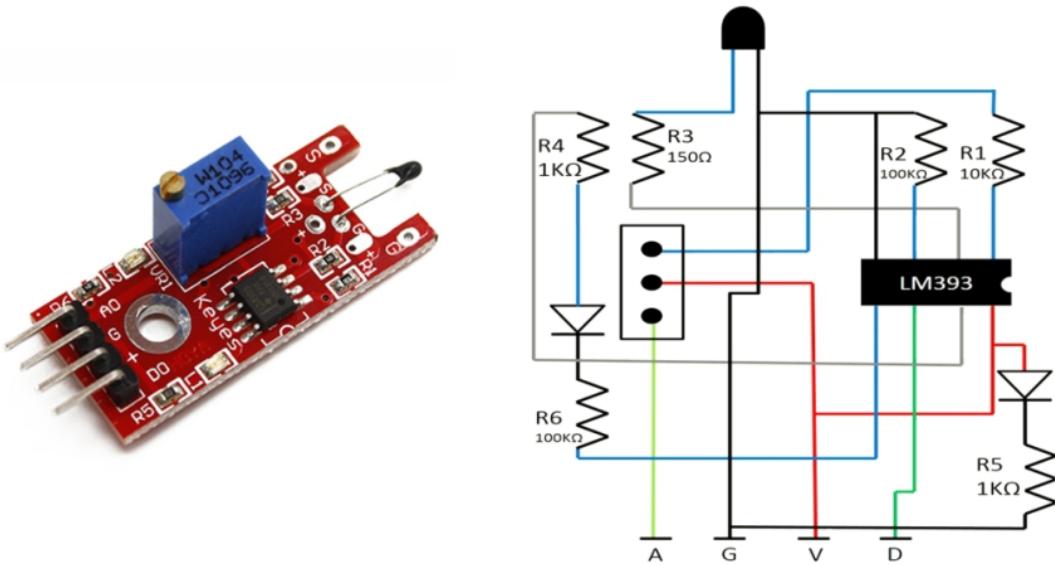
```
int ledPinA = 9;  
int switchPinA = 8;  
int switchStateA = 0;  
int ledPinB = 6;  
int switchPinB = 7;  
int switchStateB = 0;  
int brightness = 0;  
  
void setup()  
{  
    pinMode(ledPinA, OUTPUT);  
    pinMode(ledPinB, OUTPUT);  
    pinMode(switchPinA, INPUT);  
    pinMode(switchPinB, INPUT);  
}  
  
void loop()  
{  
    switchStateA = digitalRead(switchPinA);  
    if (switchStateA == HIGH && brightness != 255)  
    {  
        brightness++;  
    }  
    switchStateB = digitalRead(switchPinB);  
    if (switchStateB == HIGH && brightness != 0)
```

```
{  
brightness --;  
}  
analogueWrite(ledPinA, brightness); // A slow fade out  
analogueWrite(ledPinB, 255 - brightness); // B slow bright up  
delay(20);
```

## KY-028 TEMPERATURE SENSOR MODULE



The temperature sensor module is known as KY 0 28. This is simply the module. So how about we begin. This module has coordinated onboard one temperature sensor one potentiometer one CPU six resistors and two LED is the resistor r1 utilized in this module each Thank you um, there is these there are two it's 100-kilo ohms. There is this there are three is 150 ohms, there is Easter RF For every one-kilo ohm, there is these there are five it's the one-kilo ohm, there is these there are six is 100-kilo ohms and the principle explanation behind utilizing different diesters is to restrict flows equilibrating inside the module. At the end of the day to keep current from consuming our module, this



module needs to LVDS yet we utilize simply the LED lone to show if the module is working appropriately or not. Presently, I will show how these parts are associated together. There we have the actual sensor and obviously, sixers these Thurs and two LEDs. On the correct, you can see all the pins of the module are associated on this board. We have in dark, the progression of the ground, we have web, the progression of the voltage. We have a light green, the progression of the simple sign and obviously, we have in green, the progression of the advanced signs thermistors are temperature subordinate resistors changing opposition with temperature changes, they are exceptionally delicate and respond to little changes in temperature. They are best utilized when a particular temperature should be kept up and when checking temperatures inside 50 degrees Celsius of encompassing or thermistor accomplishes high accuracy inside a restricted temperature scope of around 50 gates air just around the objective temperature, however, this reach is subject to the base obstruction thermistors as a feature of temperature controls them is the most ideal approach to quantify and control warming and cooling. Their capacity to change in moment increases permitted the extraordinary generally framework dependability. thermistors can be inserted in the surface mounted on the gadget requiring temperature observing. Contingent upon the

type, they can quantify fluids, gases, or solids. We should discuss the pins. The pins in this module are four, we have the ground pin with G sign, we have the voltage pin with the in addition to sign. We have the simple pin with a 0 sign. Also, we have the computerized pin with the 0 sign. So we should discuss the signal. This module gives two signs, we can associate the pin of the simple sign with any simple port and the pin of the computerized signal with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

KY-028 Specifications

Operating Voltage	3.3V to 5.5V
Temperature measurement range	-55°C to 125°C [-67°F to 257°F]
Measurement Accuracy	±0.5°C
Board Dimensions	15mm x 36mm [0.6in x 1.4in]

KY-028 Connection Diagram

KY-028	Arduino
A0	A0
G	GND
+	5V
D0	2

## CONNECTIONS ARDUINO:

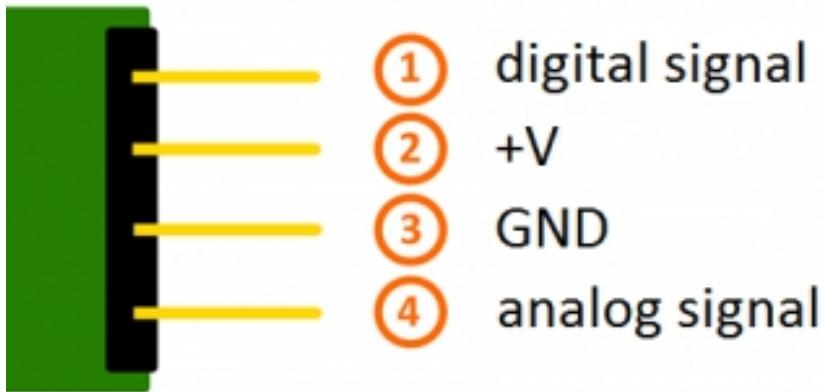
digital signal = [Pin 3]

+V = [Pin 5V]

GND = [Pin GND]

analogue Signal = [Pin 0]

## INPUT:



## EXAMPLE SENSOR CODE:

```
int led = 13; // define the LED pin
int digitalPin = 2; // KY-028 digital interface
int analoguePin = A0; // KY-028 analogue interface
int digitalVal; // digital readings
int analogueVal; //analogue readings
void setup()
{
pinMode(led, OUTPUT);
pinMode(digitalPin, INPUT);
//pinMode(analoguePin, OUTPUT);
Serial.begin(9600);
}
void loop()
{
// Read the digital interface
```

```
digitalVal = digitalRead(digitalPin);
if(digitalVal == HIGH) // if temperature threshold reached
{
    digitalWrite(led, HIGH); // turn ON Arduino's LED
}
else
{
    digitalWrite(led, LOW); // turn OFF Arduino's LED
}

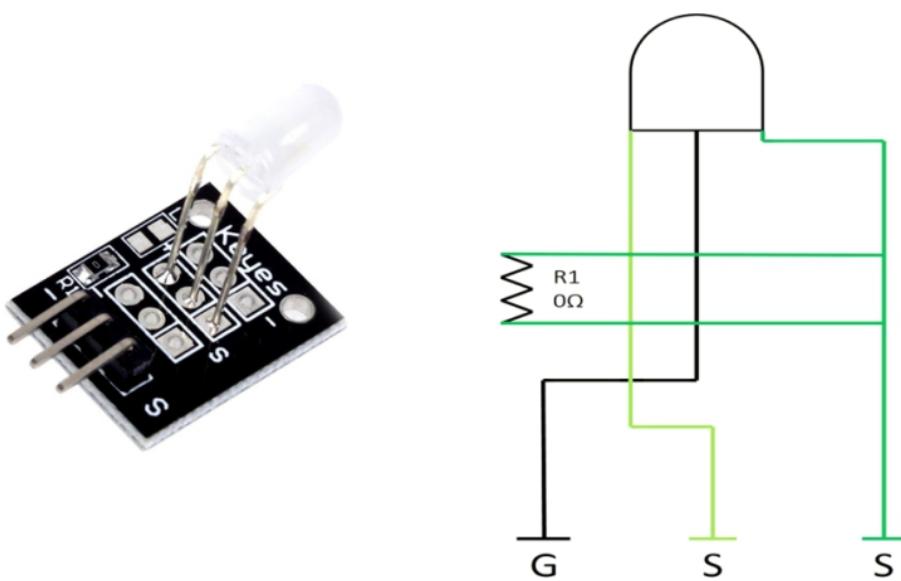
// Read the analogue interface
analogueVal = analogueRead(analoguePin);
Serial.println(analogueVal); // print analogue value to serial

delay(100);
```

## KY-029 YIN YI 2-COLOUR LED MODULE 3MM



The two shadings drove model known as KY0 29. This is a module itself. So we should begin. This module has incorporated onboard one LED and one resistor, the resistor utilized in this module. It's0 arm and I don't know the motivation behind why we utilize a 0-ohm resistor in this model. Presently, I will So every one of these parts are associated together. There we have the actual LED. Also, the resistor r1. On the correct you can perceive how the piece of the module are associated on this board, we have in dark, the progression of the ground, we have in green and in light green, the progression of the signals of the LED in this module can create light in two tones in red, and in green. With this module, we can illuminate each tone separately, or we can make a blended shading utilizing the mix of the red tone and the green tone.



We should discuss the pins. The pins in this module are three. We have the ground pin with a short sign. The voltage pin is in the centre And obviously, we have the pin of the sign with the S sign. So we should discuss the sign this module takes to computerized signal. So we can interface the pins of the sign with any computerized port of various microcontroller sheets like Arduino or Raspberry Pi.

#### KY-029 Specifications

Operating Voltage	2.3-2.6V for green, 1.9-2.2V for red
Working Current	20mA
Diameter	5mm
Package Type	Diffusion
Color	Red + Green
Wavelength	571nm + 625nm
Luminous Intensity	20~40mcd, 60~80mcd

#### KY-029 Connection Diagram

KY-029	Breadboard	Arduino
-		GND
middle	330Ω resistor	Pin 11
S	330Ω resistor	Pin 10

## CONNECTIONS ARDUINO:

LED Green = [Pin 10]

LED Red = [Pin 11]

Sensor GND = [Pin GND]

## INPUT:

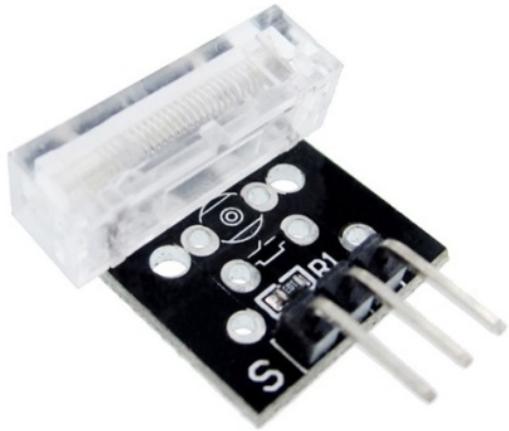


## EXAMPLE SENSOR CODE:

```
intredpin = 11; // pin for the red LED  
intgreenpin = 10;// pin for the green LED  
intval;  
void setup()  
{  
pinMode(redpin, OUTPUT);  
pinMode(greenpin, OUTPUT);  
Serial.begin(9600);  
}
```

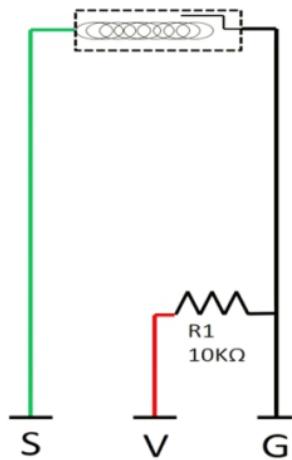
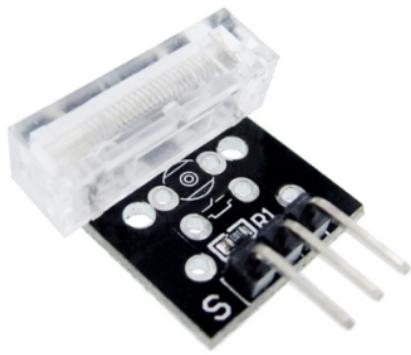
```
void loop()
{
    for(val = 255; val> 0; val--)
    {
        analogueWrite(redpin, val);
        analogueWrite(greenpin, 255 - val);
        delay(10);
    }
    Serial.println("Green");
    delay(1000);
    for(val = 0; val< 255; val++)
    {
        analogueWrite(redpin, val);
        analogueWrite(greenpin, 255 - val);
        delay(10);
    }
    Serial.println("Red");
    delay(1000);
```

## KY-031 KNOCK SENSOR MODULE



The thump sensor module is known as KY0 31. This is a module itself. So how about we begin. This module is coordinated onboard one thump sensor and one resistor, the resistor utilizing this module. It's Thank you, Alma. The fundamental purpose behind utilizing the resistor is to restrict money associating with the module. At the end of the day, to keep current from consuming our model. Presently I will show how these parts are associated together. There we have the actual sensor and obviously

The resistor r1. On the correct, you can perceive how the pins of the module are associated on this board. We have intruded on the progression of the voltage, we have in dark, the progression of the ground, and obviously, we have in green, the progression of the sign.



This module is utilized to identify NOx when a thump is distinguished, which implies that the spring that is inside the module is associated with the two contacts of the sensor that are on the contrary side of the sensor. The switch is regularly off in the resting state, however, it will be on when an outside power will stun the sensor and by the outcome, the spring will close the circuit. The aftereffect of development inside the thump sensor will bring about changes in the electric property, after giving the data that the sensor has recognized the thump, it will quickly reset in this typical state to sit tight for different thumps.

We should discuss the pins. The pins in this module are three, we have the ground pin with a short sign, the voltage pin is in the centre. What's more, obviously we have the pin of the sign with the S sign. So we should discuss the sign. This module gives an advanced sign. So we can interface the pins of the sign with any computerized ports of various microcontroller sheets like Arduino or Raspberry Pi.

### KY-031 Specifications

Operating Voltage	3.3V to 5V
Output Type	Digital

### KY-031 Connection Diagram

KY-031	Arduino
S	Pin 3
middle	+5V
-	GND

## CONNECTIONS ARDUINO:

LED + = [Pin 13]

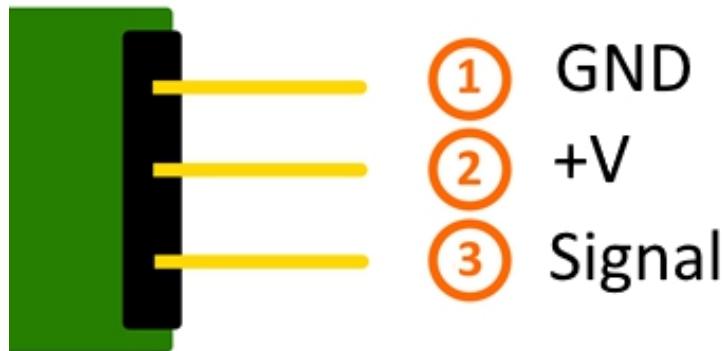
LED - = [Pin GND]

Sensor signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor - = [Pin GND]

## INPUT:



## EXAMPLE SENSOR CODE:

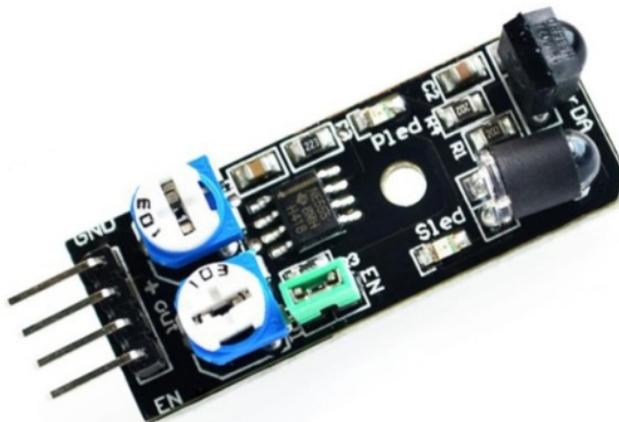
```

int Led = 13;      // LED on Arduino board
int Shock = 3;     // sensor signal
intval;           // numeric variable to store sensor status
void setup()
{
    pinMode(Led, OUTPUT); // define LED as output interface
    pinMode(Shock, INPUT); // define input for sensor signal
}
void loop()
{
    val = digitalRead(Shock); // read and assign the value of
    digital interface 3 to val
    if(val == HIGH) // when sensor detects a signal, the LED
    flashes
    {
        digitalWrite(Led, LOW);
    }
}

```

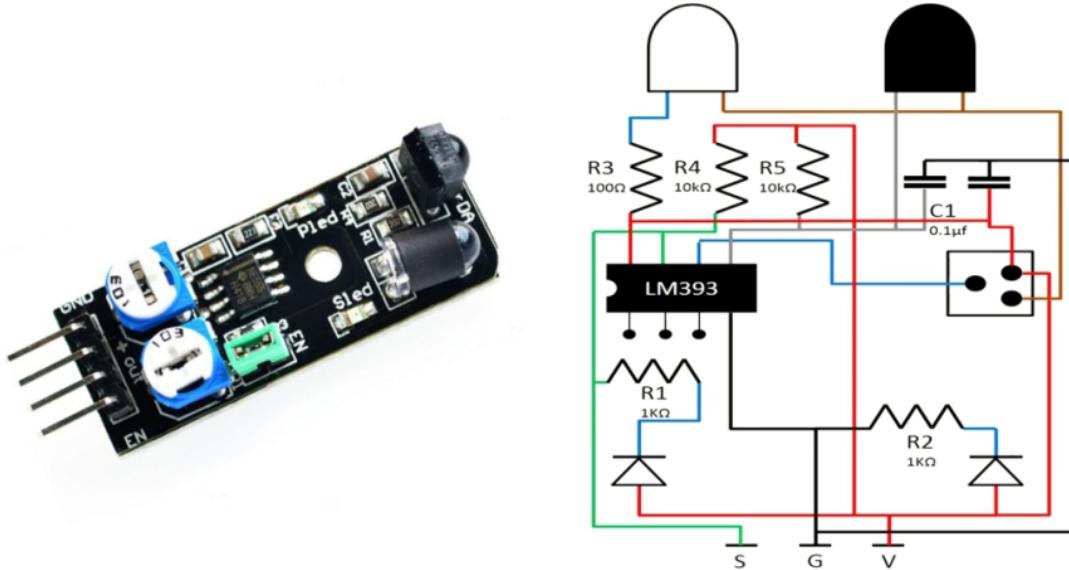
```
    }  
    else  
    {  
        digitalWrite(Led, HIGH);
```

## KY-032 OBSTACLE AVOIDANCE SENSOR MODULE



The deterrent shirking sensor is known as KY0 32. This is simply the module. So we should begin. This module is incorporated on board two sensors, one infrared transmitter and one infrared collector. additionally incorporated onboard one microprocessor, one potentiometer, five resistors, two capacitors and two LVDS. The resistor r1 utilized in this module is a one-kilo ohm resistor R two is the one-kilo ohm. There is this there are three it's 100 ohms. There is this there are four it's Thank you ohms and there is these there are five it's 10-kilo ohms. The principle purpose behind utilizing the resistor is to restrict cash ordering inside the module. At the end of the day, to keep current from consuming our module. We likewise have two capacitors, the capacitor swan with a limit of 0.2 on miniature farad and capacitor C two with a limit of 0.1 miniature farad

which is fundamentally used to store energy verla dl one is utilizing module to show is the module is working appropriately or not. Also, the LED to show if the module has distinguished any obstruction. So the LED will illuminate each time that any deterrent has been identified by the sensors. Presently I will show



how these parts are associated together. There we have the actual sensors, and, five resistors, two capacitors and two LEDs. On the correct, you can perceive how the piece of the module symbol nated. On this board, we have in dark, the progression of the ground, we have delta the progression of the voltage and obviously, we have in green, the progression of the sign. This module's used to identify the obstructions that are before the module. The greatest distance that the sensor can identify it's up to 40 centimetres. The sensor says Or communicates a 38 kilohertz infrared light that can go up to 40 centimetres. Furthermore, we can change it by utilizing the potentiometer. After transmission, the infrared light will disperse in the air. In any case, on the off chance that they hit something that is between 0centimeters to the greatest distance that we need, a portion of the red light will be consumed by the actual item and the rest will be back, which at that point will be caught by the Empire the beneficiary. In an ordinary state when the infrared recipient has not

caught any infrared light discharged by the transmitter, the sign that the collector will give it will be high, yet in the wake of recognizing the infrared light, the infrared beneficiary will give a sign that will below and in a similar time, the LED I will illuminate showing us that is an article before the sun. The module is excellent at identifying the deterrent. What's more, the sign that reveals to us that the item has been distinguished comes immediately. We should discuss the pins. The pins in this module are three, we have the ground pin with a GND sign, the voltage pin with a Vcc sign, and the pin of the sign with an outside.lets talk about the sign. This module gives a computerized signal. So we can interface the pins of the sign with any advanced ports of various microcontroller sheets likeArduino or Raspberry Pi.

KY-032 Specifications

Working voltage	3.3V ~ 5V DC
Working current	≥ 20mA
Working temperature	-10°C ~ 50°C [14°F ~ 122°F]
Detection distance	2cm ~ 40cm [0.79in ~ 15.75in]
IO interface	4-wire interface (-/+/S/EN)
Output signal	TTL level (low level if obstacle detecter, high if no obstacle)
Adjustment method	multi-turn resistance adjustment
IR pulse frequency	38kHz according to HS0038DB datasheet
Effective angle	35°
Board Size	1.6cm x 4cm [0.62in x 1.57in]
Weight	9g

KY-032 Connection Diagram

KY-032	Arduino
GND	GND
+	5V
out	Pin 3
EN[+]	Pin 2[+]

## CONNECTIONS ARDUINO:

Sensor enable = [N.C. (jumper plugged in)]

Sensor signal = [Pin 10]

Sensor +V = [Pin 5V]

Sensor GND = [Pin GND]

## EXAMPLE SENSOR CODE:

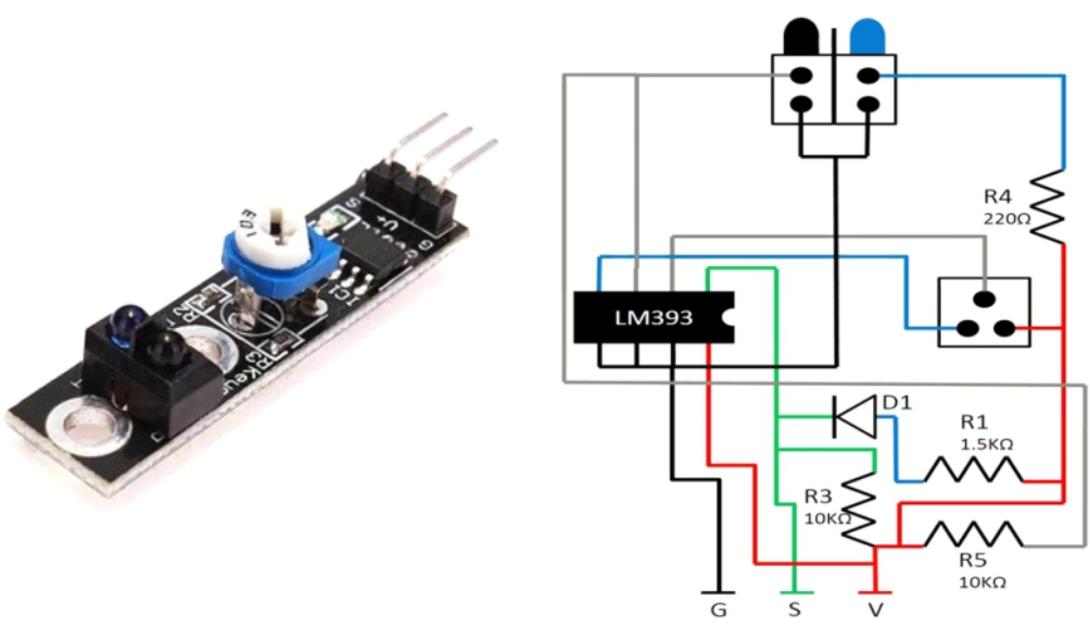
```
intledPin = 13; // LED pin on arduino
```

```
intdetectorPin = 3; // obstacle avoidance sensor interface
intval;      // variable to store result
//intenablePin = 2; // sensor enable interface (EN)
void setup()
{
pinMode(ledPin, OUTPUT); // Define LED as output interface
pinMode(detectorPin, INPUT); // Define obstacle avoidance sensor
as input interface
// [uncomment and remove jumper on module to use enable pin
//(EN)]
//pinMode(enablePin, OUTPUT);
//digitalWrite(enablePin, HIGH); // Enable sensor
}
void loop()
{
val = digitalRead(detectorPin); // Read value from sensor
if(val == LOW) // When the sensor detects an obstacle, the LED on
the Arduino lights up
{
digitalWrite(ledPin, HIGH);
}
else
{
digitalWrite(ledPin, LOW);
```

## KY-033 HUNT SENSOR MODULE



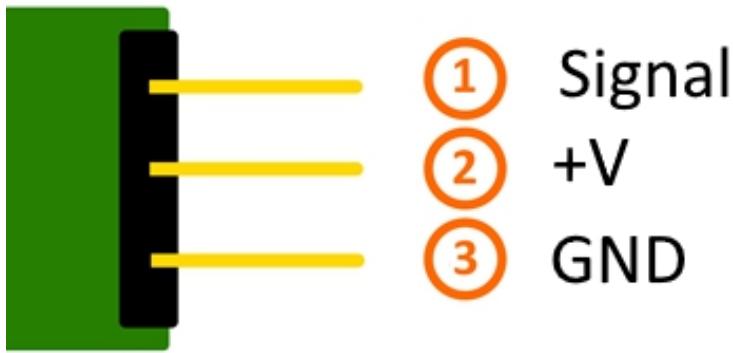
The line following sensor known as KY0 33. This is a module itself. So how about we begin. This module has coordinated ready, adjust tracker, one potentiometer. One central processor for resistors and one LED. There is there are one utilizing this module, it's 1.5-kilo-ohms resistor, r three is everything that matters these two are four, it's 220 ohms. is these are our five. It's Thank you Oh, and the principle explanation behind utilizing the resistors is to restrict current SQL dating inside the module. As such, to keep current from consuming our module. The LED lone is utilizing this module to show us if the module is in the path or not. Presently I will show how these parts are associated together. There we have the actual sensor. Furthermore, timberland Easter's as the LED deewan. On the correct you can see have the piece of the module symbol nation this board, we have in dark, the progression of the ground, we have in yet the progression of the voltage. What's more, obviously we have in green, the progression of the sign.



This module is utilized to identify if the module is inline or not. Furthermore, it does that by identifying the surface that the sensor is confronting. As should be obvious, when I direct the module outside of the path, the LED goes on to disclose to us that the sensor isn't in the path. Also, it does that by distinguishing the actual surface. All the more exactly, it distinguishes if the surface you feel like the light or if the surface ingests Light. The motivation behind why we utilize a dark tone for the path is that the dark tone assimilates the light that the transmitters shipped off the surface.

We should discuss the piece. The piece in this module are three, we have the ground pin with a GND sign, the voltage pin with a Vcc sign, and the pin of the sign with an outside. So we should discuss the sign. This module gives an advanced sign. So we can interface the pins of the sign with any advanced ports of various microcontroller sheets like Arduino or Raspberry Pi.

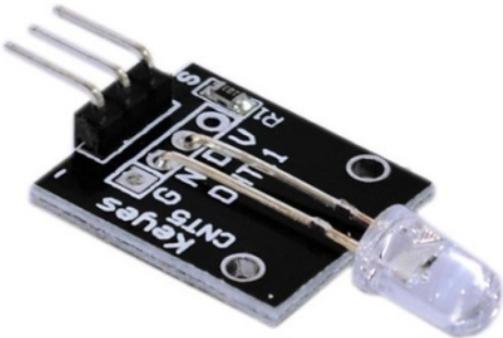
## INPUT:



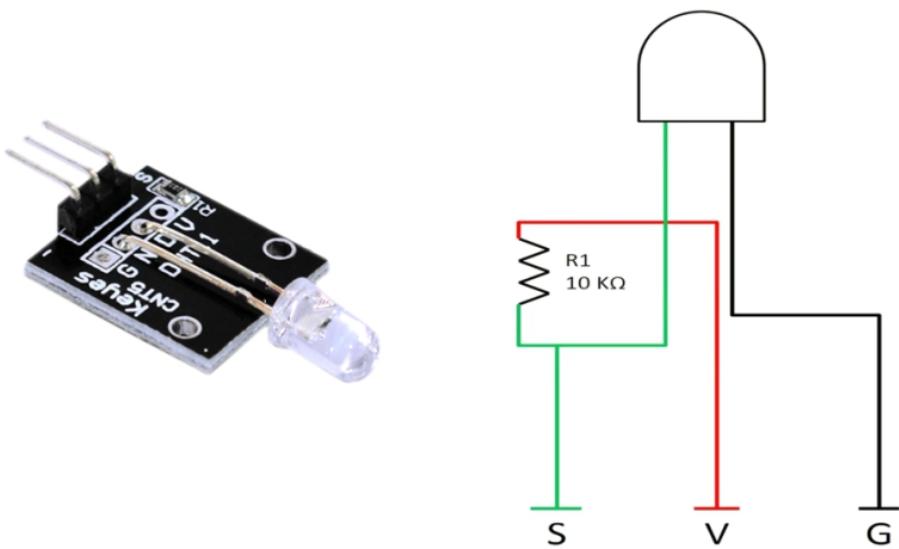
## EXAMPLE SENSOR CODE:

```
int Sensor = 3; // sensor input pin  
  
void setup ()  
{  
    Serial.begin(9600); // Initialize serial output  
    pinMode (Sensor, INPUT) ;  
}  
  
void loop ()  
{  
    boolval = digitalRead (Sensor) ; // The current signal of the sensor  
    will be read  
    if (val == HIGH)  
    {  
        Serial.println("Line detected");
```

**KY-0347 COLOUR LED FLASH-  
MODULE**



The programmed blazing shading drove model known as KY 0 34. This is a module itself. So how about we begin. The present this month module is incorporated onboard one programmed shading LED and one resistor the resistor utilized in this module it's Thank you an arm and the fundamental explanation behind utilizing the resistor is to restrict cash examining inside the module. At the end of the day, to keep current from consuming our module.



Presently, I will show how these parts are associated together. There we have the programmed shading LED, and obviously, there is

Easter r1. On the correct, you can perceive how the piece of the module symbol nated. On this board, we have in dark, the progression of the ground, we have web the progression of the voltage and obviously, we have in green, the progression of the sign. Outwardly D can create light in seven distinct tones and can change starting with one shading then onto the next naturally. This module is modified since Creation, and we can't change the time, the cycles or the request that the tones have appeared. There's nothing left but to associate the module with any microcontroller board and utilize the board like a change to kill the LED on or. We should discuss the pins. The pins in this module are three, we have the ground pin with the short sign, the voltage pin is in the centre. Furthermore, we have the pin of the sign with the S sign. So we should discuss the sign. This module takes an advanced sign, so we can associate the pin of the sign with any computerized part of various microcontroller sheets like Arduino or Raspberry Pi

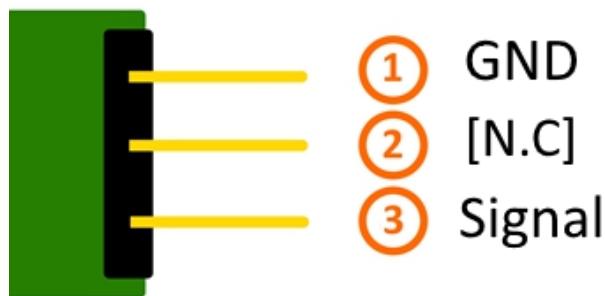
## CONNECTIONS ARDUINO:

Sensor Signal = [Pin 13]

Sensor [N.C] =

Sensor GND = [Pin GND]

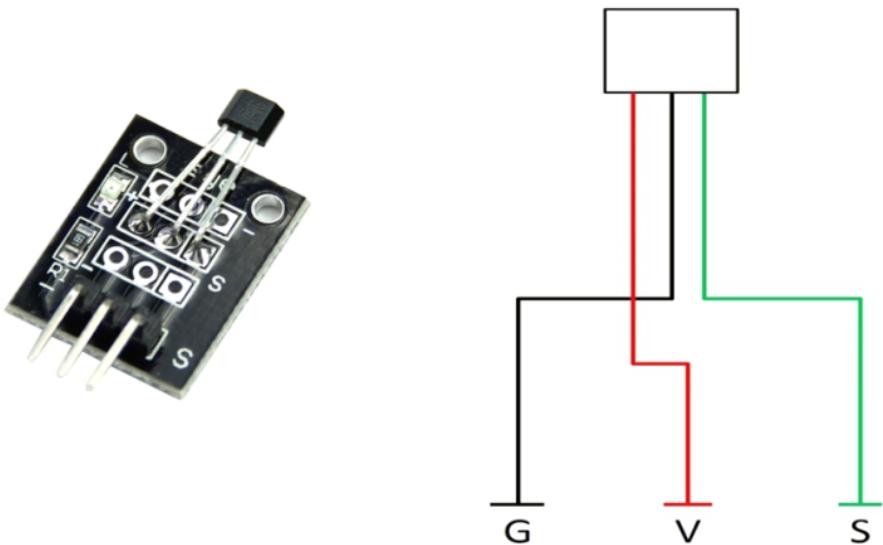
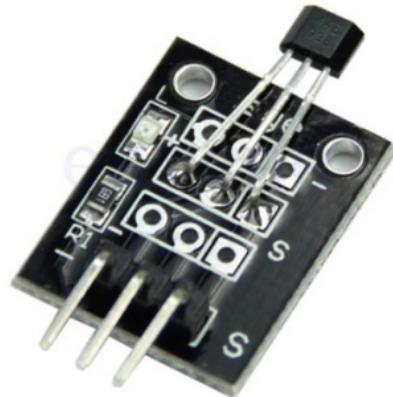
## INPUT:



## EXAMPLE SENSOR CODE:

```
int Led = 13;  
void setup ()  
{  
    pinMode (Led, OUTPUT); // Initialization of the LED output pin  
}  
  
void loop () // main program loop  
{  
    digitalWrite (Led, HIGH); // LED will be switched on  
    delay (4000); // waitmode for 4 seconds  
    digitalWrite (Led, LOW); // LED will be switched off  
    delay (2000); // waitmode for another 2 seconds  
}
```

## KY-035 BIHOR MAGNETIC SENSOR MODULE

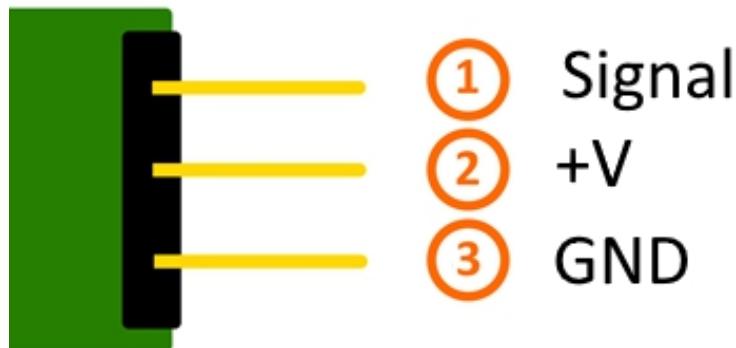


The B home attractive sensor known as KY0 35. This is a module itself. So how about we begin. As should be obvious, this module has just the Bihar attractive sensor and that's it. I will show how these segments are associated together. There we have the sensor itself on assortment you can perceive how the piece of the module is associated on this board. Along these lines, we have in red, the progression of the voltage we have in dark, the progression of the

ground and we have in green the progression of the signal. All that the sensor does is to the information attractive field says this module has just the attractive sensor ready and know items that help outwardly to check whether the attractive field has been identified, we can program by our self. Another thing that we can do to check whether any attractive field has been identified by the sensor is to open our PC. Or on the other hand, we can interface any LCD to our board.

We should discuss the pins. The pins in this module are three, we have the ground pin with a short sign, the voltage pin is in the centre. Also, we have the pin of the sign with the S sign. So we should discuss the sign. This module gives a simple sign so we can associate the pin of the sign with any simple port of various microcontroller sheets like Arduino You know or Raspberry Pi.

## INPUT:



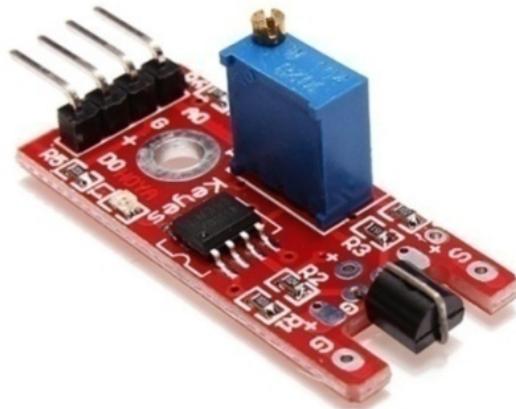
## EXAMPLE SENSOR CODE:

```
intsensorPin = A5; // Declaration of the input pin  
// Serial OUT in 9600 baud  
void setup()  
{
```

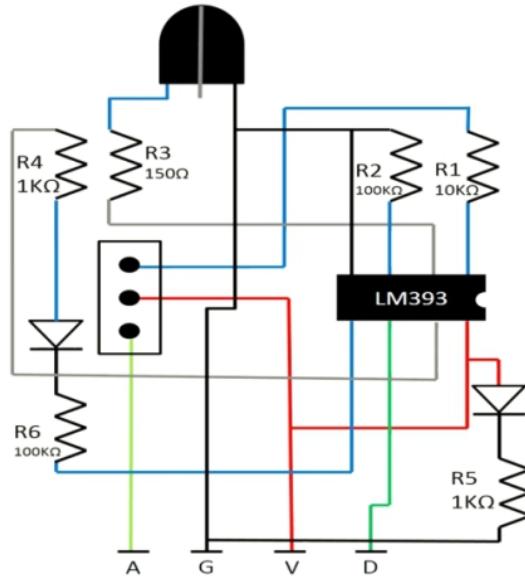
```
Serial.begin(9600);
}

// The program measures the current voltage at the sensor,
// calculates the resistance with it and a known resistor
// and outputs it via serial OUT
void loop()
{
    // Measuring of the current voltage...
    intrawValue = analogueRead(sensorPin);
    float voltage = rawValue * (5.0/1023) * 1000;
    floatresitance = 10000 * ( voltage / ( 5000.0 - voltage ) );
    // ... output via serial interface
    Serial.print("Voltage:"); Serial.print(voltage); Serial.print("mV");
    Serial.print(", Resistance:"); Serial.print(resistance);
    Serial.println("Ohm");
    Serial.println("-----");
    delay(500);
}
```

## KY-036 METAL-TOUCH SENSOR MODULE



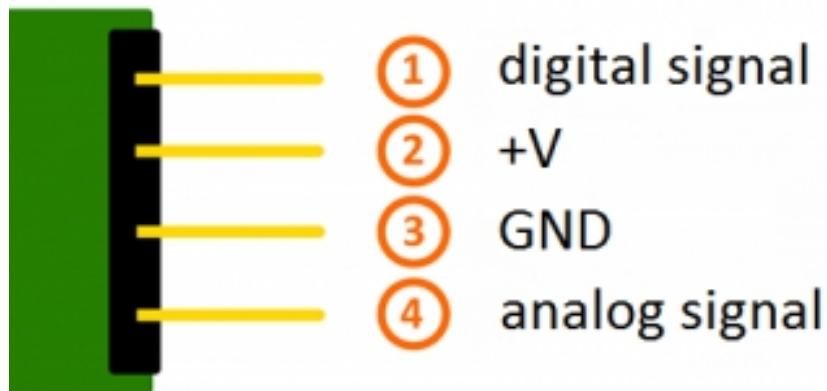
The touch sensor module is known as KY 0 36. This is simply the module. So we should begin. This module is coordinated on board the touch sensor, one potentiometer, one microprocessor, six resistors and two LEDs. The resistor r1 utilized in this module, it's 10-kilo ohms. That resistor R two is 100-kilo ohms. The resistor R three is 150 ohms. There is Easter are for its one kilo all the resistors are five, it's wonKY on their resistor RC is 100-kilo ohms. Furthermore, the primary purpose behind utilizing the resistors is to restrict the current flowing inside the module. As such, to keep erosion from consuming our module The LED lone lights up to show if the module is working appropriately or not. Also, the LED I two lights up each time that we contact the sensor.



Presently I will show how these parts are associated together. There we have the actual sensor and obviously, six resistors and two LEDs. On the correct, you can perceive how the pins of the module are associated on this board. We have in dark, the progression of the ground, we have connected, the progression of the voltage, we have in light green, the progression of the simple sign. What's more, obviously we have in green, the progression of the computerized signals. This module utilizes our NPN semiconductor, however the base of the semiconductor it's not associated with the board. This module is To distinguish if the sensor is contacted or not. In typical express, the LED I two will be off. As an advanced sign that This module will give, it will below. Be that as it may, the LED I two will turn on when we contact the sensor. And afterwards, the computerized signal that This module will give will be high, we can utilize the potentiometer to change the affectability of the sensor. We should discuss the pins. The pins in this module are four. We have the ground pin with the G sign, we have the voltage pin with the in addition to sign. We have the simple pin with a 0 sign. Also, we have the computerized pin with the 0 sign. So we should discuss the sign this module gives two signs, we can interface the pin of the simple sign with any simple port and the pin of the computerized signal with

any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

## INPUT:

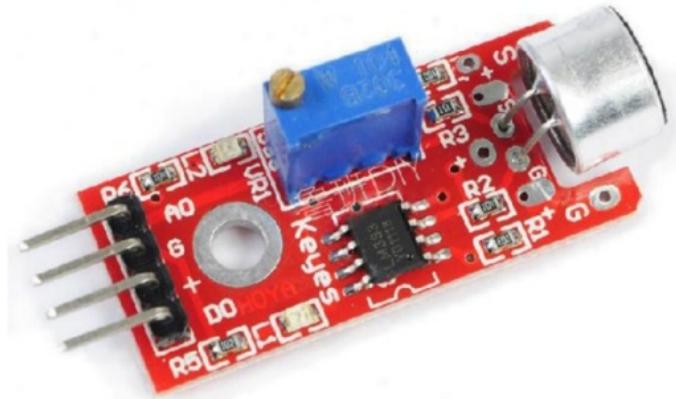


## EXAMPLE SENSOR CODE:

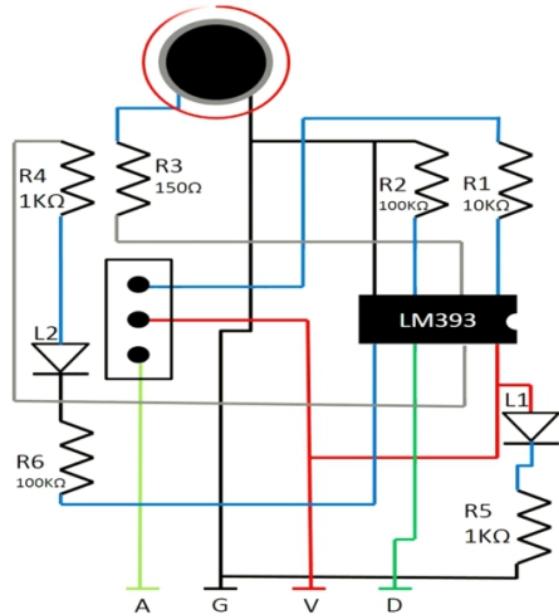
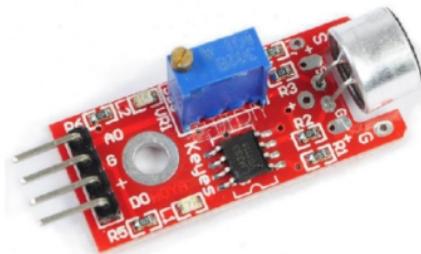
```
/ Declaration and initialization of the input pin
intAnalogue_Eingang = A0; // X-axis-signal
intDigital_Eingang = 3; // Button
void setup ()
{
pinMode (Analogue_Eingang, INPUT);
pinMode (Digital_Eingang, INPUT);
Serial.begin (9600); // Serial output with 9600 bps
}
// The program reads the current value of the input pins
// and outputs it via serial out
void loop ()
{
```

```
floatAnalogue;  
int Digital;  
// Current value will be read and converted to the voltage  
Analogue = analogueRead (Analogue_Eingang) * (5.0 / 1023.0);  
Digital = digitalRead (Digital_Eingang);  
// and outputted here  
Serial.print ("Analogue voltage value:"); Serial.print (Analogue, 4);  
Serial.print ("V, ");  
Serial.print ("Extreme value:");  
if(Digital==1)  
{  
    Serial.println (" reached");  
}  
else  
{  
    Serial.println (" not reached yet");  
}  
Serial.println ("-----");  
delay (200);  
}
```

## **KY-037 MICROPHONE SENSOR MODULE (HIGH SENSITIVITY)**



The sound identification mouthpiece module is known as KY 0 37. This is simply the module. So we should begin. This module is coordinated onboard one receiver one potentiometer on computer chip sixers diesters m two drove the resistor r1. Utilizing this module it's on there as these are our two it's 100 kilo-ohms. There is this there are three it's 150 on there is thusly it's the one-kilo ohm. There's these there are five it's one



kilo and there is Easter it is 100-kilo ohmsThe primary purpose behind utilizing the resistors is to restrict money-gathering inside the module. As such, to keep current from consuming our module the LED will illuminate to show if the module is working appropriately or not.

Presently, I will show how these segments are associated together. There we have the actual sensor and obviously, sixers these Thurs and two LEDs abrogate to cover the piece of the module are associated on this board. We have in dark, the progression of the ground, we have communicated the progression of the voltage we have in light green, the progression of the simple sign and we have in green the progression of the computerized signals. This module is utilized to distinguish clamour applauds, warms etc. It permits us to identify one sound that has surpassed a set point we have picked with the goal that the sensor can recognize it. Sound is identified through a mouthpiece and feed into the actual CPU. The recognition of the sound level is changed utilizing a potentiometer that is on the board.

In ordinary express, the computerized yield that This module will give will be high. However, when the sensor distinguishes something, the computerized yield that the module will give will below. Let's a discussion about the pins. The pins in this module are four. We have the ground pin with the G sign. We have the voltage pin with the in addition to sign. We have the simple pin with a 0 sine and obviously, we have the computerized pin with the 0sign. So we should discuss the sign. This module gives two signs, we can interface the pin of the simple sign with any simple port, and the pin of the computerized signal with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

## **CONNECTIONS ARDUINO:**

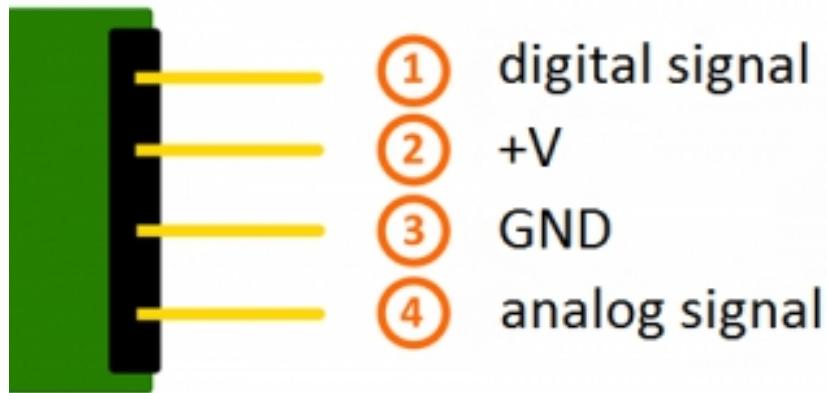
digital signal = [Pin 3]

+V = [Pin 5V]

GND = [Pin GND]

analogue signal = [Pin 0]

## INPUT:



## EXAMPLE SENSOR CODE:

```
// Declaration and initialization of the input pin
intAnalogue_Eingang = A0; // X-axis-signal
intDigital_Eingang = 3; // Button

void setup ()
{
    pinMode (Analogue_Eingang, INPUT);
    pinMode (Digital_Eingang, INPUT);

    Serial.begin (9600); // Serial output with 9600 bps
}

// The program reads the current value of the input pins
// and outputs it via serial out
```

```
void loop ()  
{  
floatAnalogue;  
int Digital;  
  
// Current value will be read and converted to voltage  
Analogue = analogueRead (Analogue_Eingang) * (5.0 / 1023.0);  
Digital = digitalRead (Digital_Eingang);  
  
//... and outputted here  
Serial.print ("Analogue voltage value: "); Serial.print (Analogue, 4);  
Serial.print ("V, ");  
Serial.print ("Extreme value: ");  
  
if(Digital==1)  
{  
Serial.println (" reached");  
}  
else  
{  
Serial.println (" not reached yet");  
}  
Serial.println ("-----");  
delay (200);  
}
```

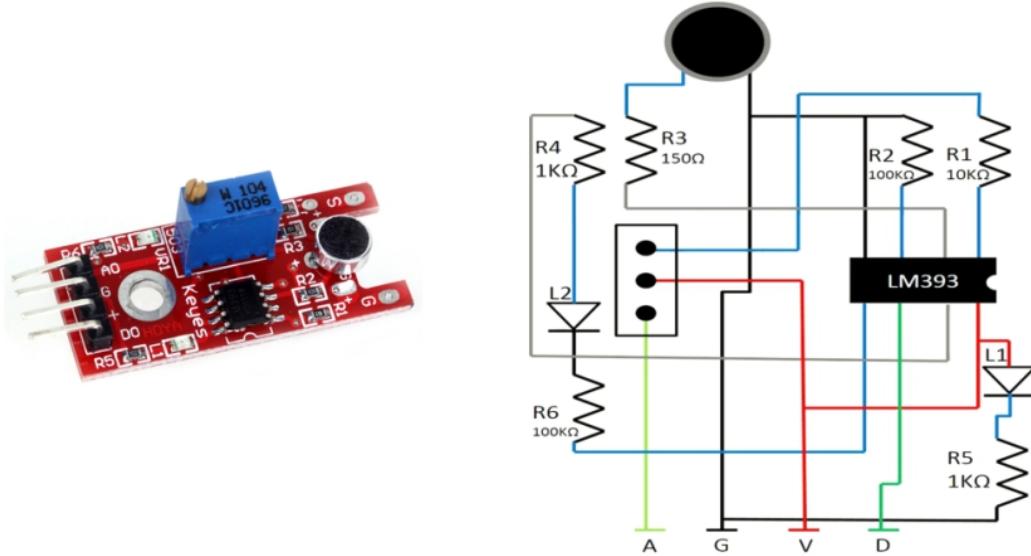
# KY-038 MICROPHONE SOUND SENSOR MODULE



The sound identification mouthpiece module is known as KY0 38. This is a module itself. So we should begin. This module is coordinated onboard one receiver one potentiometer on microprocessor sixers. diesters m two LED is the resistor r1 utilized in this module. It's Thank you arms. There is the SIR r two it's 100-kilo ohms there is the sir are three it's 150 ohms There will be there are four it's one kilo there is this there are five it's one kilo and there is this there it is 100-kilo ohms. The principle explanation behind utilizing the resistors is to restrict flows equivalent driving inside the module. All in all, to keep current from consuming our module, the LED I on light up to show if the module is working appropriately or not.

Presently, I will show how these segments are associated together. There we have the actual sensor. What's more, Sikhs are z SIRs and LDS supersede to disguise the piece of the module are associated on this board. We have in dark, the progression of the ground, we have connected, the progression of the voltage we have in light

green, the progression of the simple sign and We have in green, the progression of the advanced signs.



This module is utilized to identify commotion, applauds, hits, and so on. It permits us to identify one sound that has surpassed a set point we have picked so the sensor can distinguish it. Sound is the title is utilizing an amplifier and feed into the actual CPU. An identification of the sound level is changed through the potentiometer that is on the board. In typical express, the advanced yield that this module we'll give will be high. In any case, when the sensor distinguishes something, the advanced yield that the module will give will below. We should discuss the pins. The pins in this module are four. We have the ground pin with the G sign. We have the voltage pin With the in addition to the sign, we have the simple pin with a 0 sign. Furthermore, we have the computerized pin with the 0 sign. So we should discuss the sign. This module gives two signs, we can interface the agony of the simple sign with any simple port and the pin of the advanced sign with any computerized port of various microcontroller sheets like Arduino or Raspberry Pi.

## **CONNECTIONS ARDUINO:**

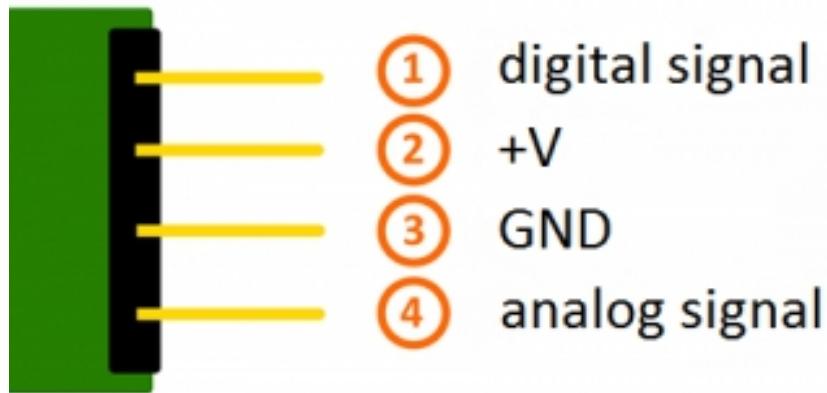
digital Signal = [Pin 3]

+V = [Pin 5V]

GND = [Pin GND]

analogue Signal = [Pin 0]

## **INPUT:**



## **EXAMPLE SENSOR CODE:**

```
// Declaration and initialization of the input pin  
intAnalogue_Eingang = A0; // X-axis-signal  
intDigital_Eingang = 3; // Button  
void setup ()  
{  
    pinMode (Analogue_Eingang, INPUT);  
    pinMode (Digital_Eingang, INPUT);  
  
    Serial.begin (9600); // Serial output with 9600 bps
```

```
}

// The program reads the current value of the input pins
// and outputs it via serial out

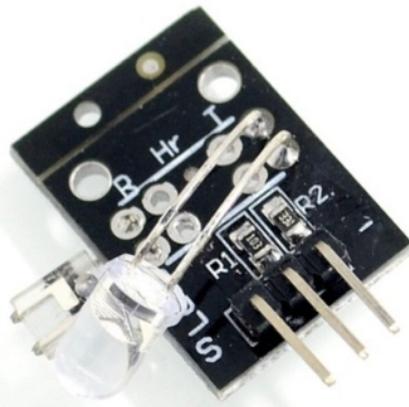
void loop ()
{
floatAnalogue;
int Digital;

// Current value will be read and converted to voltage
Analogue = analogueRead (Analogue_Eingang) * (5.0 / 1023.0);
Digital = digitalRead (Digital_Eingang);

//... and outputted here
Serial.print ("Analogue voltage value:"); Serial.print (Analogue, 4);
Serial.print ("V, ");

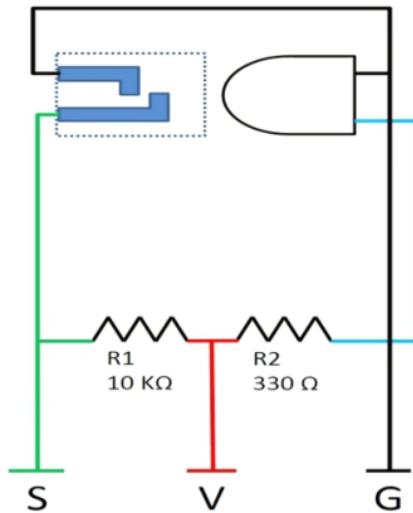
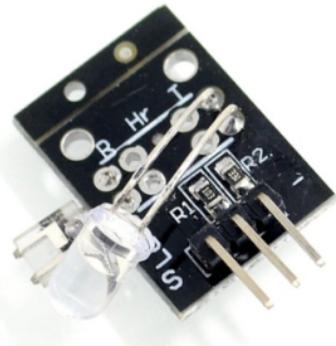
Serial.print ("Extreme value:");
if(Digital==1)
{
Serial.println (" reached");
}
else
{
Serial.println (" not reached yet");
}
Serial.println ("-----");
delay (200);
}
```

# KY-039 HEARTBEAT SENSOR MODULE



The heartbeat sensor is known as KY 0 39 This is a module itself. So we should begin This module is incorporated onboard one infrared LED one photograph diet and vacationers Easter so resistor r1 and the resistor R two resistor r1 Newseum This module is thank you And then there's these there are two 330 ohms. The fundamental purpose behind utilizing the resistors is to restrict flows circling inside the model. Presently the word to keep current from consuming our model. Presently we'll show how these parts are associated together. There we have the actual sensors. Furthermore, there is these two Erwan and there is these are two. On the right, you can perceive how the piece of the module I composed on this board, we have in dark, the progression of the ground, we have bay the progression of the voltage and obviously, we have in green, the progression of the sign.

This module is utilized to recognize the pulse. Frankly, this module isn't precise. The infrared LED will radiate the infrared light, which the greater part of them are consumed by the blood and some of them will go through



The infrared lights that will go through will be caught by the photograph diet, which at that point will give a simple sign dependent on the measure of the infrared light catch. blood in the arteries assimilates light better compared to the encompassing body tissues so as our various direct and swell because of the throbbing pulse, the power of the mirrored light ascents and falls. We should discuss the pins. The pins in this module are three we have the ground pin with a less significant, the voltage pin is in the centre. What's more, obviously we have the pin of the sign with the S sign. So we should discuss the sign. This module gives a simple sign so we can interface the agony of the sign with any simple port of various microcontroller sheets like Arduino or Raspberry Pi.

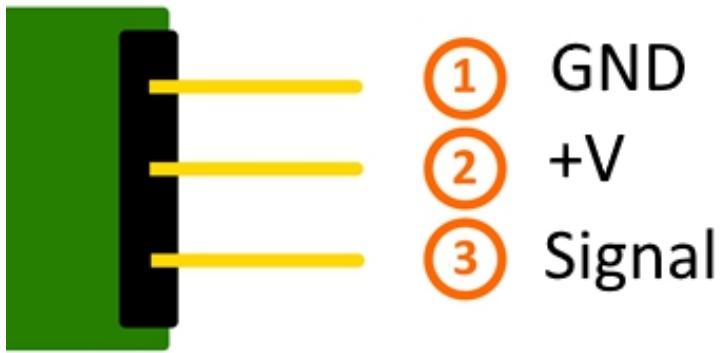
## **CONNECTIONS ARDUINO:**

Sensor Signal = [Pin 0]

Sensor +V = [5V]

Sensor - = [Pin GND]

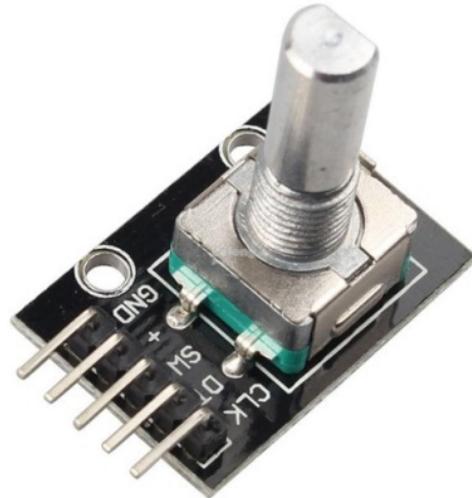
## **INPUT:**



## EXAMPLE SENSOR CODE:

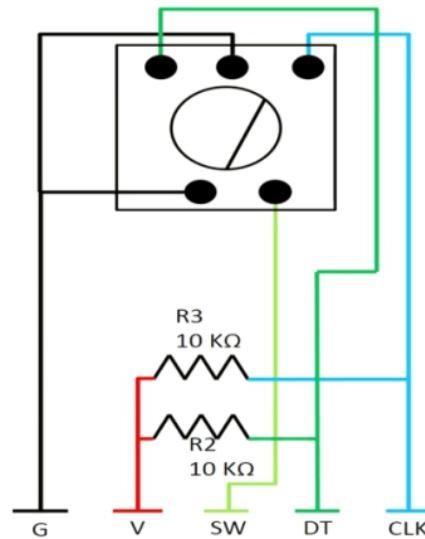
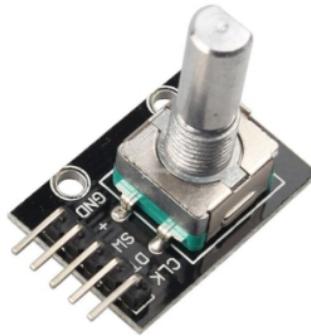
```
n = 0;  
start = millis();  
reader = 0.;  
do  
{  
    reader += analogueRead (sensorPin); // read and add values...  
    n++;  
    now = millis();  
}  
while (now < start + 20); // ...until 20 ms have elapsed  
reader /= n; // and take an average of the values
```

## KY-040 ROTARY ENCODER



The rotating encoder module is known as KY040. This is simply the module. So we should begin. This module is coordinated onboard one rotating encoder and traveller sisters resistor R two and resistor R three which are the two of them thank you look on the principle purpose behind utilizing the resistors is to restrict current flowing inside the module in another word to keep current from consuming our mode.

Presently I will show how these segments are associated together. There we have the rotating encoder itself and obviously, there is the SIR r two and there's these are our three. Albeit right you can perceive how the piece of the module connected on this load up, we have in dark, the progression of the ground, we have web, the progression of the voltage, we have in light green, the progression of the sign from the catch, we have in green, the progression of the sign from the primary switch, and we have in blue, the advanced sign from the second switch. The revolving encoder on this module can be turned clockwise and counterclockwise with no restrictions. On each pivot, this module gives a sign which we can utilize to discover the direction of the turn.



This module has additionally a catch which we can change his state by squeezing the handle or the rotating encoder down. We can even get the sign from the catch which is Mostly modified to reset their turn check down to 0. Let's a discussion about the pins. The pins in this module are five. We have the ground pin with a gm D sign, the voltage pin with an or more sign. What's more, obviously we have three sign pins named SW, dt, and CLK. So we should discuss the sign. This module gives three computerized flags so we can associate the agony of the sign with any advanced port of various microcontroller sheets like Arduino or Raspberry Pi.

## CONNECTIONS ARDUINO:

CLK = [Pin 3]

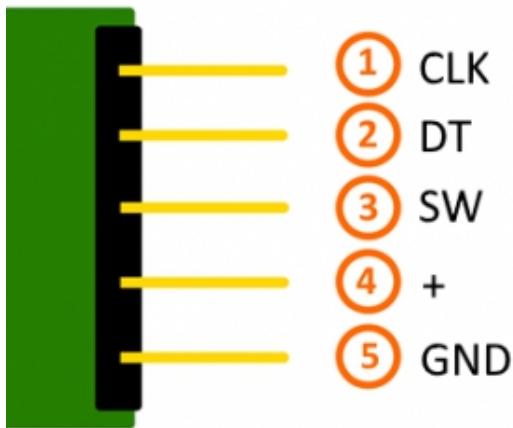
DT = [Pin 4]

Button = [Pin 5]

+ = [Pin 5V]

GND = [Pin GND]

## INPUT:



## EXAMPLE SENSOR CODE:

```
// Initialization of the needed variables<br />
int Counter = 0;
booleanRichtung;
intPin_clk_Letzter;
intPin_clk_Aktuell;

// Definition of the input-pins
intpin_clk = 3;
intpin_dt = 4;
intbutton_pin = 5;

void setup()
{
    // Initialization of the input-pins...
    pinMode (pin_clk,INPUT);
```

```
pinMode (pin_dt,INPUT);
pinMode (button_pin,INPUT);

// ...and activating of their pull up resistors
digitalWrite(pin_clk, true);
digitalWrite(pin_dt, true);
digitalWrite(button_pin, true);

// Initial reading of the Pin_CLK
Pin_clk_Letzter = digitalRead(pin_clk);
Serial.begin (115200);
}

// The program checks, which of the status pins have changed first,
to calculate the rotational direction,<br />// after a change of the pin
status was detected.

// You will get this information if you compare the pin status of the
current run with the pin status of the run before.<br />// After
detecting the rotational direction, the steps from the start position will
be counted and outputted.<br />// Pushing the button of the rotary
encoder will reset the current position.

void loop()
{
    // Reading of the current status
Pin_clk_Aktuell = digitalRead(pin_clk);

    // Check for a Change
if (Pin_clk_Aktuell != Pin_clk_Letzter)
```

```
{  
  
if (digitalRead(pin_dt) != Pin_clk_Aktuell)  
{  
    // Pin_CLK has changed first  
    Counter++;  
    Richtung = true;  
}  
  
else  
{    // ElsePin_DT changed first  
    Richtung = false;  
    Counter--;  
}  
  
Serial.println ("Rotation detected: ");  
Serial.print ("Rotational direction: ");  
  
if (Richtung)  
{  
    Serial.println ("Clockwise");  
}  
else  
{  
    Serial.println("Counterclockwise");  
}  
  
Serial.print("Current position: ");
```

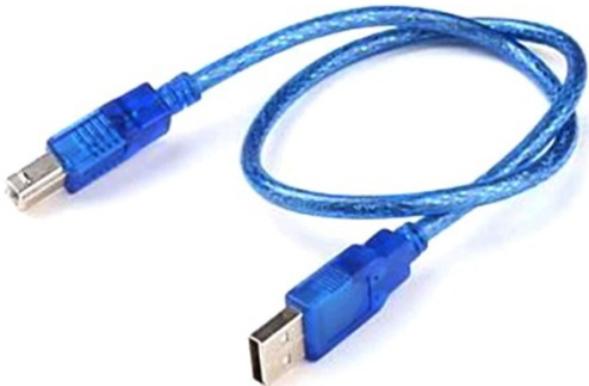
```
Serial.println(Counter);
Serial.println("-----");
}

// Preparation for the next run:
// The current value will be the last value for the next run.
Pin_clk_Letzter = Pin_clk_Aktuell;

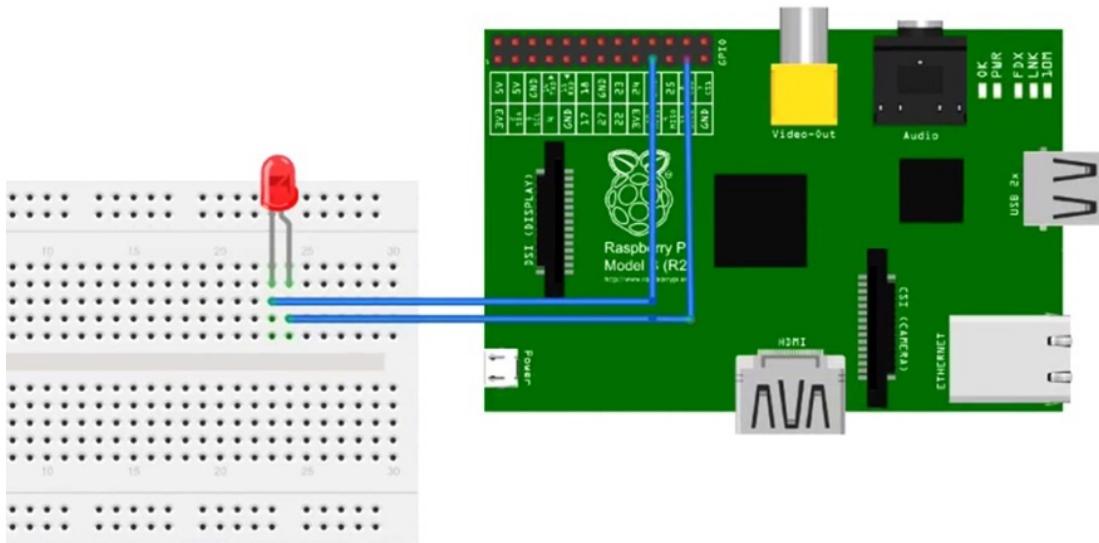
// Reset function to save the current position
if (!digitalRead(button_pin) && Counter!=0)
{
    Counter = 0;
    Serial.println("Position resetted");
}

}
```

# RASPBERRY PI CONNECTIONS



This is a cable we will use to connect Arduino to the raspberry body. So let's start. For a connection box to connect the advantage to number 11 as shown in the image below. This board cuts the bread. This is raspberry by. And this leads. We will connect this causing to be number 11 in the Army in a ball. Other treadmills will be well connected to the ground.



So let's start using fluty sized software to make this simple circuit. Now here we have software to visit circuit simulations and circuit designs. This is called Zing flight. Free. You can get it and download it go to Ceviche. I believe Billyboy. And you will find that we are proven by buying. Come on, go with a hospital by B. All worries will trust Billyboy. You can't choose all this. As you can see this in the body of the hospital is. Now we need a good one. This is good. It's Eilidh. We also need a large board ok. Now we have our board, we need all this to cause this ball. But before going further, we need to review this connection. As you can see, this round must be connected to the Arduino board land. Another trash can which should be connected to number 11 OK. Now if it's here, you will find that this is our number, this is number 9 8 11. This is the number 11 G-B or 11 as you can see. So we went here and we can connect it with this ok. There he is. We make the last bit about connections as you can see here. We have connected the land. There have been a number of those last about this advantage. This lead will blink when receiving the sentence from what we know.

## **RASPBERRY PI CONFIGURATION AND CODE**

```

import serial
import RPi.GPIO as GPIO
import time

ser=serial.Serial("/dev/ttyACM0",9600) #change ACM number as found from ls /dev/tty/ACM*
ser.baudrate=9600
def blink(pin):

    GPIO.output(pin,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(pin,GPIO.LOW)
    time.sleep(1)
    return

GPIO.setmode(GPIO.BEAD)
GPIO.setup(11, GPIO.OUT)
while True:

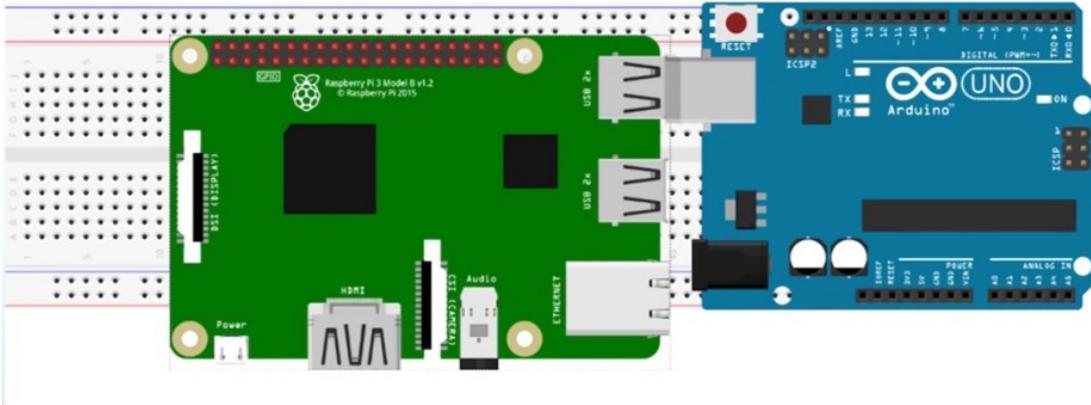
    read_ser=ser.readline()
    print(read_ser)
    if(read_ser=="Hello From Arduino!"):
        blink(11)

```

Let's start turning and go with the right spirit with and three orbin in a new window if you want. But talking about Trust Baranbar you can check our course or satisfy according to step-by-step steps to learn how to play it by using it and how to write your Won code in it. You don't need to write the following code in a new window and save it on your desktop is a good place so you don't lose it. This is the code. Let's start by explaining this very simple raspberry with the code so you will know what you write. First, we import serial communication and time libraries. We also import the input-output library that can be borne by RB which for us with the input and output barbusinput. This is what stands for this. This is to import. Next, we define serial communication so that it is the same as serial. This is the location for serial communication devices in this case Arduino. We will change this and that the fuel comes but for now, you need to understand this line and this is a change in badly 9600. This is i. M. But that is here as found from us. Dev D D D Y If we are this command, we will see a list of serial communication devices after connecting Beary we all find something like this with the iron bar afterwards. We need to change that number with the new bar after the connection. We will discuss this again in some of the upcoming lessons. C from squares is 9600. It is to sit around the level of serial communication between Arduino and the next raspberry we have a difference. It's for Blinking LED now John scattered and will choose out. But for Ben which is only a little high input-output so that it will send 5 volts or 1 logic for this so it can flash then it will wait one second and then send a low zero. Then it will wait for the second minute after that. Again to hear. If so, this is like a veteran repetition. That means back to the cable point. When this function is called that we have general goal modes and sit to connect the board and there are optimal

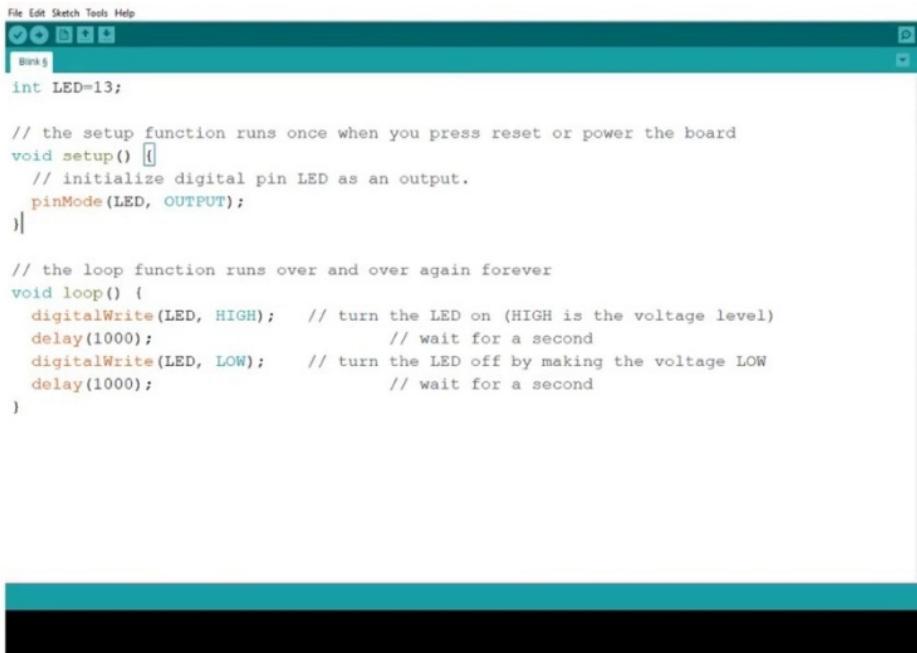
settings for bin number 11 where we connect the LED we define this Ben about it but just a little out so it can be considered as output. Now, while we need to read the Series C series line which is C from the tree line it will read the incoming data using the communication field between us better to buy and store these values with valuable research values to Brinton. But in the lead set in the window in Rasberry now if the value in the search read the same as Halo from Arduino and this sign will Blink x number 11. Number 11 and BlinkX will go here. To execute this. Then return it. So as you can see empty is a function in Bifold. This is a sublimation bytecode raspberry. Let this revision to import serial library times and input-output libraries. This is for these two lines to define the communication protocol and define the US into serial communication is connected with Albinus. Links to this data. These lines are a function called Blink when we define the ban it will use it to blink that these two lines are to define the number 11 as the output to define the board. Now this one will continue to repeat when going through it will read the communication data scene sent to the Rasberry will print it. Then it will ask if it is the same as this line it will blink x number 11 flashing means calling an empty call function and run the command at the bin number 11 so that it will go here. It will begin as high then low which means that it will live and death then it must wait for the fear of instructions. It's for the last bit of bytecode and raspberry with configuration. I know that I explain the code very quickly but because it's very simple and we don't want to waste time on this code.

## **ARDUINO AND RASPBERRY PI CONNECTION**



As you can see from this very simple picture that the basic connection scheme connects this, you ask to be born for this, you must be bought. This is very simple. We have said that we will use this cable. Who came with Arduino boards to connect these two boards. Now let's look at this and our simulation software as you see here. Let's get Arduino OK. Now, this is our board. And the reality is all yours using the cable. But for now, this is our USP and this is you can only bring them close to each other. GOOD. There he is. These three balls must be connected using the cable we mentioned before. Now connections are very simple and fundamental.

# HOW ARDUINO THINKS



The screenshot shows the Arduino IDE interface with the 'Blink' sketch open. The code is as follows:

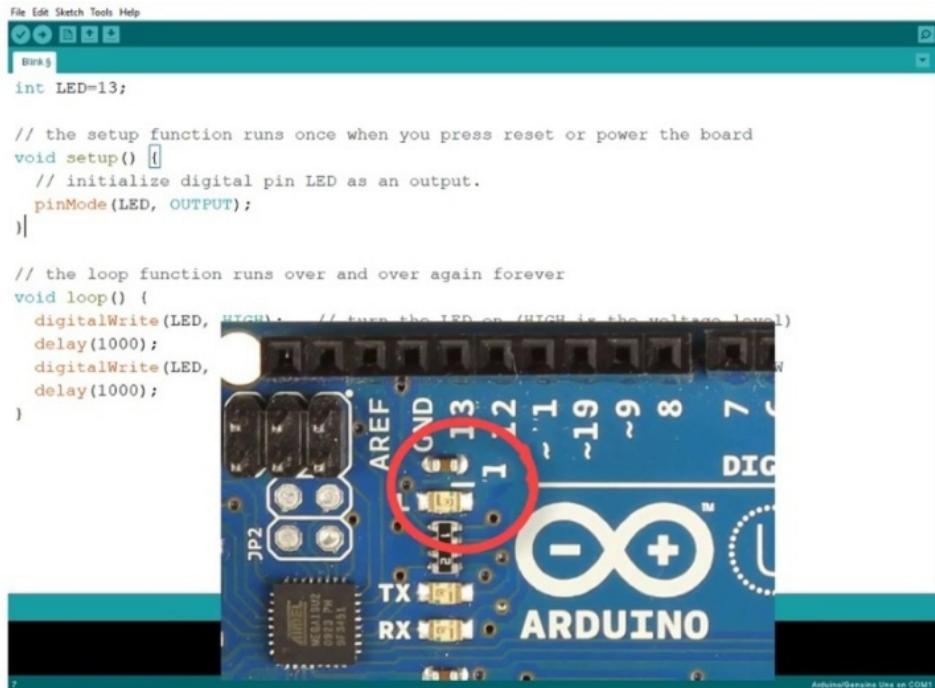
```
File Edit Sketch Tools Help
Blink: 
int LED=13;

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

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

Where we will talk about how I think so how people think of how we know it is programmed with that and zero. But hopefully, it's not a programmer we have to write code. Most programs are born through the use of C++ and C to very common computer languages. Now you might be wondering what computer language even has computer language is a tool used to develop a series of instructions.

Computers can understand. In this case, my computer and I did not submit a series of instructions that were quite a bit at least in this lesson. I look into the code how simple Arduino scripts usually start with initialization or variables to make an integer value 1 and use for example. This line of code creates an integer value of all calls and sets a value of 13. Now we need to do something with this valuable allocated. The next step is to create a function where we will use the data stored in leading to turn physically and cancel it.

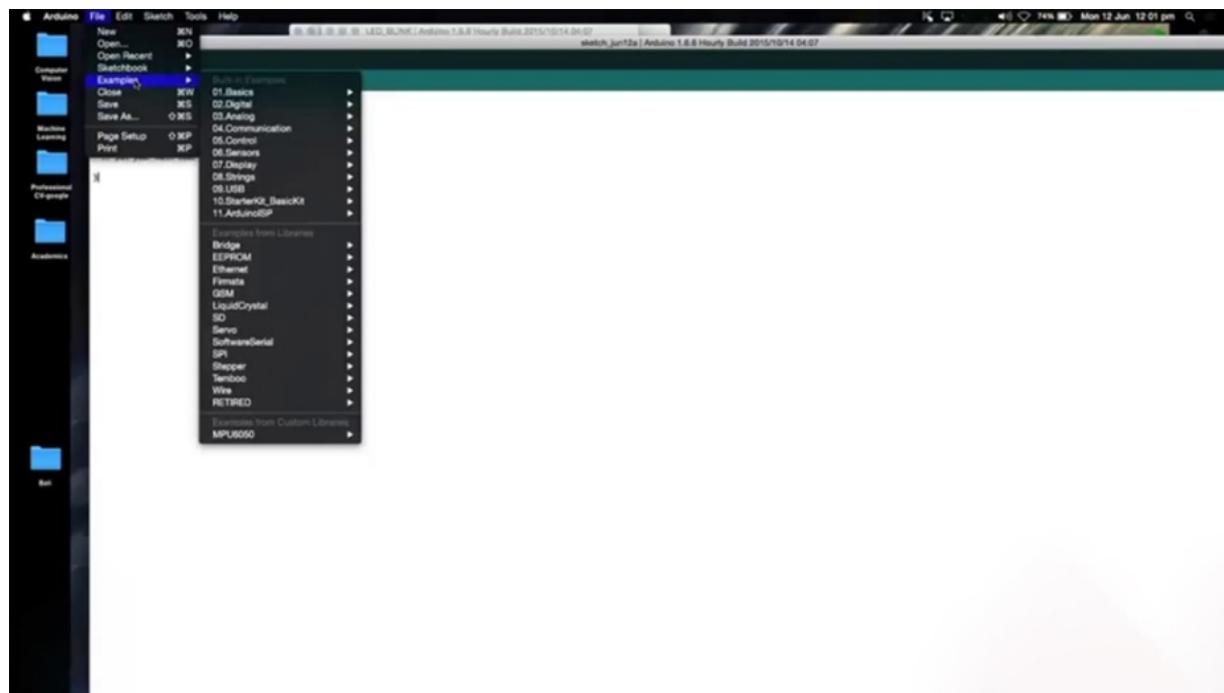


This is making a function from where there is no value that is returned hence void function in the context of this Arduino program allows you to configure any physical in or exit what and when using this line as shown here and then the mod hole this sets the bin number to 13 And you start thinking about what will be robbed. Let Arduino or what will change the settings at A because it means 30 per hour. Do you mind what we can? And this also makes it easier to change the prohibition in the future because we can only change variables to different integer densities and Adreno happens to be instant. So it's easy enough to try subprograms without additional hardware. Regardless of the device itself, you can see a circle in the image below or if you have hardware. You can't add you just like that there is still one but lose actual instructions when we know where digital.

## Set Up

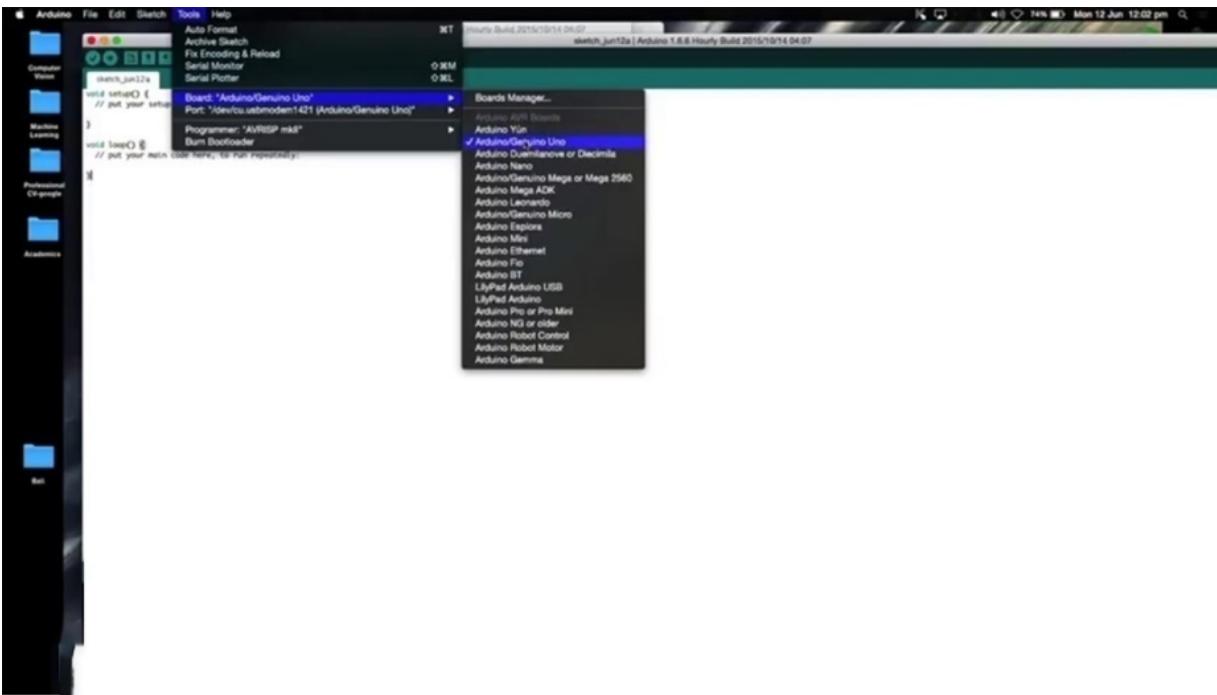
- Power the board by connecting it to a PC via USB cable
- Launch the Arduino IDE
- Set the board type and the port for the board
- TOOLS -> BOARD -> select your board
- TOOLS -> PORT -> select your port

The basic setting is the power of the board is lowered. As you can see, the power from the board comes from a PC using USB. If you are originally for testing purposes, you will connect it to a PC to upload your program. And when you run it in independent mode when your program has been uploaded on this board, you can run it from this

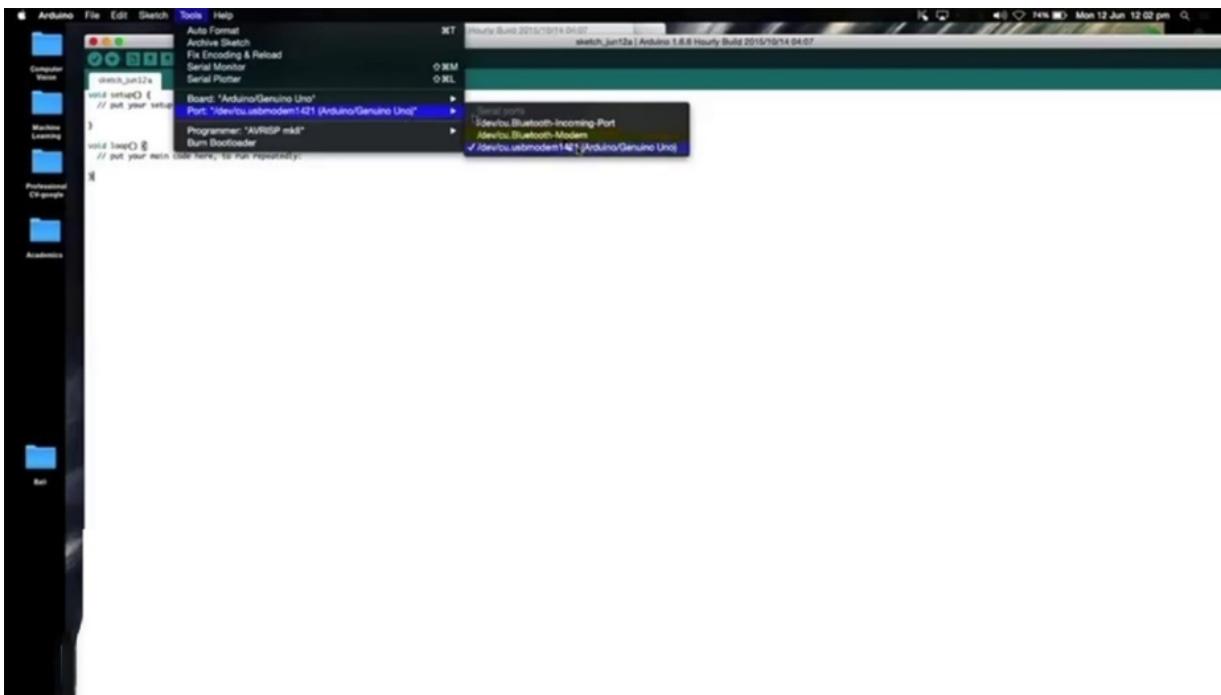


power supply input. You install a five volt DC adapter and it will function properly. Then next to the last ID then select the type of

board. Let me show you. Here I have my Arduino ID because you can see you have my basic function. This is the code verification button. This is the code upload button. Then you have a file menu, you can create a new sketch. Sketch is a program that you wrote for Arduino. Then you can open a new program, open an existing program, open the latest program and activate. You even have an example, a basic example provided with ID, which can work with various Arduino-based boards. Now move to this sketch, the most important part is this tool every time you connect your Arduino board, I have connected my Arduino board to my PC now



This tool is a board available. You can see because I use Arduino Uno it has been automatically selected but if it is not automatically selected you can choose the appropriate board



Arduino Uno. For my Mac, this port is USB Modem 1421 You can see Arduino Uno already selected. Now we are ready. Okay, one more important thing is here you can see this button this is a serial monitor. This is one of the good Arduino features that when executing serial programs, you don't need to have such an external console or software, you can use the default serial monitor to see the program. Now after the board and port are properly selected,

## Set up (contd..)



As I said, you choose the board then the color port responds on your PC for a Windows-based system, it will be more or less it will show you fighter sports. This might progress to 1015 whatever you choose correctly.

# Arduino IDE Overview

Program coded in Arduino IDE  
is called a SKETCH



Then

I've shown this to you. Your Arduino sketch. When you remember the program written into Arduino, because it was written for Arduino called a sketch. This mainly consists of two parts one is setting, and one is called a loop. Setup is analogous to normal main programs, C or C ++ programs. The main function you use is analogous to the settings

## Arduino IDE Overview (contd..)

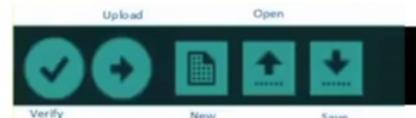
- To create a new sketch
    - File -> New
  - To open an existing sketch
    - File -> open ->
  - There are some basic ready-to-use sketches available in the EXAMPLES section
  - File -> Examples -> select any program



functions at Arduino. And, as the name suggests, the loop function is used for an iterative looping over instance. it's more or less common than the file you click on the new one

## Arduino IDE Overview (contd..)

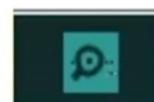
- Verify: Checks the code for compilation errors
- Upload: Uploads the final code to the controller board
- New: Creates a new blank sketch with basic structure
- Open: Opens an existing sketch
- Save: Saves the current sketch



This will open a new file and you can try various examples and sketches. We have AI for this one, this is the verification button. The main feature is before uploading our code, if you have syntactic errors or logical errors like that, it will be arrested during verification it will say your compilation fails. After you pass this verification check, you can upload your code.

## Arduino IDE Overview (contd..)

- Serial Monitor: Opens the serial console
- All the data printed to the console are displayed here



okay. Now, this we talked about this is the serial monitor, whatever data is transmitted through the serial port is printed on the serial monitor.

## Sketch Structure

- A sketch can be divided into two parts:
  - Setup ()
  - Loop()
- The function setup() is the point where the code starts, just like the main() function in C and C++
- I/O Variables, pin modes are initialized in the Setup() function
- Loop() function, as the name suggests, iterates the specified task in the program



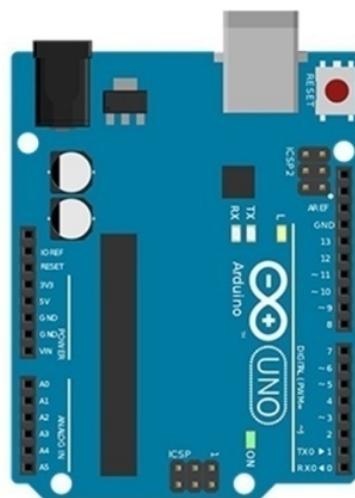
```
File Edit Sketch Tools Help
HelloArduino
void setup() {
    Serial.begin(9600);
}
void loop() {
    Serial.println("Hello Arduino!");
}
```

The sketch structure, as I said, it consists of two separate parts and the loop section. The function setting is the point where Arduino's compiler starts the code. It's like a Western animal with the main function in C and C ++ and various input-output variables in mode do you need to be as you remember you have 14 digital output pins. You must explicitly tell your system whether you want to use a PIN in reading mode or input mode or output mode. Then the loop function is used for iterations. In this code example you can see we only use the inbuilt serial port serial port, we only write dot serials starting 90 690 600 is a baud rate you can have various levels of Baud. We will arrive at us to discuss it in consecutive lectures. You can have various limits for different systems but 9600 more or less than Baud levels commonly used for most systems. And in a void loop, you want to iteratively the Hello Arduino loop. Serial Dot Print LN that if you write a dot print series, it only prints string Hello Arduino. If not, if you write LN print is a new line. This will print Hello Arduino in a new

## Supported Datatype

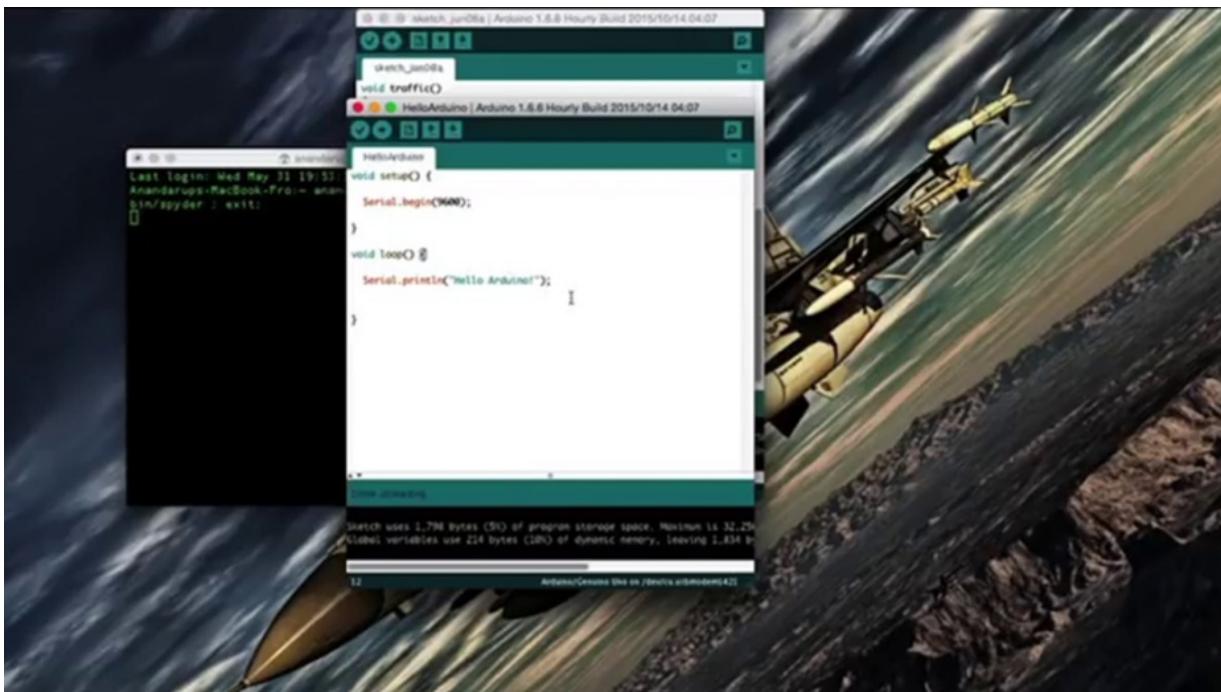
- Arduino supports the following data types-

Void	Long
Int	Char
Boolean	Unsigned char
Byte	Unsigned int
Word	Unsigned long
Float	Double
Array	String-char array
String-object	Short



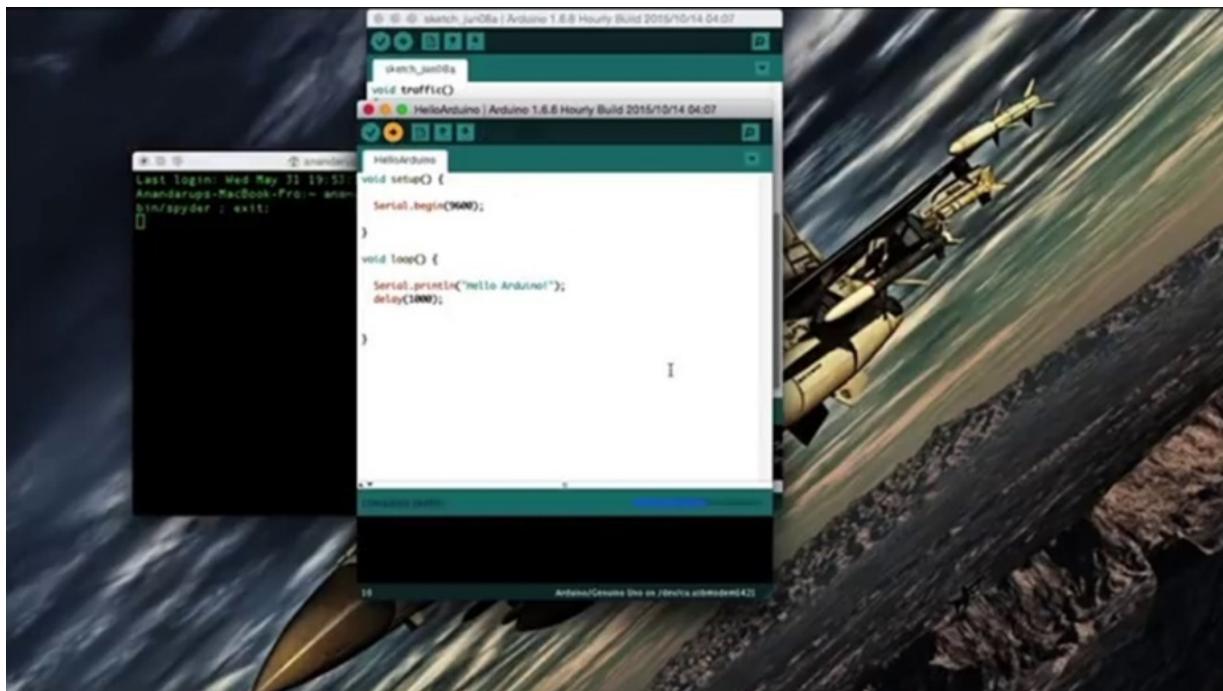
line.

Before this will see the sample code. As you can see in this Hello Arduino code in word settings, we have written dot serials starting 9600 and in a loop void, we just wrote serial dot print In, hello arduino.



Now before doing whatever verifies the code. As you can see, it compiles sketches. If your code compilation is correct, it's true. It only shows how much memory is used and all that. If if you are

wrong, suppose I remove this coma point now I verify the sketch it will give you a mistake. This is a good practice before uploading it blindly you only verify your code. Okay, now the code verification is successful. The port has been selected, I upload my code. Now compile sketches and upload it to Arduino boards. Now, the code has been uploaded to the Arduino board because the function of this program is to print Hello Arduino on the serial port iterate will open a serial monitor. As you can see, it scored Hello Arduino. Good? Quick enough, we can modify it will install a function called delay,



Say a delay of one second. Thousands are delays in milliseconds we will delay one second. The code has been verified, we upload it again. Now we open a serial monitor again. Can see now the delay has increased mold after one second I hope this is easy. Now let's move on to the next menu. Okay, like other programs, Arduino AI supports various types of data that you have in the word bytean bytes, float arrays, string objects, cat long cats that are not signed. This is rather similar to your normal C program.

## Arduino Function Libraries

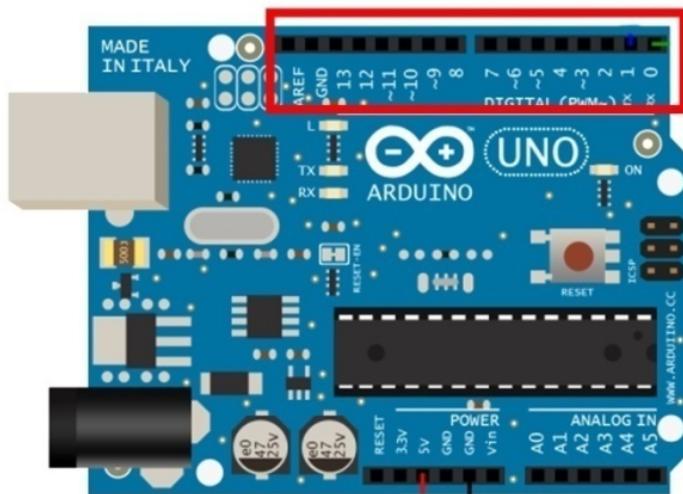
- Input/Output Functions:
  - The arduino pins can be configured to act as input or output pins using the `pinMode()` function

```
Void setup ()  
{  
    . . .  
    pinMode (pin , mode);  
}
```

Pin- pin number on the Arduino board

Mode- INPUT/OUTPUT

Arduino has many and many libraries. Because this is an open source platform. Collaborative people al-people and companies and organizations, they upload their own Arduino library. For most functions will get easy access to various libraries. As you know, our PIN can be configurated to act as input or output depending on your needs. To do this, PIN mode this function is used. You can see the syntax pin mode, pin, comma mode. This pin is a number, the actual pin pin on the Arduino board as you can, if you focus on this board, you will see the various pins I have written here 123 because this is a digital pin, it's digitally written. In pin mode on the PIN, you simply write the PIN number.



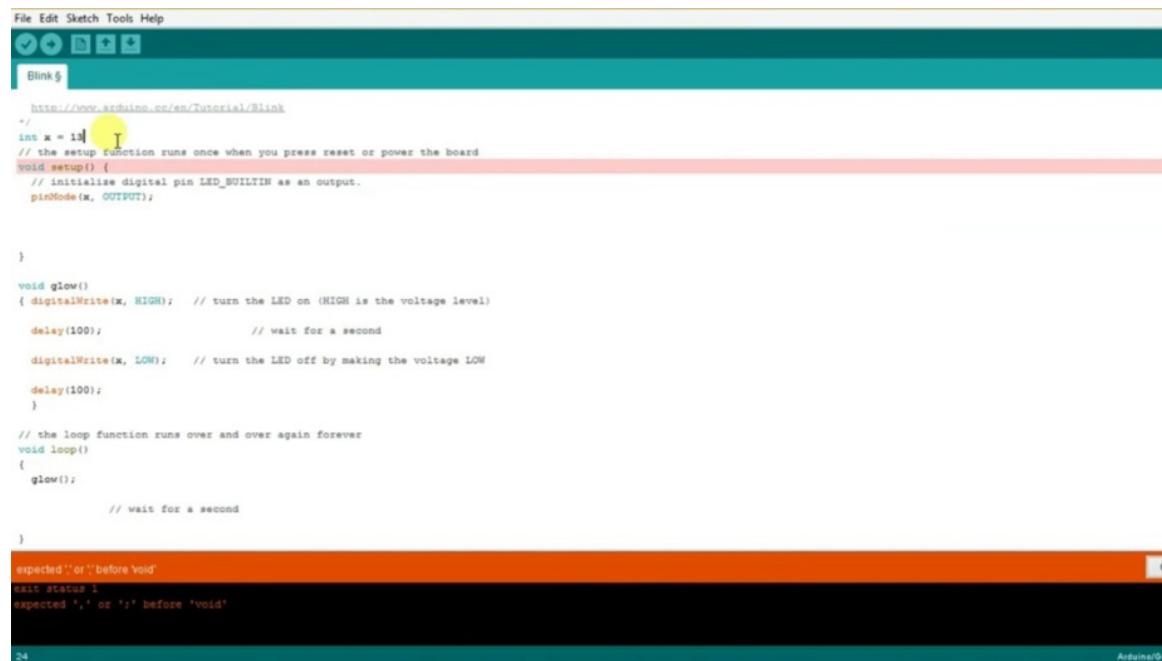
Simple and your mode just write input or output if you want a PIN to work in input mode like you connect various sensors, which you will get sensory input. You put a pin in input mode and if you want to activate something, maybe the lights on the LED or motorbike, you put the pin in the output mode.

### Arduino Function Libraries (contd..)

- `digitalWrite()` : Writes a HIGH or LOW value to a digital pin
- `analogRead()` : Reads from the analog input pin i.e., voltage applied across the pin
- Character functions such as `isdigit()`, `isalpha()`, `isalnum()`, `isxdigit()`, `islower()`, `isupper()`, `isspace()` return 1(true) or 0(false)
- `Delay()` function is one of the most common time manipulation function used to provide a delay of specified time. It accepts integer value (time in milliseconds)

## ARDUINO SYNTAX WITH EXAMPLE

We will understand the basics of programming or I mean to say the syntax used in Arduino programming. Okay and let's start college. It's in three modules and we will start. Okay, let's look at the syntax, syntax as I said before that the syntax of this program is similar to grammar in English. Okay, let's look at the important points. First of all, here we can see a semicolon in programming defined as a termination statement. For example, if we choose integers and define the value to variables, just say we have defined the value of integer x equal to 13.



```

File Edit Sketch Tools Help
Blink 5
http://www.arduino.cc/en/Tutorial/Blink

int x = 1; // the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);

}

void glow()
{
  digitalWrite(x, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(100);           // wait for a second
  digitalWrite(x, LOW); // turn the LED off by making the voltage LOW
  delay(100);
}

// the loop function runs over and over again forever
void loop()
{
  glow();
  // wait for a second
}

expected ';' or 'void'
exit status 1
expected ',' or ';' before 'void'

```

But if we try to compile it, after being compiled, you can see that there is an error. There are eight simple necessary commas or semicolons after the statement so that it can stop okay. Let's put a semicolon or date. Now see that it's perfect, but something to remember that always after calling a function, there should be no semicolon.

The screenshot shows the Arduino IDE interface with a sketch named "Blink". The code is as follows:

```
File Edit Sketch Tools Help
Blink:1
http://www.arduino.cc/en/Tutorial/Blink

int x = 13;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);

}

void glow()
{ digitalWrite(x, HIGH); // turn the LED on (HIGH is the voltage level)

  delay(1000);           // wait for a second

  digitalWrite(x, LOW); // turn the LED off by making the voltage LOW

  delay(1000);
}

// the loop function runs over and over again forever
void loop()
{
  if(x>d)
  {
    ;
  }
}

expected unqualified-id before '}' token
exit status 1
expected unqualified-id before '(' token
```

A yellow circle highlights the opening brace '{' at the end of the 'if' statement. The error message 'expected unqualified-id before '}' token' is displayed below the code.

If you place a semicolon after the function is stated, it will be said that it is impossible or will show it will give an error. Good. Now the next thing is that if we call something statement like a statement, for example, say if, just say if Xis is greater than five, remember something after calling a statement, there should be no coma point sign that will be stopped with the help of two curly brackets, it should be There is no semicolon that must be placed in it.

The screenshot shows the Arduino IDE interface with a sketch named "Blink". The code is as follows:

```
File Edit Sketch Tools Help
Blink:1
http://www.arduino.cc/en/Tutorial/Blink

int x = 13;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);

}

void glow()
{ digitalWrite(x, HIGH); // turn the LED on (HIGH is the voltage level)

  delay(1000);           // wait for a second

  digitalWrite(x, LOW); // turn the LED off by making the voltage LOW

  delay(1000);
}

// the loop function runs over and over again forever
void loop()
{
  if(x>d)
  {
    ;
  }
}

'd' was not declared in this scope
exit status 1
'd' was not declared in this scope
```

A yellow circle highlights the semicolon ';' at the end of the 'if' statement. The error message 'd' was not declared in this scope' is displayed below the code.

And other important things their coma in other important functions are dead. Our founder's curry wire must be a simple coma point placed or it will give us a mistake. First of all, let's look at the correct version of it. This gave us a mistake, the mistake was invalid, D was not stated. Okay, let's explain to give us five. Okay, let's just compilation. It compiled perfectly just removing the semicolon after that. This will give us okay mistakes it doesn't.



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** File Edit Sketch Tools Help
- Sketch Name:** Blink 5
- URL:** http://www.arduino.cc/en/Tutorial/Blink
- Code Area:**

```
File Edit Sketch Tools Help
Blink 5
http://www.arduino.cc/en/Tutorial/Blink

int x = 13;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);

}

void glow()
{
  digitalWrite(x, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(x, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);
}

// the loop function runs over and over again forever
void loop()
{
  if(x>5)
  {
    x=5;
  }
}
```
- Status Bar:** Done compiling
- Message Bar:** Sketch uses 622 bytes (1%) of program storage space. Maximum is 32256 bytes.  
Global variables use 11 bytes (0%) of dynamic memory, leaving 2037 bytes for local variables. Maximum is 2048 bytes.
- Bottom Right:** Arduino IDE

You can say that you can make an empty statement from it no need to put a semicolon sign but for better practice that after the statement is given, like X equal to three, you must stop it with a semicolon. Let's just check it. Well, the coma is there. The termination of a statement with a coma point sign comes it makes it compile the level for compilers and can be converted into hexadecimal values that are inserted into the cattle inserted into the micro or okay antelope, let's just sit. After that, we can see that there are two backslashes. Were they? Let's look at the backslash in Reno used to comment.

```
File Edit Sketch Tools Help
Blink
by Colby Newman

This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink

/*
int x = 13; // this is x that isequal to 13
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);
}

void glow()
{
  digitalWrite(x, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(100);           // wait for a second
  digitalWrite(x, LOW); // turn the LED off by making the voltage LOW
  delay(100);
}

// the loop function runs over and over again forever
void loop()
{
  if(x>5)
  {
    x=3;
}

Done compiling

Sketch uses 622 bytes (1%) of program storage space. Maximum is 32256 bytes.
Global variables use 11 bytes (0%) of dynamic memory, leaving 2037 bytes for local variables. Maximum is 2048 bytes.
```

For comments, I want to tell you that if I make something like X equal to five, there is no problem x given the same as the value to just correct, let me correct. I will see the first line, you can see that integer x is equal to 13. But if you want to notify some other programmers or if you want to tell yourself that what your return is, you can write a simple line in English without Make it enter the compiler. Give two semicolons after that, give it to do two backslashes after that and write something the thigh is the exit that is equal to 13 lines won't enter the compiler.



The screenshot shows the Arduino IDE interface with the title bar "File Edit Sketch Tools Help". Below the title bar is a toolbar with icons for file operations like Open, Save, and Print. The main area is titled "Blink §". The code editor contains the following text:

```
File Edit Sketch Tools Help
Blink §

// This sketch blinks an LED connected to digital pin 13.
// Turn on 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 many boards
// 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/Hardware/Specs

// modified 8-May-2014
// by Scott Fitzgerald
// modified 1-Sep-2014
// by Arturo Guadalupi
// modified 6-Sep-2014
// by Colby Kaauwa

// This example code is in the public domain.

// This sketch starts by setting up pin 13 as an output
int x = 13; // this is x that is equal to 13
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);
```

---

The line is perfect, but what happens if you want to compile several line numbers here you can see that there are several liner numbers compiled. So let's just practice. Thighs are done with the function of backslash an Asterix for the beginning and something important must be remembered that backslash then now for termination, it must be done with a reverse slash for termination of lines, it must be done with the help of Asterix and backslash. Index symbols, backslash and Asterix to study sticks and backslash blacks for the right termination.

File Edit Sketch Tools Help

Blink §

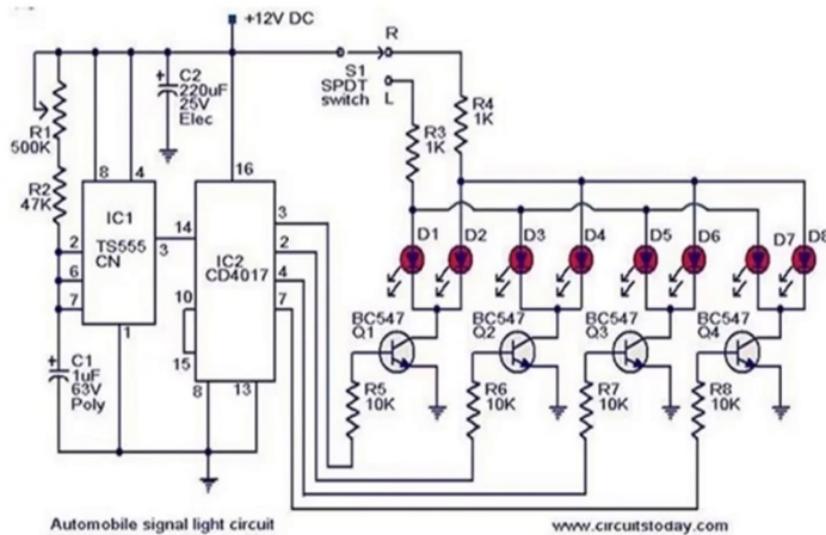
```
/*
 * hello
 * world
 * this is a blink
 */
int x = 13; // this is x that is equal to 13
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);
}

void glow()
{
  digitalWrite(x, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(100);           // wait for a second
  digitalWrite(x, LOW); // turn the LED off by making the voltage LOW
  delay(100);
}

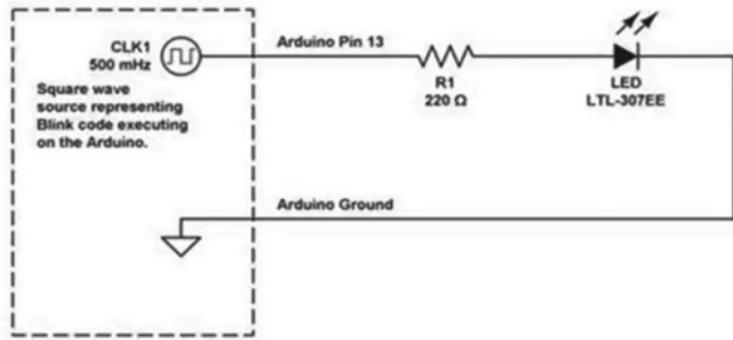
// the loop function runs over and over again forever
void loop()
{
  if(x>5)
  {
    x=5;
  }
}
```

# ASSEMBLING A CIRCUIT

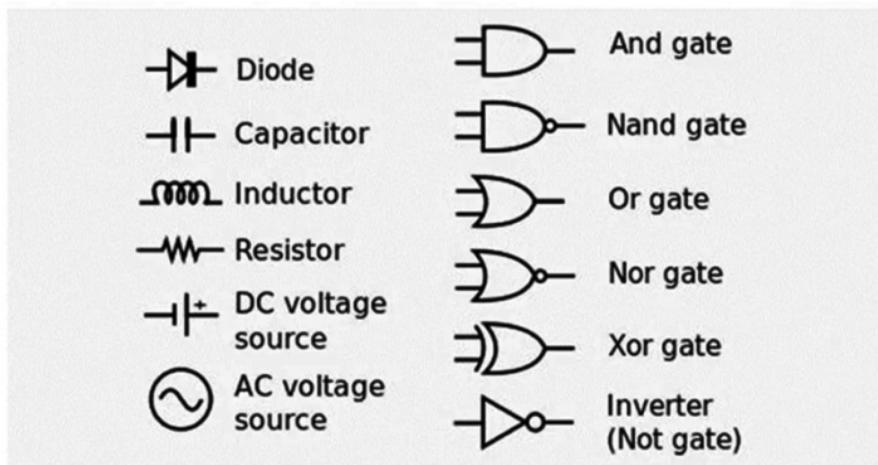
So now you know what Arduino boards and cables. Time to combine it into a set. This is a centralized picture of the bread and I will explain it to you below. By dragging to live to you for the breadboard as it says only you can have a circuit that allows you to make a blink diode emitted lightly. So this is quite clear even though it might be a little confusing. So I will tell you exactly what you mean. First, let me show you the toolbox requires you not get used to it.



This is the link below not that one of the metal trash canes is longer than the other. That's important. You have to remember it. Another thing is a disaster. The eyes of the two sisters rather complicated. And to fully understand it, I have to explain things like now and voltage. However, it is not required to understand the use of circuits does.



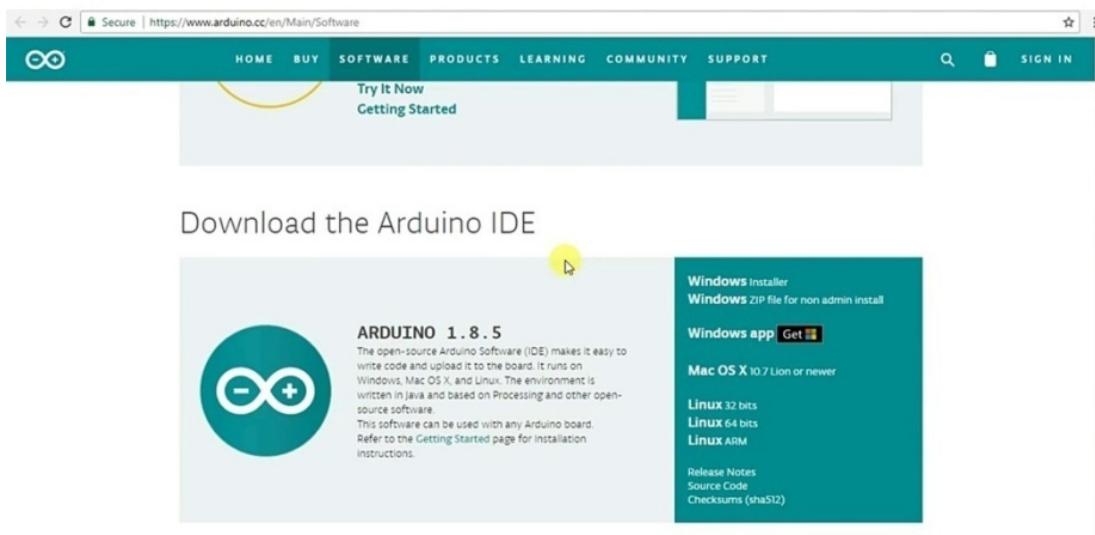
Because the Adreno platform has so much support on the internet that you should easily find a diagram that tells you the right resistor what you need and where to put it. To make the program take it but one big charge is like what looks like a resistor. Also, note that they have a line on them and the lines also beat on the connection above so you know the direction where the resistor must be installed. So let's bring the photo again. Now you have seen Bart. So basically what you have to do is plug your LEDs longer to meet carbon in 2009 and the shorter to 10 resistors that are three hundred thirty sleeves according to the need for circuits then connected to a reduced anode. Cullum. You can't except the resistor at Annie Hall in the Minus column because we know that they are all connected.



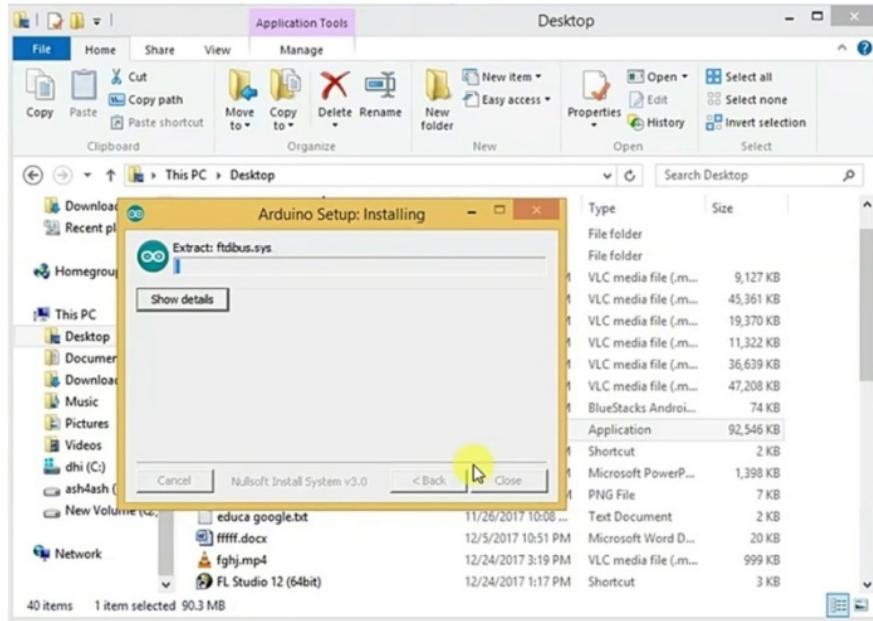
Now it's enough to connect the way out if it means we have a way out of line 9 connected to Kensington and then I will use the cathode connected to 5 errors and 1 2. I will not get there. Do we know how all the small holes labelled like on the right is number 13 on the left is 500? Also on that list, there is a ground. All you have to do is block the corresponding cable and as soon as you plug your cable into the true Haul, you have to do a few more steps to make the rollout blink. At first, you must plug in the wire to Arduino. Then you must upload a certain card to Arduino to make you flash. Now we will learn how to make it blink. Thank you for watching this lesson. If you have questions, please ask me and I will see you at the next lesson.

# ARDUINO SOFTWARE FOR THE FIRST TIMERS

The next module is the Arduino software for the first time. In the thigh module, we will discuss Arduino and what is the red board. Let's start with the download of programming and installing software in an integrated development environment and writing our first program that blinks LEDs or you can greet the world. Before going there. I want you to go to the graph to go to Google and look for Arduino.



After searching Arduino, you can click on the Arduino Arduino button home and after doing it, just click on the software after clicking the software you must choose your type of installation. It can be a window that can be a Mac or can it be Linux for me I have chosen the Windows Installer after clicking that you will get to the next page that tells the contributor to download or only for our case now we will only download after downloading Arduino software will be good okay I have downloaded it before. So you can see here how the Artemis installation is okay. Thighs are Arduino 1. 8 points four for Windows.



So it is a file that can be executed. Click just and it will start asking security questions, click Yes on it. Click I agree. Click on the next click on the location it will request a location where the agenda must be installed. For me, he saw the program policy and the Arduino folder name. After clicking it, it will only start installing. If you want to see how things and how the library or file will stop, just click Ongood as you can see here, either in one minute will be installed. Well, I will only ask the driver to be installed in Arduino. So just close. Good. Now on the desktop, you can see that there is an icon, named Arduino. Okay, just look there. If you don't find an icon, don't worry. Go to C. Go to the X86 Files Program, click on Arduino and there you can find an icon that Arduino e Xe just makes a shortcut or have it on textiles and you can start okay, click it will request access to the Java file click Yes. And now you can see RD. Thighs are integrated development environments for Arduino here you can see that it has all the features needed for us for the board if you can park water that you can find here port and all the buttons needed.

## Content

- Operators in Arduino
- Control Statement
- Loops
- Arrays
- String
- Math Library
- Random Number
- Interrupts
- Example Program

We will discuss basic topics, operators in Arduino, control statements, loops, arrays, strings, math libraries, random numbers interruptions and our example program that will be a little complicated than the previous one.

## Operators

- Arithmetic Operators: =, +, -, \*, /, %
- Comparison Operator: ==, !=, <, >, <=, >=
- Boolean Operator: &&, ||, !
- Bitwise Operator: &, |, ^, ~, <<, >>,
- Compound Operator: ++, --, +=, -=, \*=, /=, %=, |=, &=

So, the basic operator as a normal programming C, C ++ or Python or another language you have the same base as the multiplication division operator plus-minus then the operator comparison if you are equal to not the same as fewer than, bigger, and all operators, Then you have a Boolean operator Bitwise and a compound operator.

Moving to the control statement, this will cover various examinations and looping. So, the FL statement is normal in Arduino, you start with an IF statement. So, if you have a condition and in this curly bracket, if the conditions are stated right, if other if the other statement conditions are correct, or if there is no statement above the correct, then this loop will be executed. Move to the Case Switch, you have switches and Joyce has one case option and statement and then the rest function.

## Control Statement (contd..)

- Switch Case
  - Switch(choice)

```
{  
    case opt1: statement_1; break;  
    case opt2: statement_2; break;  
    case opt3: statement_3; break;  
    .  
    .  
    .  
    case default: statement_default; break;  
}
```
- Conditional Operator.
  - Val=(condition)?(Statement1):(Statement2)

For each case, so, the case option for your statement then breaks and so on. And in the end, you have a default case after that, you again have a rest function, then you have a conditional operator it will avoid the use of conditional operators like this in Arduino. So, it is a condition if it is true, it will execute the statement of others will execute the statement of two types of this statement that is best avoided during Arduino programming.

## Loops

- For loop
  - `for(initialization; condition; increment){`  
Statement till the condition is true;  
}
- While loop
  - `while(condition){`  
Statement till the condition is true;  
}
- Do... While loop
  - `do{`  
Statement till the condition is true;  
`}while(condition);`

So, in the loop, you have a base for loops then you have a temporary loop, you have a do-while loop. This is a pretty common example you have a nested loop which is a loop in another loop you can have a lot of looming loops in each other. So, they will have an infinite loop

## Loops (contd..)

- Nested loop: Calling a loop inside another loop
- Infinite loop: Condition of the loop is always true, the loop will never terminate

So, to run unlimited loops, for example, you all you need, you develop a system where you need to turn on and off or lights or LEDs or other distant devices as long as your device is checking.

So, remember from the last lecture, which I showed an example of a flashing LED. So, you can see, if we put it in an unlimited loop, as long as the Arduino board is empowered, it will continue to blink. So, your function can be made more complicated. Instead of LEDs, you can have motors instead of a motorbike, you can have a camera mounted on the motorbike and they keep spinning you have you can have many sensors associated with cameras and motorbikes. So, you can get an example, you can build a security system that will continue to run as long as your processor works. Alright, and strength is being supplied You can always connect it to the battery supply to produce power for it.

## Arrays

- Collection of elements having homogenous datatype that are stored in adjacent memory location.
- The conventional starting index is 0.
- Declaration of array:

```
<Datatype> array_name[size];  
Ex: int arre[5];
```

Then you have an array, an array is a collection of elements that have homogeneous data types and stored in and only in memory locations. The conventional initial index is zero in Arduino. So, the declaration of your array just started with the data type it might be in the form of an integer array, so the name array int and its size. So, for example, in an array, this is an array of the name of the variable five will allocate five spaces for your array.

## Arrays (contd..)

- Alternative Declaration:

```
int arre[]={0,1,2,3,4};  
int arre[5]={0,1,2};
```

- Multi-dimentional array Declaration:

```
<Datatype> array_name[n1] [n2][n3]....;  
Ex: int arre[row][col][height];
```

Then you can have an alternative declaration. Suppose this Int Array and this empty bracket is the same as in this curly bracket you have 01234 so, this will automatically in this area, once again you have entered five you can only enter three variables three values in this array and the rest will still occur empty, maybe to use later, you can fill out also when you have a multi-dimensional array declaration that is the same as the previous one, you have an array name data type, then the dimension for the first dimension let no n two n three. For example, if you want to declare an array for normal images of RGB images, so, you have three red, green and blue channels. So, each image will have a 2D structure with a line and column. And there will be depth for each RG and B. So, maybe for the data type, you have an Int Array line column

## String

- Array of characters with NULL as termination is termed as a String.
- Declaration using Array:
  - `char str[]="ABCD";`
  - `char str[4];`
    - `str[0]='A';`
    - `str[1]='B';`
    - `str[2]='C';`
    - `str[3]=0;`
- Declaration using String Object:
  - `String str="ABC";`

Then move to the string is an array of characters with null because the termination declaration might use paint strings. Here str is an array. So this ABCD is stored in the STR four structure strain and you can individually access each ID that you can save a BC or maybe zero. So this uses the same location if you want to be stored individually in different locations. Sorry to come to the same location. If you continue to save this, the latter will be the last character saved will be updated, the other will be overwritten. If you want to save in a different location, you just change it from the string zero str. strange str and so on. So you will have a consecutive ABC zero, side by side in these locations. Another thing you can have also had a data type string. So the string str is the same as ABC will give you ABC at all you don't have to save in individual locations. So this is one of the benefits of using Arduino.

## String (contd..)

- Functions of String Object:
  - str.ToUpperCase(): change all the characters of str to upper case
  - str.replace(str1,str2): if str1 is the sub string of str then it will be replaced by str2
  - str.length(): returns the length of the string without considering null

So some string functions are commonly used. So str to a big letter point to pay attention to uppercase letters, t, u and c are lid. So this must be strictly followed because this is part of the syntax, thus changing all string characters to uppercase. Then you have string str dot replace one string and string two. So, one string,

## Math Library

- To apply the math functions and mathematical constants, “**MATH.h**” header files is needed to be included.
- Functions:
  - cos(double radian);
  - sin(double radian);
  - tan(double radian);
  - fabs(double val);
  - fmod(double val1, double val2);

If it is a structural substring then it will be replaced with the length of the STR N DOT it returns the string length without considering zero characters. Then another library commonly used is a mathematical

library to apply mathematical functions, dot H math headers must initially be called otherwise, you will not be able to access these functions. So, some common functions are COS, which in double radians in sinus, tan, absolute fabs floating, mod floating right. So one double value and double value. So, you have two values and mod f will give you a modular division and the results will be floating-point numbers. Then, it continues with the MAT library, you have XP which indicates six by the exponential function you have a log function, this will give you a natural logarithm of the value then you have a log 10 and you have a square function power function. The first argument is the base, the second argument signifies strength.

## Random Number

- randomSeed(int v): reset the pseudo-random number generator with seed value v
- random(maxi)=gives a random number within the range [0,maxi]
- random(mini,maxi)=gives a random number within the range [mini,maxi]

then you have a log 10 and you have a square function power function. The first argument is the base, the second argument signifies strength.

Then another example commonly used is random numbers. So, one of the functions of this random number is a random seed. So, the syntax is a random seed because the capital you need to focus on is because this is the default syntax for Arduino. So, Random Seeds Int V rearranges the PSEUDO-random number generator with the value of B. So, you already have a seed value is the starting point where random numbers will initialize their functions, you provide an initial

value from it, a random number will produce, then Mac side random, giving random numbers in the zero range to the high max, then you have a random mini and maxi, it gives random numbers in the range, min and max.

## Interrupts

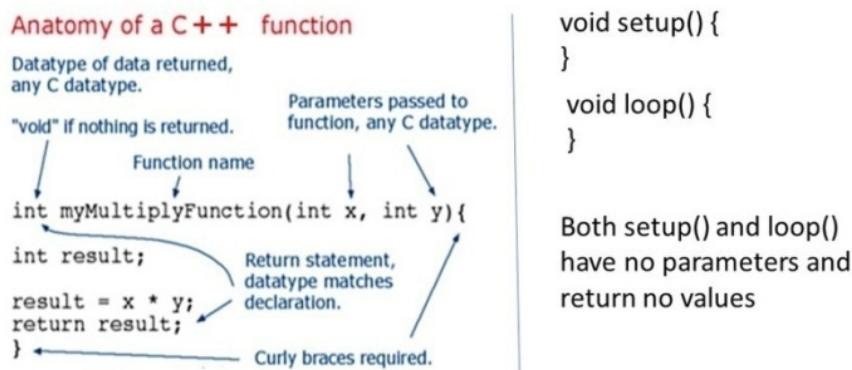
- An external signal for which system blocks the current running process to process that signal
- Types:
  - Hardware interrupt
  - Software interrupt
- `digitalPinToInterrupt(pin)`: Change actual digital pin to the specific interrupt number.
- `attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);`
  - ISR: a interrupt service routine have to be defined

Then move to your interruption has an external interruption signal this is an external signal that the system blocks the process running now until it receives the signal. So, you have two types of interruptions one hardware and the other is software. So, I will give you an example. Suppose, you are in a circle, you are waiting for the inspection condition, whether the inspection conditions apply or not. And maybe from external sources. You get check-in conditions, for example, you have a digital button or digital switch connected to our Arduino board. So, every time you press the switch, your system will flash the LED if not, it will turn off the LED. So, this can be considered partially considered interrupted. So, this will be an external interruption. So, as you can see the digital pin for interruptions and then the pin changes the digital pin to a specific interrupt number, then attach the digital pin interruption to interrupt then ASR mode later. So, whether we are known as an interrupt service routine, it must be defined explicitly. So, these are some more complicated functions so, we will not focus on this

# ARDUINO SYNTAX

## Functions

- loop() and setup() are procedures
- You can create your own functions



Well, the syntax you can say that syntax in programming languages is similar to grammar in English or other languages. So, syntax is a method where the program must be assembled correctly. So the compiler comes compiler can read, you can say a compiler is a machine used to understand and check the program if properly written in the syntax and then change the C language that is simple to understand the code for hexadecimal language and hexadecimal language only consists of two parts, namely 1010. Then after that it can be written into each line can be written in a hexadecimal code that contains a base of 16 to 123456789 and zero then AB, C, D, E, F, the thigh is the maximum strength of the hexadecimal value that is being written next time, about Understanding of syntax, Arduino is written in C++. So we don't need to have the basic knowledge for thigh courses because we do it in practice and can get we get a better understanding of language if you have a previous knowledge before you can do it and it will be better for us for now if you have a previous knowledge What's right for it, it will be good for you because you will be better in programming and all the other things.

Okay, declare a variable and declare the mould well I say that variables can be expressed as objects or instruments where values are included. For example, you can say that x is equal to five meaning the variable x save the value of five, you can change in milliseconds, you can say the value of x can be changed US census, one, eight or anything. Thighs are known as variables, well, what about when I told you in Arduino there are around 28 pins where there are 30 digital inputs, five analogue pins that can be converted into details. So, we have about 10 digital pins and they can work on arguments. So, we have to declare a pin with a variable. The important part is that the variable can also save the PIN value as if you have set a variable as well as I will show you an example to be written. For example, if you have written the variable x equal to 13 in you put all places or x where 13 has been returned, you can see that the program will still work because when the system goes to X, it will take digital print value x is 13. So will understand that x is the same as PIN 13. And when we place x at the point of numbers as a variable, it will also understand

```

File Edit Sketch Tools Help
File Edit Sketch Tools Help
Blink §
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
*/
x = 13; // the setup function runs once when you press reset or power
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);
}

// the loop function runs over and over again forever
void loop()
{
  digitalWrite(x, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(100); // wait for a second
  digitalWrite(x, LOW); // turn the LED off by making the voltage LOW
}

```

```

File Edit Sketch Tools Help
File Edit Sketch Tools Help
Blink §
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.

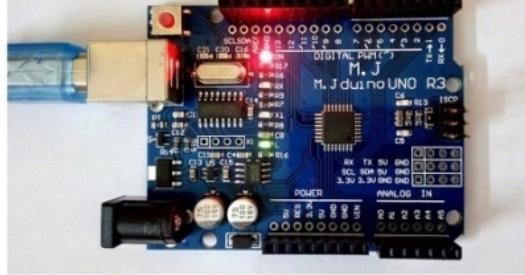
http://www.arduino.cc/en/Tutorial/Blink
*/
int k = 13;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(k, OUTPUT);
}

// the loop function runs over and over again forever
void loop()
{
  digitalWrite(k, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(100); // wait for a second
  digitalWrite(k, LOW); // turn the LED off by making the voltage LOW
}

```

So, let's take it, first of all, we have to compile the thigh error you can see that mistake because we don't have to stop, there is an error in the line because we haven't ended. So, let's end up with a semicolon. Now, please take it as a compiled version, you can say

that the value of X is not started and the thigh is done by me to notify you that you cannot declare a variable directly without giving a function without defining the function because it's not Python we have to declare every variable with The function into that place can be flexible, can be long or can be a character that I will say about it in the next section. But here we have to put value in T, okay, when we have placed the identity of the value you can see that the colour has turned into blue it means that it is the village index. Okay, now, check. There is no compile error which means that the X value is stored as a variable as a value of 13 and has been saved and every point is okay, let's just upload it to the world. First of all Deidre Connections Board, Comport Seven Boards, Arduino Uno. Now do the upload, you can see that it will and then upload.



```

File Edit Sketch Tools Help
Blink &
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
/*
int x = 13;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);

  digitalWrite(x, HIGH); // turn the LED on (HIGH is the voltage level)

  delay(1000); // wait for a second

  digitalWrite(x, LOW); // turn the LED off by making the voltage LOW

  delay(1000);
}

// the loop function runs over and over again forever
Done compiling

Sketch uses 938 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

29
ArduinoG

```

Now you can see that the flashing board means that this method is correct. Okay, now let's go to the next section. After that, you can see that the next part is to run a program at once. While the difference we must, we must know the difference between running the program at a time and run it in a continuous loop. You can see there flash finking continuously, it means walking in a circle. But what will happen if we run it is once let's just take it we have to put it in F we have to put it in the word settings, okay for a moment let me do it

coming here to cut all Ctrl X programs written here Ctrl V, You can see that after defining the program written with the same syntax. Let's compile alright after compiling it we can see that this program is written I take the microcontroller here so you can check that it has been run once or not. I have finished the program it has been running and you can see that it will blink only for one thing. Thighs are a method for programs that are running on thighs once called Void Setup. And if you want to run a program for an old Infinite you have to put it in one room. Well, you can keep the words empty but it is not possible for empty keyword settings because the PIN declaration in serial monitor calls and all functions carried out here are placed in word settings. Okay now, let's go to the next section. Here you can see that running the program several times or to run it for a long time. Okay, let's go back to the program, our ID, we can see that your ID goes here, select the syntax program in Curtir you can go to Breeneau between two curly brackets because if we don't write it between the curly bracket, this will say that That is an a-okay mistake. Once again Check Integer IDs stated in the desert, our value of our variables has been declared as the world settings have been filled with conditions when mod x is equal to 13. It's output okay, now, let's go down to the word loop you can see that on the digital word loop right x too high for 100 milliseconds, x equal to low for the next one hundred milliseconds written in the loop so that it will write it will run longer for the unlimited room okay let's check and upload this program is done and you can see that it runs for time okay okay, let's go back to another part okay after that we will come to see what function if we run the program for one time nothing but if we will call values for several times, we need a function where the program returned for one time and whenever we want to call it several times we will call the function that just SA Yes Give you a simple example.

File Edit Sketch Tools Help



```

Blink §
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
/*
int x = 13;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(x, OUTPUT);
}

// the loop function runs over and over again forever
void loop()
{
  digitalWrite(x, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(100);              // wait for a second

  digitalWrite(x, LOW);     // turn the LED off by making the voltage LOW
  delay(100);              // wait for a second
}

Done uploading.
Sketch uses 938 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

```

Okay here cut all programs with control x. Okay, after arranged before Lu V right v o id void glow g l o w you can see that Void Low has been written gives you two curly brackets, after two brackets given after it gave two curly brackets, okay. Among them, paste the program and in the word loop called the function of G, L or W gives them a parenthesis in ending with a semicolon, you can see that the function has been called here and the value has been returned here. Now, let's just take it, we've checked it out, there are no compilation errors. And we have uploaded it to the board and managed to be uploaded and you can see there flashing still ongoing. Okay, the function method is correct.

# FINISHING CONFIGURATION

```
sudo apt-get install python-serial
```

```
sudo pip install pyserial
```

Make

sure the code is uploaded to Arguido and your spirit by the interface. Be sure to activate cereal and eye to see body configuration. Go with the menu then the difference. Then tell us a little about configuration. As you can see here then activate these two to see and serial communication than just click OK. After this, you must start from Rasberry then open your terminal and run this command. The first command so that it is installed by Syria for serial communication. So they will be installed by Syria for Sterling with the telephone serial communication protocol. Now let's look at the key here is our window. Let's start the terminal OK. Now as you can see here we need to the right so they will be installed via the phone see it through OK. There seems to be a problem because I've been told this because ok we need to put it. So. So don't be kegagey about packaging. If demanding via Sylvio phone. Now let's continue and activate the web. So bib and so on with the Syrian kit it seems we need to install Bebkok. Now we need to install the configuration menu. But let's read the first sentence so I will enter the store and buy some cereals so that the pie and cereal are ready to burn.

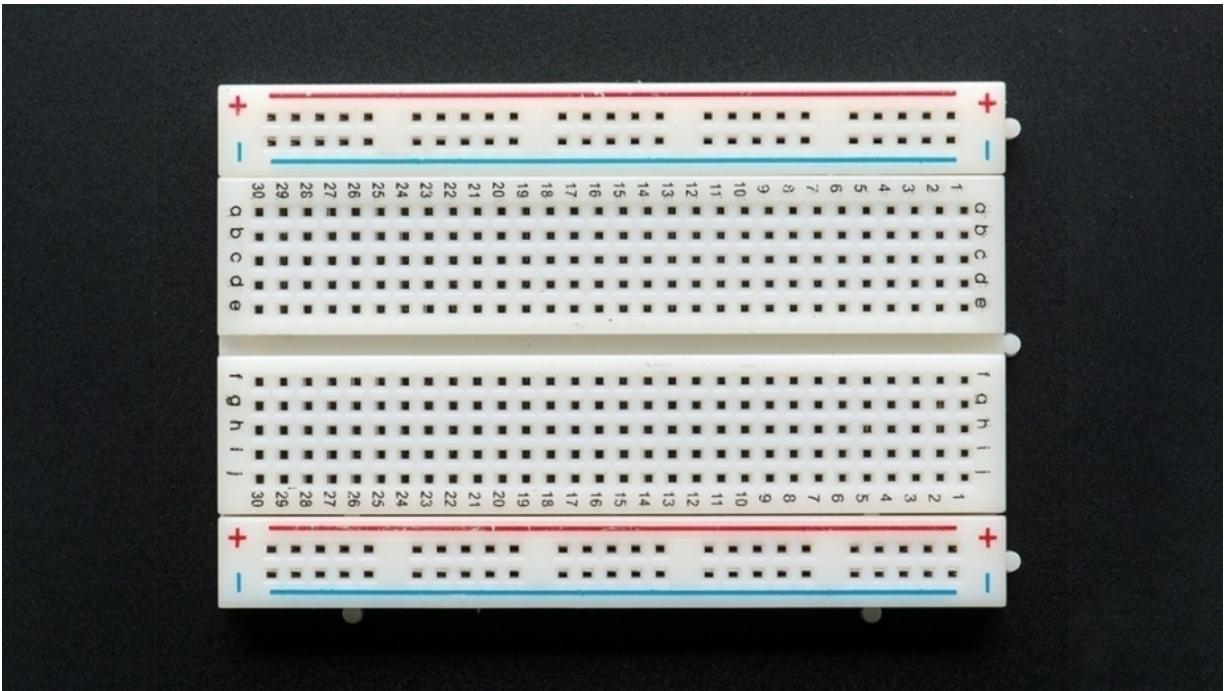
```

pi@raspberry: ~
File Edit Tabs Help
/dev/tty1  /dev/tty20  /dev/tty31  /dev/tty42  /dev/tty53  /dev/tty7
/dev/tty10 /dev/tty21  /dev/tty32  /dev/tty43  /dev/tty54  /dev/tty8
/dev/tty11 /dev/tty22  /dev/tty33  /dev/tty44  /dev/tty55  /dev/tty9
/dev/tty12 /dev/tty23  /dev/tty34  /dev/tty45  /dev/tty56  /dev/ttyA0
/dev/tty13 /dev/tty24  /dev/tty35  /dev/tty46  /dev/tty57  /dev/ttyMA0
/dev/tty14 /dev/tty25  /dev/tty36  /dev/tty47  /dev/tty58  /dev/ttyprintk
/dev/tty15 /dev/tty26  /dev/tty37  /dev/tty48  /dev/tty59
/dev/tty16 /dev/tty27  /dev/tty38  /dev/tty49  /dev/tty6
/dev/tty17 /dev/tty28  /dev/tty39  /dev/tty5  /dev/tty60
/dev/tty18 /dev/tty29  /dev/tty4  /dev/tty50  /dev/tty61
pi@raspberry: ~ $ ls /dev/tty*
/dev/tty  /dev/tty19  /dev/tty3  /dev/tty40  /dev/tty51  /dev/tty62
/dev/tty0  /dev/tty2  /dev/tty30  /dev/tty41  /dev/tty52  /dev/tty63
/dev/tty1  /dev/tty20  /dev/tty31  /dev/tty42  /dev/tty53  /dev/tty7
/dev/tty10 /dev/tty21  /dev/tty32  /dev/tty43  /dev/tty54  /dev/tty8
/dev/tty11 /dev/tty22  /dev/tty33  /dev/tty44  /dev/tty55  /dev/tty9
/dev/tty12 /dev/tty23  /dev/tty34  /dev/tty45  /dev/tty56  /dev/ttyA0
/dev/tty13 /dev/tty24  /dev/tty35  /dev/tty46  /dev/tty57  /dev/ttyMA0
/dev/tty14 /dev/tty25  /dev/tty36  /dev/tty47  /dev/tty58  /dev/ttyprintk
/dev/tty15 /dev/tty26  /dev/tty37  /dev/tty48  /dev/tty59
/dev/tty16 /dev/tty27  /dev/tty38  /dev/tty49  /dev/tty6
/dev/tty17 /dev/tty28  /dev/tty39  /dev/tty5  /dev/tty60
/dev/tty18 /dev/tty29  /dev/tty4  /dev/tty50  /dev/tty61
pi@raspberry: ~ $

```

Now let's look at the command to install bib. As you can see here, we have different ensembles. We need to sort this library but let's say we need to update the library at first so we get a cash date you can see it update the slab. After that, we will use this sentence to install bib which we need to install. That is the area of communication as you can see every time you face a problem, you must find a solution. Don't just use this problem and finish the lesson or say I can do it you have to find a machine to look for, someone, to help you look for the onideskok updating. You need fewer minutes on things than when you come back here. But let's see what else we need to do. And we need to connect what we know with our enthusiasm by using unspeakable and carrying out this order allows us or list DTI devices to register all or to get a list of USP devices or Sierre connection devices then find online with Dev Ttya C zero or something like Dev. Why is he on one cheek for my sc, I don't know about Z in one or two that you can see? We have many and many devices. Here we have the only other problem. As you can see so we have to edit our code. I am one of them I 0 so open the bison again and change that I see. Same as serial. Ty se 0 to see someone 9600 to number SE you have found so if in your case, you get zero then the line must look at the same as Dev DTT so I zero 9600. But in our case we have one so we need to save it in one. This is a very important step you have to do so you allow communication between us all by if you are not 0 1 it will be narcanism communication because the Arduino board we have seen the Arduino board

## THE BREADBOARD



Now, this board is bread where you plug in things to talk about the plug here you can see in the image below you can use this customized cable type or you can use ordinary wire and just strip the tip. They are cables with a little belt plugin. You can call isolation from ordinary copper cables. But this is easy because they come pre-cut and pre-stripped. Also, they are collared so it's easy to distinguish them. Now I have shown you two things. Let me take a moment for a short time to expose the bread to you. Looks complicated with letters and numbers but quite simple. A breadboard is just a short piece.



You don't need it but it will allow you to test the circuit without having to unite it. Soldering If you don't know when you tie a certain type of metal around something to hold it together to be used at the end of the cable, the board makes a specific electrical connection between the hole so the wire doesn't have to be soldered. The only trick we need to know is where the connection is located. But I will show you and explain to you everything. As you can see in this picture so I will explain every location on the board. There are four areas that need to be clarified the area number as you can see. We have one area to three and four regions for areas 1 and 2 other blocks in the connected area. It's all blogs and Nedia is connected with it. Does this provide an area for you to control power flow? Which circuit will have at least one wire attached to it. To block the area on and one wire installed to block the area to assume them as the rhetoric side that flows from one and through it through the device to the other end without the finished circuit. Nothing will happen and from all on one floor. For the recording area, one with a plus is known as the Y cathode that it is also a minus sign known as. You don't need to remember it. I just told you. Area 3 here. Inside there are three R5 from blows that are all connected. For example, if you block the wire in it to the air block at the left end and the other flows to the earmuffs at the right end of the two cables it will have a connection and you

plug it anywhere on the board it will not be connected to the cable there. 12 inside. In your theory, each plug can have or connect yourself and there is no other place. Area for here. If you connect these two transport with our wire. The connection will extend to the line opposite. Why is there on the left and right side of the same number sharing board. They are not connected. If you want them to share connections, you need to bridge the gap with a wire. Once again this is only a short decrease chain that causes cables to their respective locations. If you want to enter Arduino or build some nobed sort you don't want to use Beadboard but instead, you just want SOLEDAD all shared parts. However, bread blades are great teaching tools.

## IF A CONDITIONAL STATEMENT

We will talk about one of the most important things in programming which is a conditional statement about conditional statements, it can be a simple statement if.

Reference > Language > Structure > Control structure > If

if...else

[Control Structure]

### Description

The **if** statement checks for a condition and executes the proceeding statement or set of statements if the condition is 'true'.

### Syntax

```
if (condition)
{
    //statement(s)
}
```

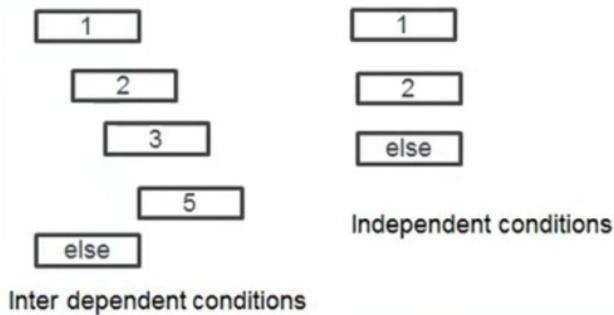
### Parameters

First of all, it will check his condition is a boolean expression. This means that the conditions stated in two brackets if the condition is satisfied, then only in it will go to a statement if the condition is not

satisfied. It won't go into the loop or in another case, it will go to Sloop. Good. He accused C and N. First of all, let's learn, okay the statement of conditions in programming is used to determine a condition where the condition is a boolean expression, it can only be activated when the conditions stated and state that, of course, recognize if conditions are different Then it won't happen, when going out of the loop and the maximum time it will remain on the loop. If the statement can stand alone it means that it does not need an L statement, while the statement, if the L statement can be possible but individual statements are not possible because they need a statement if they. Good. But the most important thing about this condition is that it is the possibility to be a hierarchy.

#### If statement

The if statement can be used in a branched manner Such as when the condition 1 is fulfilled ,it will take the other condition into consideration and fulfil the required condition 2 if the parameters are satisfied



You can say that in the case of the thigh first of all the first condition is satisfied then it will go to the second condition. If the second condition is satisfied, then it will run the statement. I have never been in the thigh statement the third condition is returned it will go to the third condition and your thigh can see that it will go to the most included conditions that are placed in the hierarchy. But if the first condition is or not satisfied point, the program will jump directly to the statement while the condition can be placed regardless of each other without a variety you can say that if five will run where to go to any

condition besides five or six then it will go to the condition is. Okay, now let's look at a simple example of what conditions and how they don't exist. Okay, go to Arduino. Create a new document. First of all, we will write about our LED PIN connected to pin nine.

```

File Edit Sketch Tools Help
Fade
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Fade
// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(9, OUTPUT);
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(10);
  Serial.println(brightness);
}

Done uploading.

Sketch uses 2428 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 192 bytes (9%) of dynamic memory, leaving 1856 bytes for local variables. Maximum is 2048

```

```

File Edit Sketch Tools Help
sketch_dec27b | Arduino 1.8.4
sketch_dec27b.g
void setup() {
  pinMode(9, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int x = 1;
  if(x == 2) {
    digitalWrite(9,HIGH);
    delay(2000);
  }
}

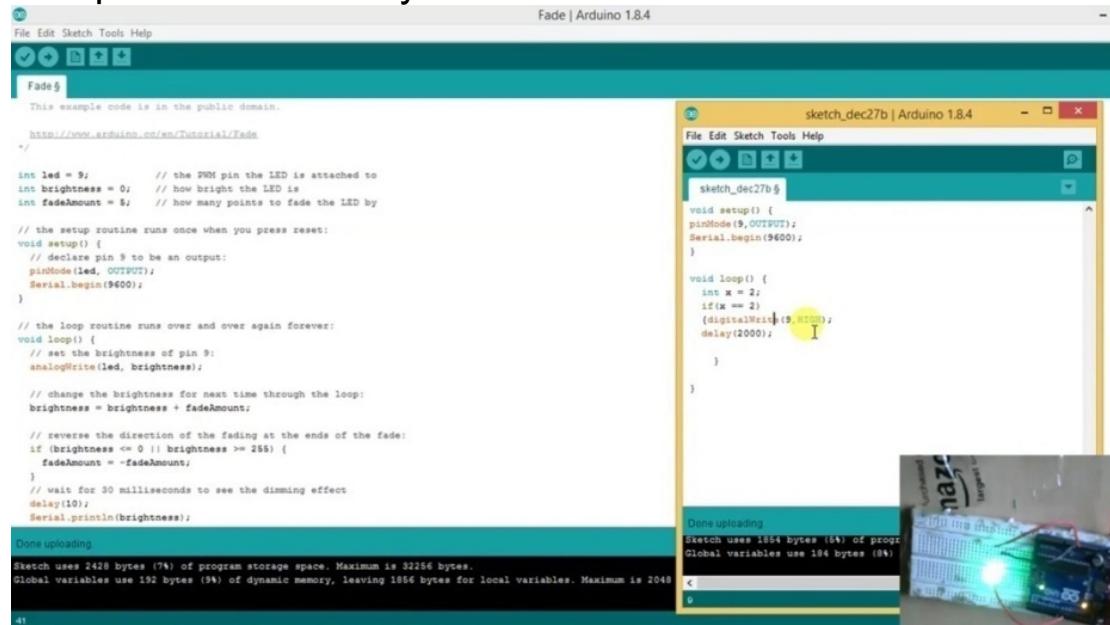
```

Compiling sketch.

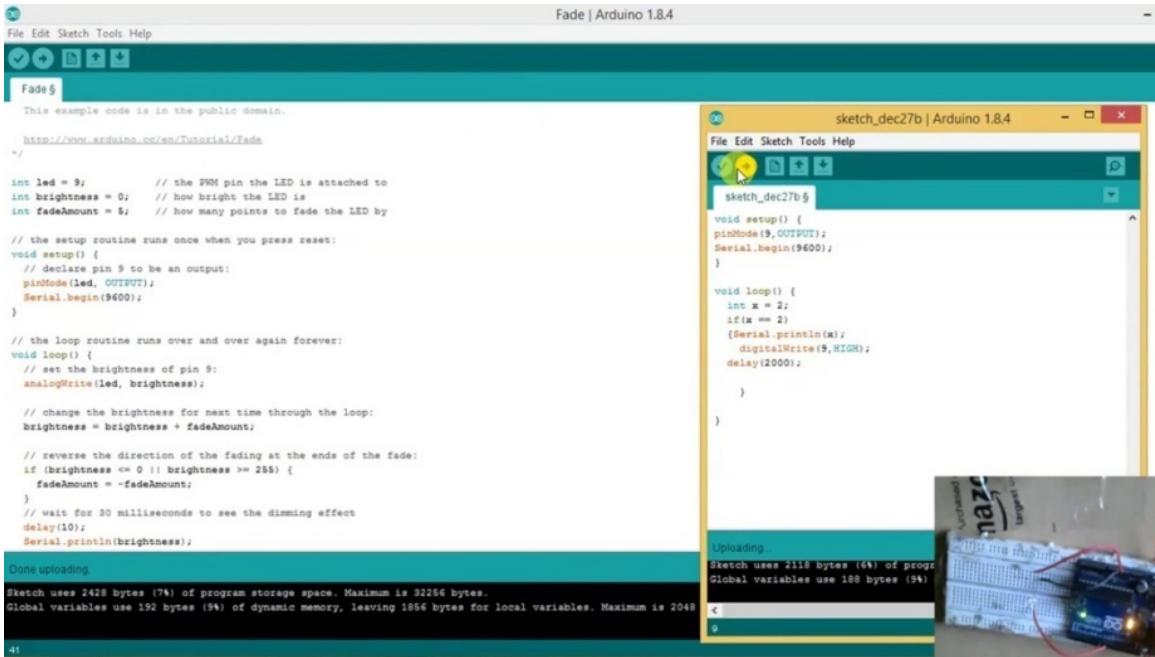
Okay, first of all, it's indeed the thighs of the Modemurder PIN and the death of PIN nine as the output or UT U P Pin nine has been declared as an output okay but we want to see the conditions happen more or better with the vision of our condition declare the serial also we have declared inner cereals giving it cancelled from 9600 okay, what is done now goes to the loop void loop we are okay in the void loop first for a condition we need an integer x okay. Now go to the condition you can make a simple statement with conditions like if x is equal to two then the thighs are conditions and there have been stated. Go to two curly brackets and write something digital w writer digital writer nine number nine high okay in the program, first of all, we declared a nine pin as the output that we call a series monitor after that integer and provide conditions in a statement if it says Round x is equal to two then only it will go to a statement that says that the nine digital right is equal to the height and you can say that it will be a good thing will run for about two seconds. Good. But if the loop doesn't come out, it will come back and it will run continuously. Okay, let's just take it. We here will value one. Good.

Upload first compile the sketch and it will be a few seconds uploaded you can see that if x is equal to one well that has a syntactic error I have to give it to the same person now, let's see we make a small mistake and I will explain about it, First of all, if you want to say if it is the same as you must always give to the same value with science if you have given it the same as sending a task that says that x told the value given on the right side that means that x save the value of two, But if we give A equal to the sign saying that X compares its value to the right integer that has given anything else does not blink on any car function. Now you only give an integer value to run it.

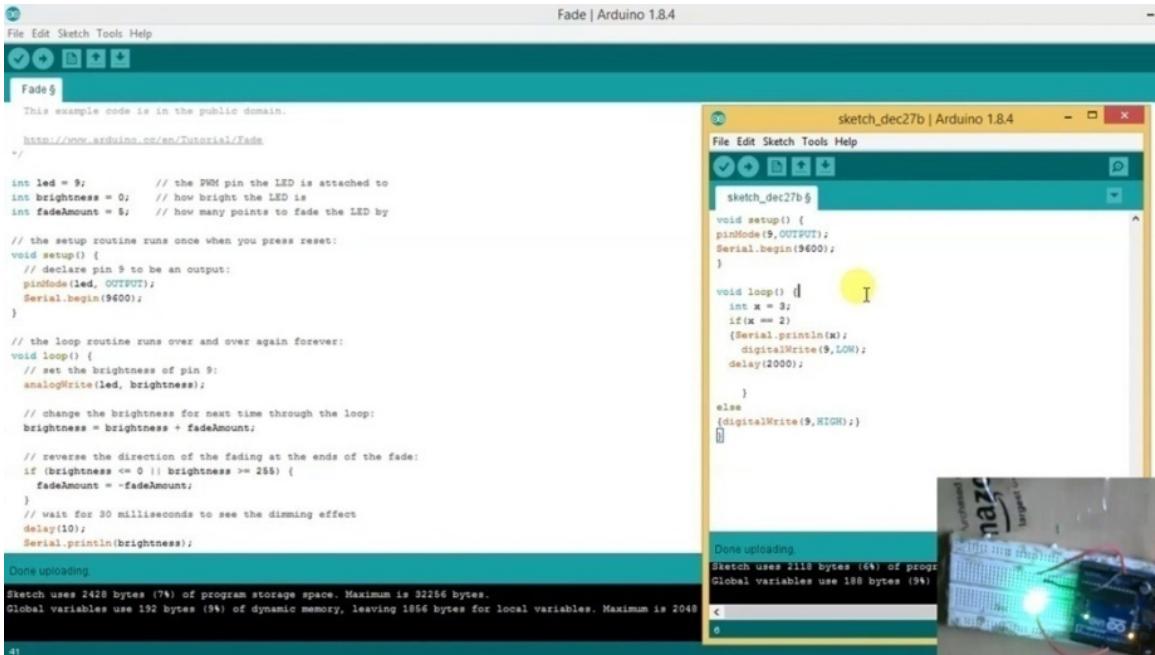
Compilation and then you can see that the LED will shine fine.



But if you want to see the situation how can we get a clear declaration without giving it to a country? Go here in conditions and read the serial point conditions, okay. Dot TP RI NT LN is important to write it to the next line after the first line is written, we give X. Okay, let's post and upload it to the world.

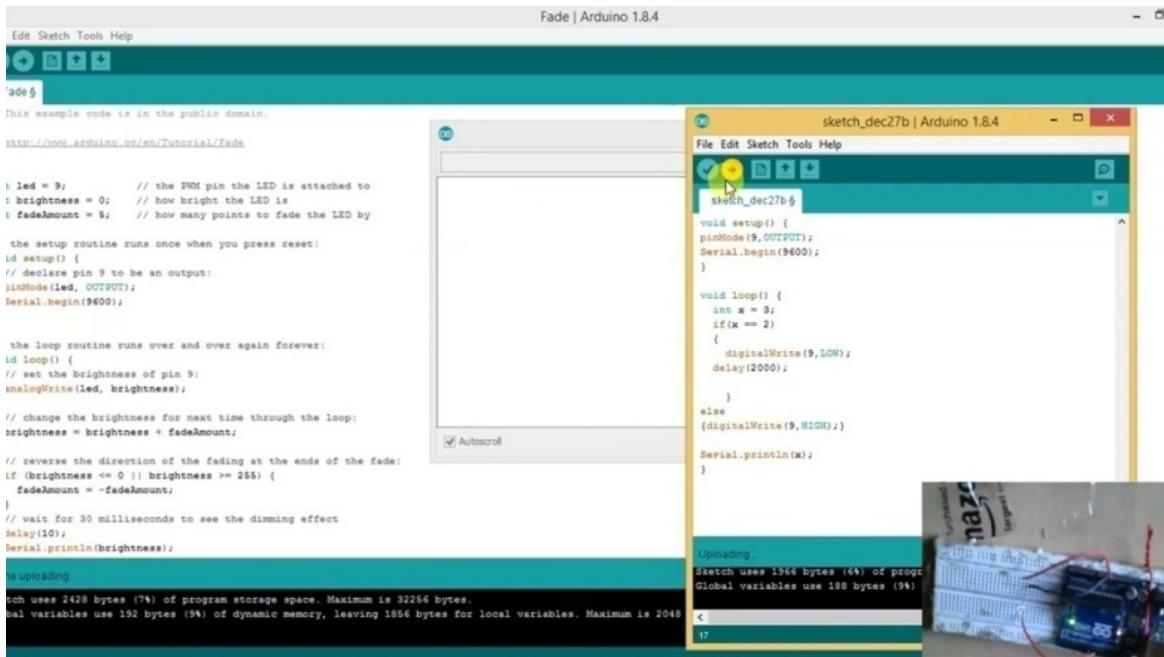


We uploaded it and Ledger CNC then monitored the X value equal to two. Okay, now, about the L. statement you can simply write e l sp without braces or anything that goes down writing two brackets. digital writer. Nine is equal to let's go there and do exponents.

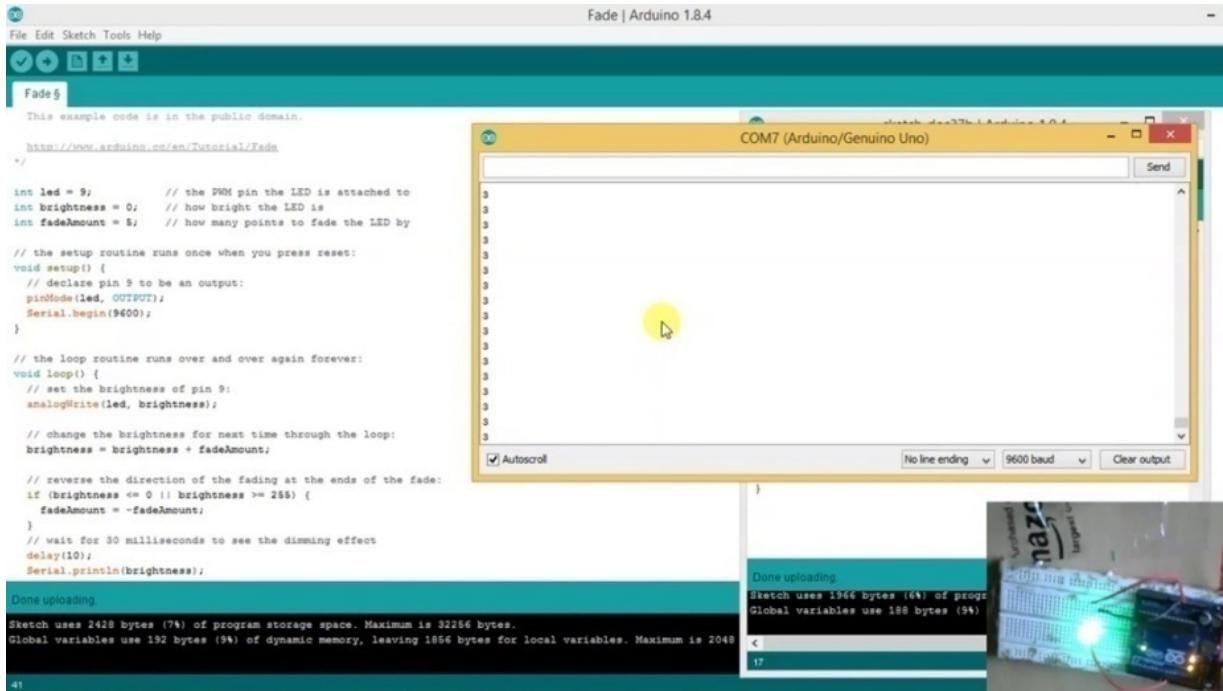


We get it high thigh highs will be high and if the value of x will be equal to two then the thighs will be low or w and we will change the value x equal to three. Now in the thigh program, we can see that we

have stated the value of x equal to three, if the value of x is equal to two, then it will enter the loop that pins nine must below but it is not possible here because we have set the value x equal to three so that it will immediately jump to another statement that says that the light will be high or will shine fine, let's compile and upload we will take a minute now you can see that the LED shines and if we want to see the value of x, you can go to The serial monitor and check here the value x is equal to okay there is an error that I declare that I have stated the value in it if the statement so that the serial print will not occur because in the statement and the statement itself does not occur. Okay, go here.

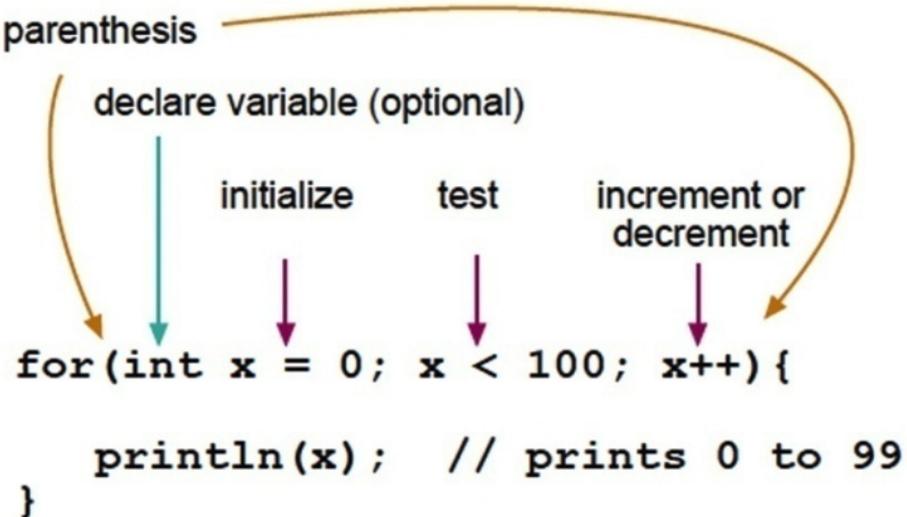


After a good statement after being stopped, write the X value will always print the evaluator condition that has been fulfilled or not. Okay, let's check it given the value x equal to three. Now if you say there is a value x equal to two, then it will enter the IF statement.



Okay, let's just check you can see that it will take it taking a few seconds to let's give a restart, okay you can see that the light doesn't shine and the value x is equal to two. So it has entered the problem. It has entered the loop if and in the flow, we have stated the statement that the light will not blow.

## FOR LOOP



Well for loops is an important compound of programming that has been used in almost all types of programming for loops used to run specific code types. For loops, it is used to run code lines for a certain period or to take certain intervals or take certain objects. After that, it must come out the condition out okay, let's try writing to loop at RD owner not to get an example of his physical computing. First of all, for loops is used to write a statement for time political intervals then after you have to get out. There are two methods out, first of all, default without writing anything else. But the second important method is called a break from the loop we can solve the loop anytime we need. We will discuss both now. Okay, let's just, first of all, open Arduino, open a new sketch and ask for the thighs method.

File Edit Sketch Tools Help

sketch\_dec27c

```
int x = 0;
void setup() {
  // put your setup code here, to run once:
pinMode(9,HIGH);
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
for(x = 0;}
```

We have introduced the LED on pin nine. So we will do our work about it. Okay, let's write some code. Pin mod. M must be capitalized because of the small-temperature. More LEDs Nine. High thighs are in parentheses correctly. Okay, and open the serial port. For a better understanding of this situation. CDL starts 9600 Baud level sets, okay, now comes here, thighs are part where the loop plays a fine role for thighs is the word to declare the forum, giving two parentheses. And let's look at the syntax, first of all, if we want a variable and if we want it to get a fixed value and it must open everyone, we must be open to every situation, we must always create a global variable.



```
File Edit Sketch Tools Help
sketch_dec27c:1: error: expected identifier or '(' before 'I'
I
int x = 0;
void setup() {
  // put your setup code here, to run once:
pinMode(9,HIGH);
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:

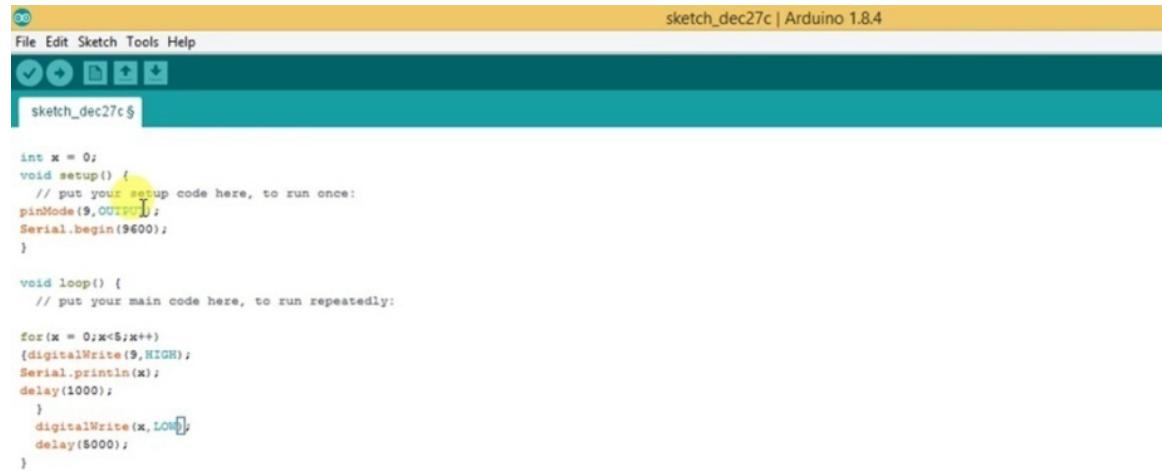
for(x = 0;x<5;x++){
  digitalWrite(x,HIGH);
  Serial.println(x);
  delay(1000);
}
}
```

If we write it in the word loop, it will be a local variable, but if we write it before the settings and before everything, just stating x equal to zero now has become a global variable and the value is taken only for the first time. After that, in each value conditions will be changed as for loops, okay, in the low world, we have written for loops and we have given parentheses but now allowing us to do work.

Extraordinary. Call to zero, give a comma point to say that the statement in a thigh condition is given x equal to zero, x less than, say five. Now x must be increased. Thighs are writing methods for loops. First of all, declare it gives two balance notes, writes the V one variable value, x is given a zero value. After a semicolon, we have given X smaller than five, as we understand, add Xor to one

step, as we want we can also write x equal to x plus one, but it is simpler to write x plus four one increase Time get two curly braces, open and write ID in L Digital Writer. Dispose of breath, coma. Height means to loop when x is smaller than five, it will give you a digital right pin value. Having high Edmonds and light will shine de la Vida de la must be introduced to a better program experience. And it's important in most cases to get better perceptions about what we write and what the program is because the speed of micro gondola processing is very high, but we want to control it according to our work. Okay about the words we want to get on the serial monitor. State it is L Serialdot Printer using always print LN to get a line and make it function in the next line. The right thigh serial dot printer

And we print the x value on the screen. Okay, it's cool. Four X is equal to zero x larger x. Four X is equal to zero x smaller than five x will increase every time before reaching five.



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch\_dec27c | Arduino 1.8.4
- Menu Bar:** File Edit Sketch Tools Help
- Tool Buttons:** Includes a magnifying glass, a gear, a square, a triangle, and a checkmark.
- Sketch Name:** sketch\_dec27c
- Code Area:**

```
int x = 0;
void setup() {
    // put your setup code here, to run once:
    pinMode(9, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:

    for(x = 0;x<5;x++)
    {digitalWrite(9,HIGH);
    Serial.println(x);
    delay(1000);
    }
    digitalWrite(x,LOW);
    delay(5000);
}
```

Okay here now what should be after the loop after ending the loop we are given a digital writer value on the low but process will be very fast because the microcontroller speed is very fast so we have to delay for the next five seconds, let's say the light will die for the next five seconds giving a value of 5000 here and stop the line okay. We have stated in x the same zero void setup that we call the nine led

equal to the nine led pinaud equal to all output, place everything should be the capital we call serial start 9600 okay in our word log has made x is equal to zero x less Of the five x must increase by each step and add each step. All return statements will run okay, the Light pin nine will be high for the next five seconds and the value will be printed on the screen. But after getting it high on x is equal to greater than or equal to the same five, it will stop the loop it will come out of the loop and go to the next statement that shows that the pin nine must below for the next five seconds okay, let's compilation only.



The image shows the Arduino IDE interface with a sketch named "sketch\_dec27c". The code is as follows:

```

int x = 0;
void setup() {
    // put your setup code here, to run once:
    pinMode(9, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:

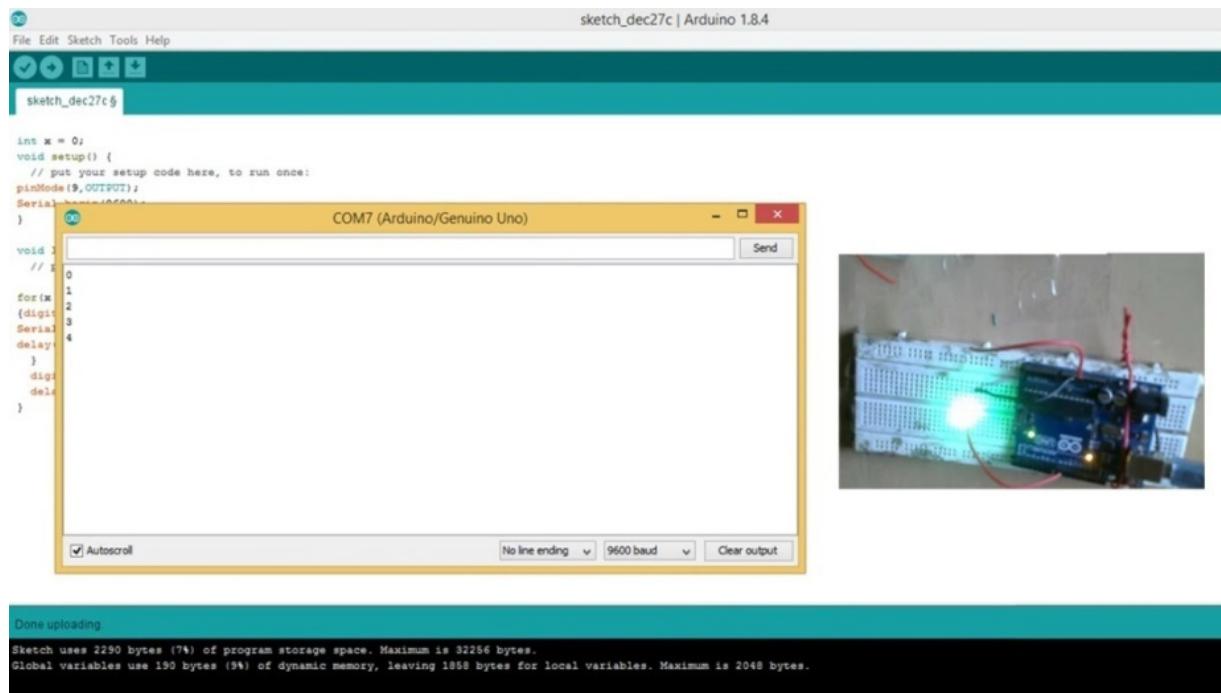
    for(x = 0;x<5;x++)
    {digitalWrite(9,HIGH);
    Serial.println(x);
    delay(1000);
    }
    digitalWrite(9,LOW);
    delay(5000);
}

```

To the right of the code is a photograph of an Arduino Uno connected to a breadboard. A green LED is mounted on the breadboard, connected to digital pin 9 via a resistor. The Arduino board has several jumper wires attached to its pins.

Done uploading.  
Sketch uses 2290 bytes (7%) of program storage space. Maximum is 32256 bytes.  
Global variables use 190 bytes (9%) of dynamic memory, leaving 1868 bytes for local variables. Maximum is 2048 bytes.

Take time, no problem. No problem. Okay, let's just upload into the world. And uploaded done. Okay, let's look at the serial monitor better for a better understanding. The serial monitor is open at the back, okay. Let's get it now. The high light for the next five seconds and after entering the fiveit will below for the next five seconds. Again it will be high and again it will be below. Think of the brighter export of better experience than four low we can only introduce values known as rest. If we want to rest before five, let's have three.



So we can write equal to three do the right curly braces Break B R A K Break is a method for getting out of the loop. We have written it on the loop,



See that in the loop we have written x if x is equal to three, then it will immediately jump out of them for a circle. Okay, let's upload it to the world. For loops, I save the code. Okay, it's uploaded on the board.

Let's see what happens it's uploaded and uploaded correctly open the serial port, you can see that on the three it will come out of the loop okay you see that after the three of them, it just comes out of the loop because of the break we have included. This is a very important statement for writing programs and getting performance as far as hours in a little gain performance according to our needs and the thighs are about the loop mostly used to call the array value because it is very important to get the specified reading list for each value at each step but The best method for calling an array is for us to discuss more arrays.

## WHILE LOOP

### while() loop

```
Condition  
|  
while [digitalRead(2) == HIGH] {  
    Serial.println(x);  
}
```

Loop conditions are same as “if conditions”

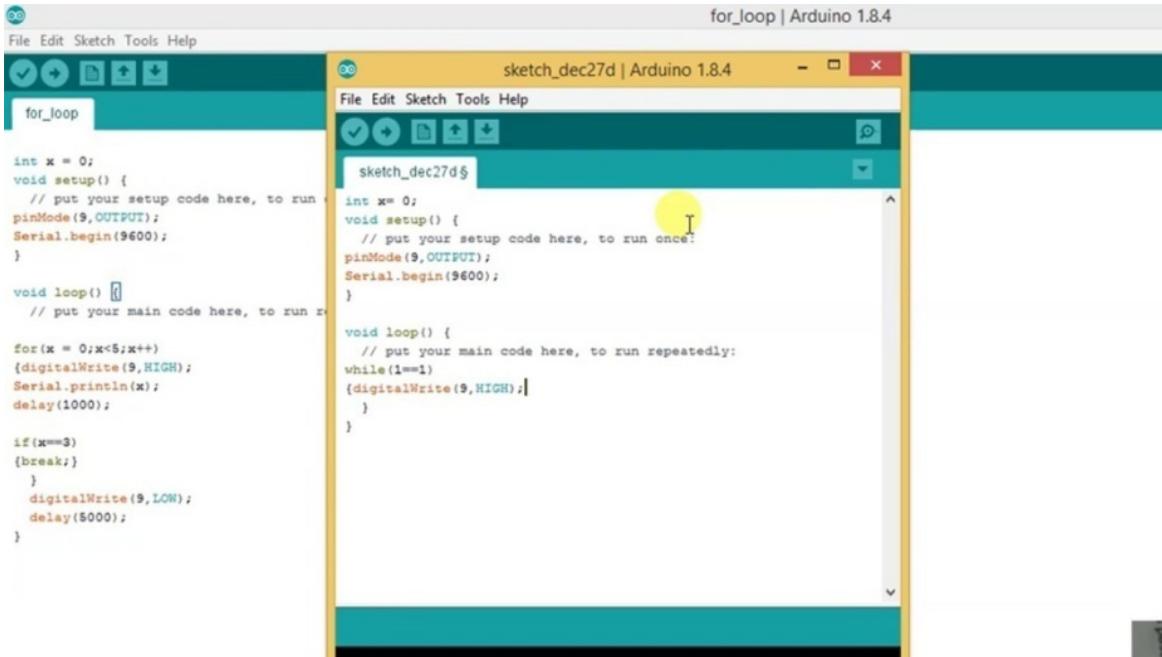
While loops well while loops are similar to loops as we have learned before, but there are some basic differences between for loops and times in while the loop does not require an increase, during the syntax loop says that it requires an increase and four uses a statement Complexes even you can use Boolean statements to run temporary loops while thighs don't allow for loops. Another important

thing for loops is a very good example of calling an array while the loop should not be shorter for such an experience. The basic difference between four loops and temporary loops is that special loops are specifically simple and controlled loops while one while the loop is individual and can run at an infinite time and can also use the Boolean value as a condition that can require it. It can also use the Boolean value as a condition needed to get an okay statement, let's do an example of the current loop.

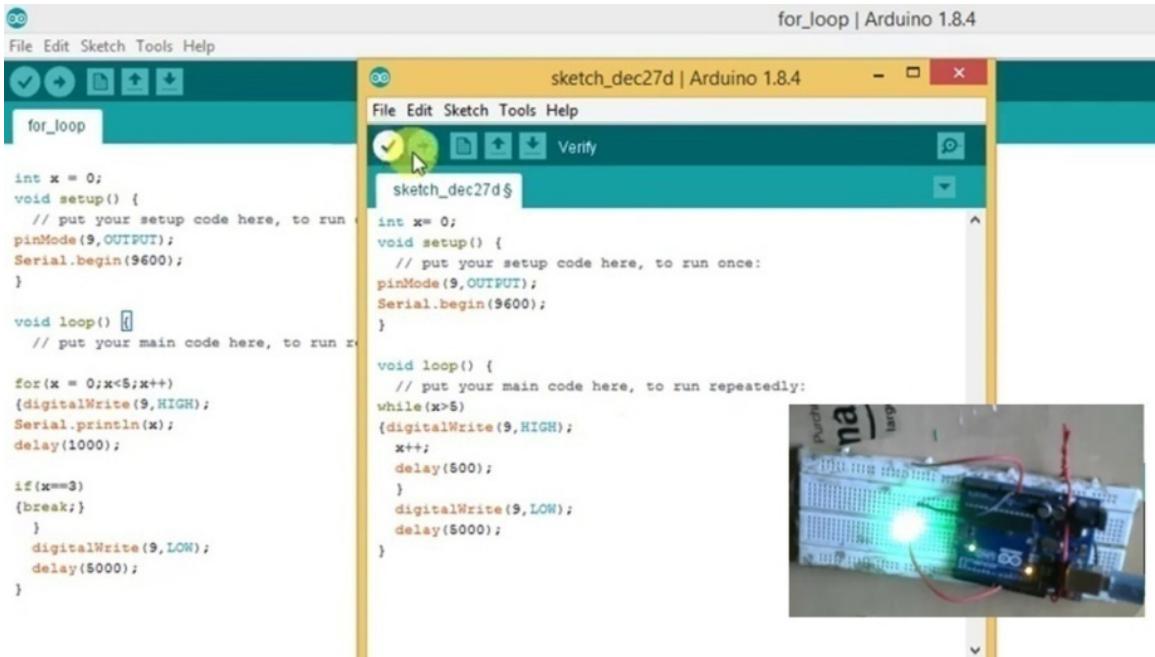
The screenshot shows a web page titled "Difference Between For & While Loops". The URL is "cs-fundamentals.com/tech-interview/c/difference-between-for-and-while-loop.php". The page content discusses the differences between the two loop types, noting that for loops are more compact and keep control statements together. It then compares their syntax:

for	while
Syntax of for loop: <pre>for(initialization; condition; iteration) {     // body of the loop }</pre>	Syntax of while loop: <pre>while(condition) {     // body of the loop }</pre>
The <i>initialization</i> is an assignment statement that is used to set the loop control variable. The <i>condition</i> is a relational expression that determines	The <i>condition</i> may be any expression, and true is any non-zero value. The loop iterates while the condition is true. When the condition becomes false, program control passes to the line of

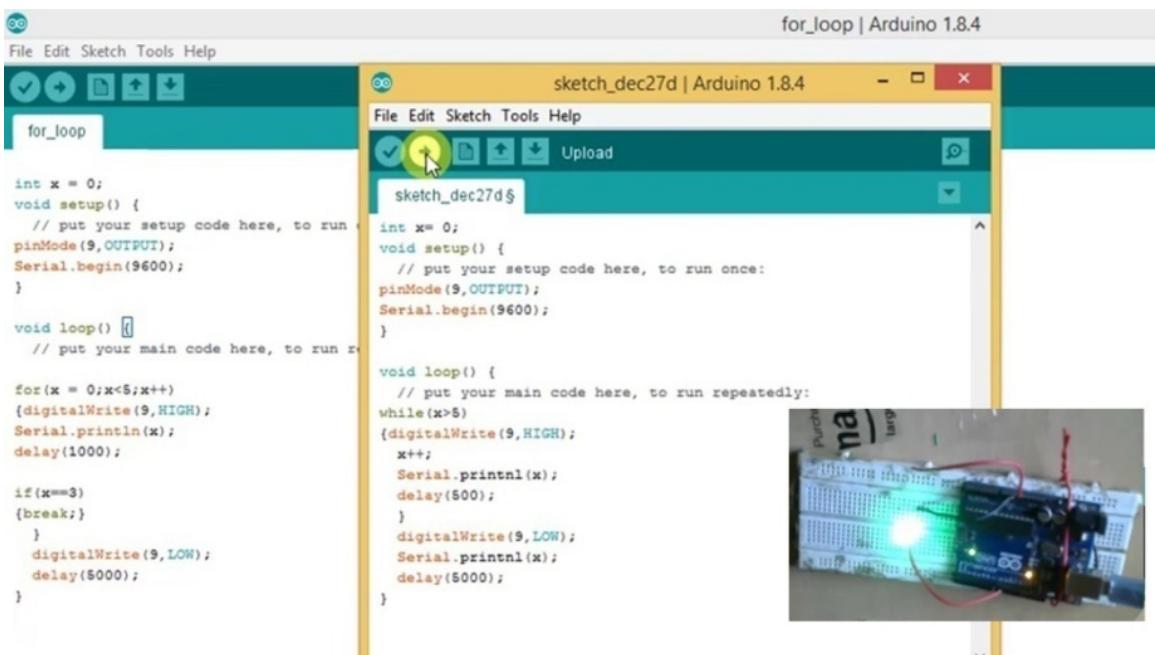
If you want to learn the basics of temporary loops and how they can be used for different examples of programming that I show on websites that will give you a fundamental difference between them. Now you can see that here it returns it for a while the loop time conditions and for loop conditions are met. Good. Let's go back and open Arduino and let's write a simple code for violence. As usual, I have provided an integer value of X and it is a global integer because this is a global variable with entities as stated before, I provide a global signature variable so that it can be operated from all parts of all other programming and then just say that we have connected LED to pin nine okay.



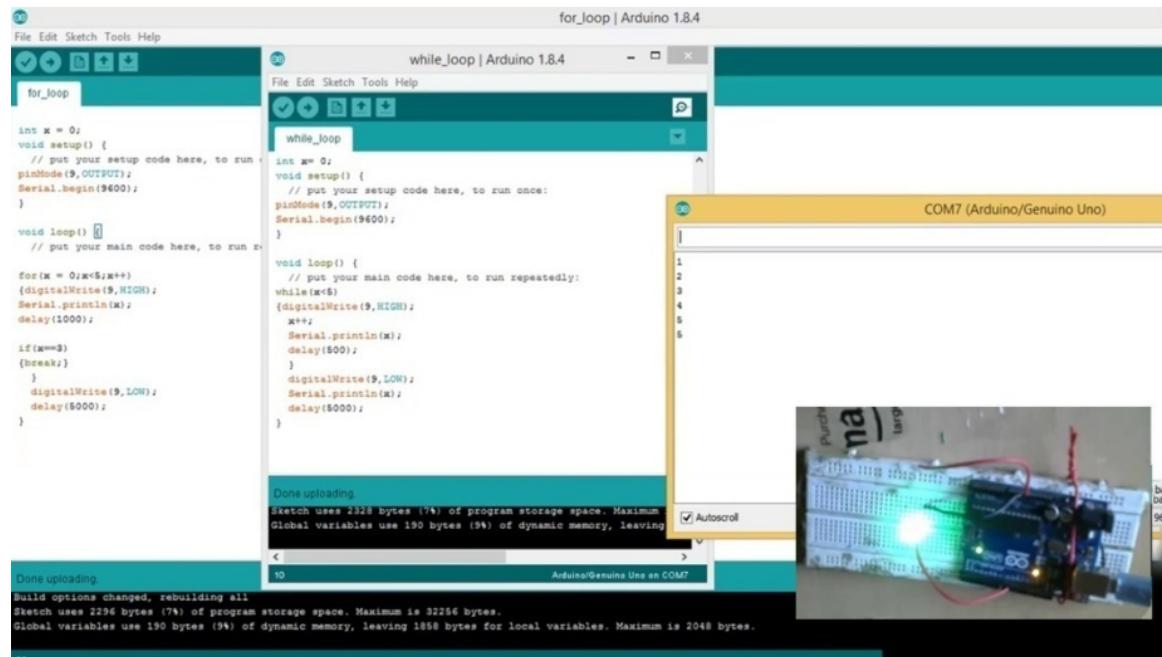
So, the Declaration of Model B Model B Modenecomma output must be capitalized as I said before and open the monitor series for a better understanding. If the word does not change the orange thigh means that you have made a simple mistake or something like capital or making small letters alphabet or uppercase or syntactic errors whatever is done then the word will not change to their default color for example, to call something, to call something which will turn into orange let's say that is fixed for all tests. We will sample other councils in college. Okay, now enter the loop while initializing simple calls while you can say one with one thigh is the most common example to call a temporary loop. This will make the LED shine for an unlimited time. Okay you can see that here if one is called that it is the basic knowledge of mathematics that is always true, it will run thighs is an example of a Boolean statement while the loop can bring it. Okay, let's upload into the upload world it's fine, you can see that the light will always shine because one is always the same as one and the thigh T is a known fact.



Now, in a few moments you can do other statements and while you can say that while x the same as x is always greater than five, the thigh is the best example. You can also add hikes with solid eggs such as EXO differences with every step while our loop has done the activities that are said at the time of the loop. Say that while x is smaller than five, digital, eight and nine equal to high let's give delays say 500 milliseconds okay.



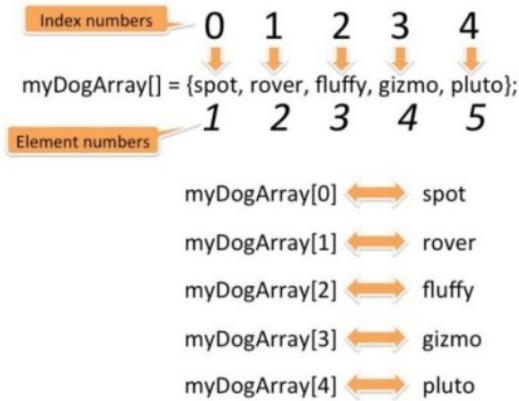
Now after exiting it must be given a statement or will return overall for the loop. Nine low commas will run for the next five seconds. To provide a description of 5000. In a temporary loop, if x is greater if x is greater than five it will come out of the temporary loop. Until then it will run locally adjusted continuously. We can also print it on the screen. So let's print on the screen, the LN serial series, it means the next line printing, every value, say it's very good. Now it's enough to copy all statements and you can also attach it here for a better understanding of this program. Good. Now, I have stated that there must be a syntactic error, you can see that the word does not turn into orange. The same mistake here let's delete and let's upload to the world. The ledger stores it as a name when the loop. It was uploaded and uploaded. Now you can say that in a digital serial monitor, I think there is a place in the code while x is the moeller that I say also that there is an error now change the syntax uploaded once more. While x is smaller than five, the light will shine after x is larger. After a while, x smaller than five light will grow after getting a value x bigger than five light with light does not work lightly will know for the thigh light is a simple syntax and we have done it with an increasing function.



You can see that the loop is running continuously after the temporary loop has been put in the next line, but you can see that the value of x

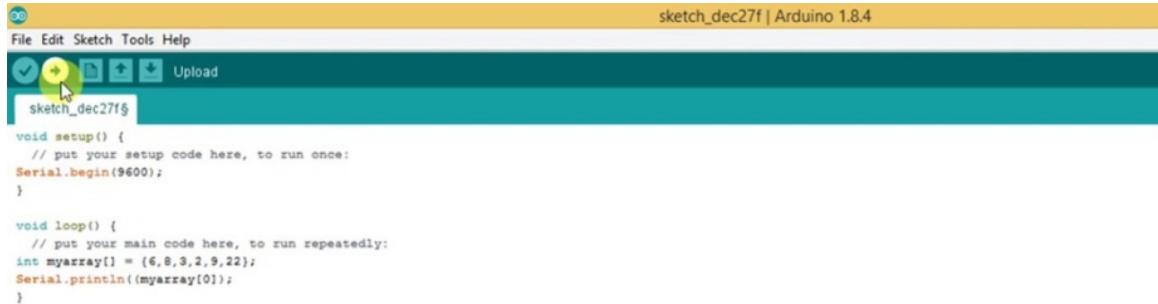
is when the X value has been announced and cannot be changed. So the function will know to allow the LED to shine once again. Thighs are examples of temporary loops and we have stated that the book is great considering the difference between them all for loops is used as a simple loop. For variable declarations, it has a much larger level of control over the loop. And it can also be both loops can do brake operations. Now, while the loop can also run on a Boolean statement like one with one of the physical tools available in the world like one equal to one, or one equals one or you can say that two are the same as for anything that can be a physical statement. And it can also run on the function of the increase method that you can see that we are here doing the same function as to loop using a temporary loop but there is a difference in the last value log y always less than one. In the case of thighs, you can see that when smaller than five. So last week you will read here are four while if you do the same function in the loop, the last value is we will read the same as five in the thigh in while the loop will look for info for five.

## USE AN ARRAYS



We will discuss the array, the Well array can be said as an array, an array is an integer that can store a large amount of value into it and they all have a specified place or the number specified to call them. The thigh method is used to store large amounts of value and call them when needed. We have discussed earlier for an array loop easily can be easily called with the help of loops. Now we will discuss the array and make a small program from it. Okay, first of all, open Arduino. Create a new document, Create a new Arduino open to make a new sketch now arrive here. First of all, we want to see the value. So we first will call the value of the Spri serial monitor n Sorry for this error. Dot serial starts 9600 with brackets alright. Now, let's get on topics or arrays can be called an integer array, my array is no difference used for brackets. These brackets are the thigh syntax used to say that the thighs are not integers but it is an array of giving the same value to now use the two curly-curses in this curly bracket, you can put integer values, I say six commas three point two point six again okay let's say nine coma22 okay see there are how many elements in array 6832 922 now listen carefully that the first value array is not my zero eras, but the value. Now listen carefully that in the thigh array we have around six elements. So you can't define the first value using my array method for the first value

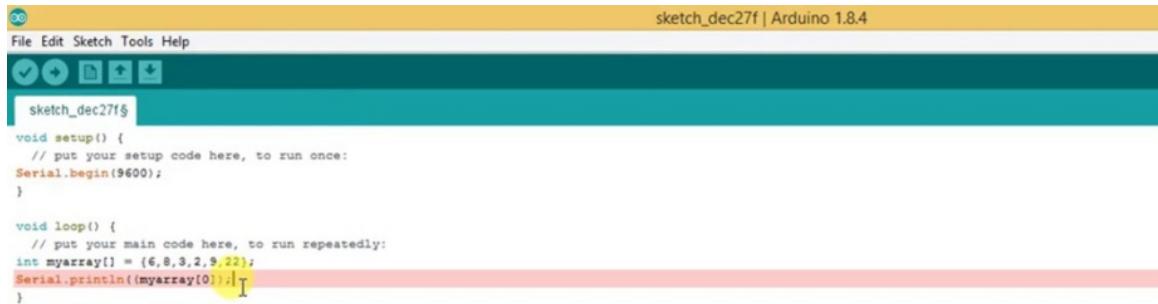
array starts with a value of 04 if you want to get a print six, you have to use my array value 046 Let's try it.



```
sketch_dec27f | Arduino 1.8.4
File Edit Sketch Tools Help
Upload
sketch_dec27f
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  int myarray[] = {6,8,3,2,9,22};
  Serial.println(myarray[0]);
}
```

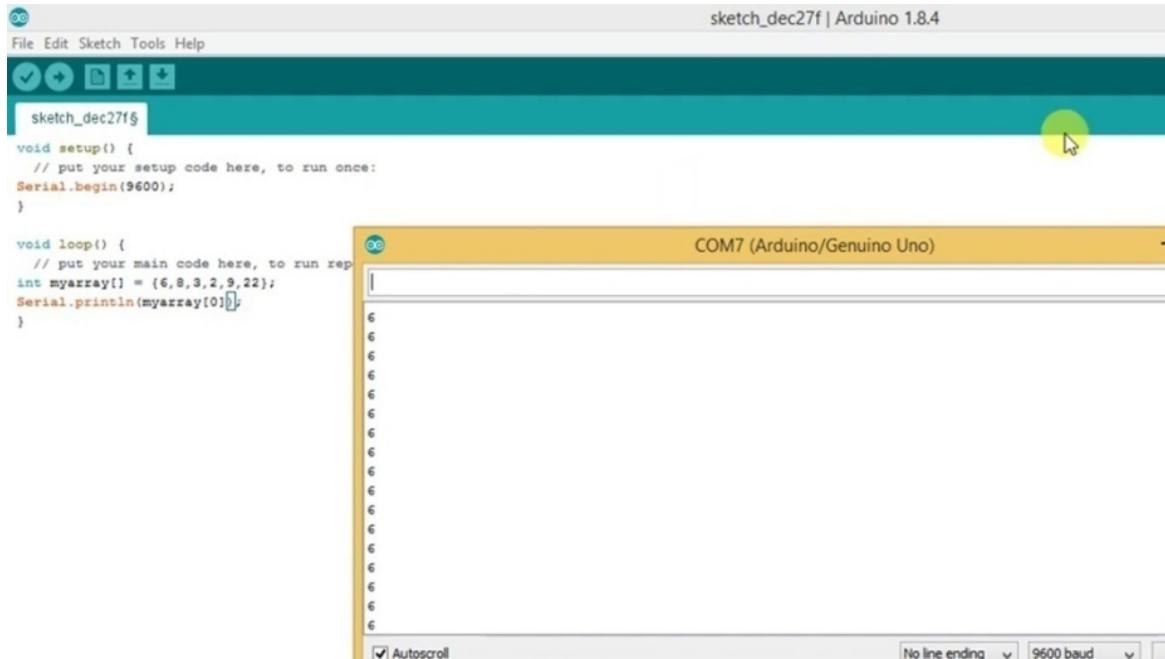
Tell Serial Dot Print LNNSAY arrays I use full syntax made into decades and say well using zero. Okay, stop the line uploaded to the board. Wait a minute. I just connected my Arduino. I upload it to the blackboard. And let's just upload. This shows that it gave me a mistake.



```
sketch_dec27f | Arduino 1.8.4
File Edit Sketch Tools Help
Upload
sketch_dec27f
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  int myarray[] = {6,8,3,2,9,22};
  Serial.println(myarray[0]); I
```

Come see. I played a series monitor from the thigh waiting for a second. I haven't closed the bracket. That's his mistake. Okay, let's see now. Just look at the upload compilation and wait a minute. There are some previous problems, let's just try it. Already managed to upload a zero inheritance. And it's six.



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The title bar says "sketch\_dec27f | Arduino 1.8.4". The code editor window contains the following sketch:

```
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
    int myarray[] = {6,8,3,2,9,22};
    Serial.println(myarray[0]);
}
```

The serial monitor window is titled "COM7 (Arduino/Genuino Uno)". It displays the value "6" repeated eight times, indicating that the array element at index 0 is being printed repeatedly. The monitor settings are set to "No line ending" and "9600 baud".

If you want to call the next value, just add it from zero to one. Now uploaded once again. You can see eight values that enter the screen. Thigh is because when we call an array, the array element I say that it is the second in the line.

```
sketch_dec27f | Arduino 1.8.4
File Edit Sketch Tools Help
sketch_dec27f§
void setup() {
    // put your setup code here, to run once:
Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
int myarray[] = {6,8,3,2,9,22};
Serial.println(myarray[1]);
}
```

COM7 (Arduino/Genuino Uno)

```
6
8
3
2
9
22
```

So, for one second you use one for the second third use and for the first you use zero thighs is a method for calling an array, but what about calling a complete array list? Yes, it's possible with loops as we have discussed before.

```
sketch_dec27f | Arduino 1.8.4
File Edit Sketch Tools Help
Verify
sketch_dec27f§
void setup() {
    // put your setup code here, to run once:
Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
int myarray[] = {6,8,3,2,9,22};
for(int x=0;x<7;x++){
Serial.println(myarray[x]);
delay(500);
}
}
```

Okay, let's just try it. For silver it might delete everything about anything else. Right for you two brackets, x equal to zero x not previously defined. So we have to declare it. x increment okay,

whether the syntax is correct. Let's do two Kurawal brackets. CJ Load Printer Print LN I call my Array Array Call to the method and what should be the value here the value must be placed here is x, but there is one problem because of the processor speed. We have to make it slow so we can see values clearly okay. Give a key delay of 500 milliseconds, stop the line, compile it. There is no compilation error.

The screenshot shows the Arduino IDE interface. The top bar says "sketch\_dec27f | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for save, upload, and other functions. The main code area contains:

```
sketch_dec27f§
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run rep
    int myarray[] = {6,8,3,2,9,22};
    for(int x= 0;x<7;x++)
        (Serial.println(myarray[x]));
    delay(500);
}
}
```

To the right, the Serial Monitor window is open, titled "COM7 (Arduino/Genuino Uno)". It displays the following output:

```
6
8
3
2
9
22
13312
6
8
3
2
```

So upload it to the board. We have managed to load the word. Let him do it once again. Okay, the program has been uploaded on the board. Let's look at the serial monitor. Starting once again for veterans to stand the first element 6832 922 each element is being called there is a systematic error that we have used the word seven so, it also takes seven which is impossible. So, it gives you trash when you enter the program correctly, it will lose six it won't give me a mistake okay give a value of six more uploaded and contact the series monitor now you can see that without the program error will run.

The screenshot shows the Arduino IDE interface. The top bar displays "sketch\_dec27f | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations. The main area shows the code for "sketch\_dec27f\$":

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run rep
  int myarray[] = {6,8,3,2,9,22};
  for(int x=0;x<6;x++)
  {Serial.println(myarray[x]);
  delay(500);
}
}
```

To the right, a serial monitor window titled "COM7 (Arduino/Genuino Uno)" shows the output of the code:

Output
6
8
3
2
9
22
6
8
3
2

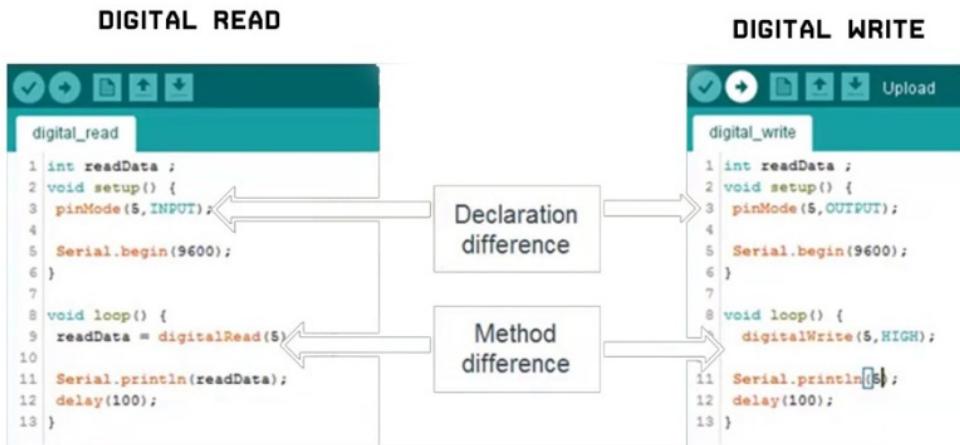
Okay, there are no rendering and array programs called calling all elements as needed. But what is the use of the array used in many cases to call some sensors or some you can say LED lights, for example, L is an Array of LED lights can be called the thigh method so that some lights can be obtained and die at certain times that match the loop or Whatever your condition, you can connect the roll completely and you can specify 563 pins randomly and you can call it with the help of arrays?

## CONTRAST BETWEEN INPUT AND OUTPUT

First of all the difference between input and output, thighs are the theoretical part. So we have to get, thighs are the theoretical part so we have to pass it. First of all, there are only two methods where Arduino Pins can gain access. The first is to provide digital input or the second is to give it output. Output and input can be digital or analog according to the PIN description. Most pins are able to provide output or analog input while each pin on the Arduino board that has a number capable of providing input and output in the form

of data such as the five analog pins will be converted into a digital pin using Dale Digital Pin 14 to Details PIN 19 OK, First -Anda, input in the pin given in the form has a voltage if we connect the data pin to read something and connected to the ground will be given a temporary zero value if we provide a voltage of five volts that are connected to the world in the form of soil, it will be given the same value applies to Analog while there is a difference between digital and analog whether there is analog you can get the measurement value. For example, if the voltage is three it will be given a value of 600. For five voltage, it will be given a value of something around 488 40 which is similar to it will be a value range from zero to one zero three when reading and to write it will be a good way to Judging humans to say that zero to two days for that minus one that is equal to zero to 255 will be the value of writing reading, while the input value to read is zero to 1023. Thighs are because the analog converter to digital configured in such a way in a readable microcontroller Tour to 10 values as input values. For eight that will be used will be given as output for the voltage source, but the scale for it is the same, this zero to five voltages is only the difference is that in reading you can get four times right as it is written alright.

Now let's discuss the code and what the difference is. Here you can see that both forms will have the same method for input or output as we first will discuss the details in what the code difference when writing input or output data. The two methods only have one difference between the mode for the input are written capital input, and for output, it is returned the capital output in the same method, but when it comes to programs in the world loop there are some differences in methods available, they are subject to but when it comes to the word display, there are some differences to call on the pin for digital directly and very simple just declaring a pin as the digital right gives a pin and



Set the value for digital output. Lower high declaration because this is a read function that does not require a pre-artificial value that will be read from the environment or from the open where it is connected. Okay, and with the same method, you can see the print value using the same serial print series to read digitally and the Dot Print LN series for digital writing methods.

## LED MULTIPLE

We can run it independently while controlling around three LCDs at the same time, okay first about LED there is a simple light diode that is connected to Arduino when a detailed pin is high, they are transfix when the pain is low. , they will not blow thighs okay is the thigh muscle theory Okay is the LED theory. Now what we will do if we want to control a large number of LEDs say that they are connected to five pins, nine and three, but we cannot control it randomly. If there is no method if we give if we will write the program for it will cause a lot of light to the program will be unstable. So there is a method using an array as we have learned before. If we use an array to control the list you can control the list or array or you can control the list or Array Array Array along with Arduino boards and some small programming methods that control LEDs with a thigh

array are important physical computing parts please concentrate. Let's start by opening Arduino. Okay, first in the setting you can see that I have connected the LED to a two pin, pin three and Pinfour, okay, this is the LED pin so, first, let me make it simpler okay we are in volt settings. First of all, we will call the pin mode of all the LEDs so we get to us to give them high values and we get the output we set it as output we set women as the output okay pin mode to give both. Give three and give four when the capital-output or TPU TP minus the error entered now is only copied and entered on three parts okay. The first function of the Void function has been completed. Now we will come to the main points, first of all, make my array give the name to the int LED given a bracket that is quite given to do two curly brackets to call arrays. Thighs are places where the array value is stored. Okay, we were given a two pin, three pin and four pin here okay stop lines. Now, it's only called a loop. Let me show you an example. Just say d digital WRI TE digital rightlvdsl e d s, give two parentheses and just say. So the three LED pins will shine. Okay, let's just take the program and we compile it there are some mistakes and the mistake is to say that let me improve the program.

```

multiple_ide_array | Arduino 1.8.4
File Edit Sketch Tools Help
multiple_ide_array
void setup() {
  // put your setup code here, to run once:
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
}

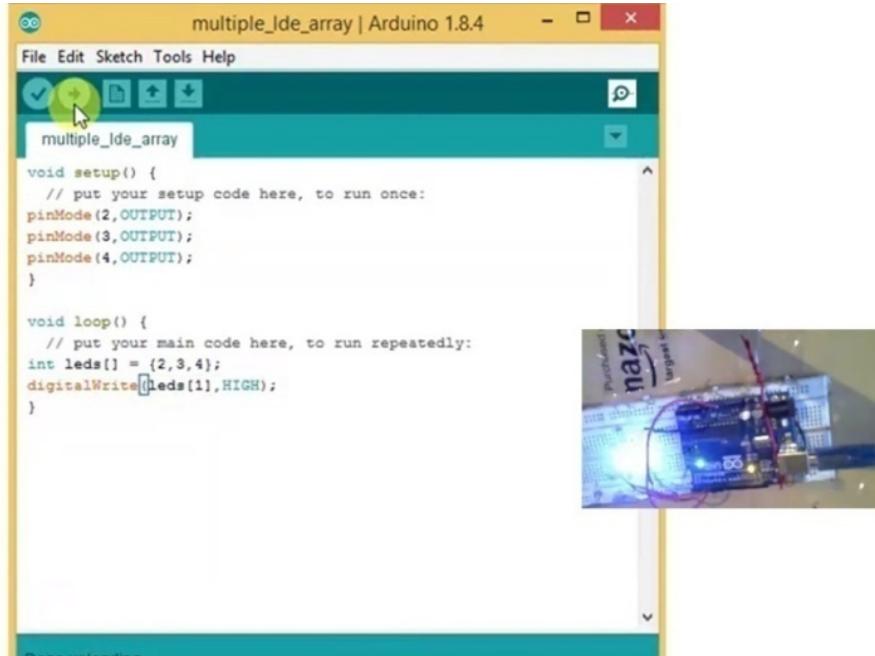
void loop() {
  // put your main code here, to run repeatedly:
  int leds [] = {2,3,4};
  digitalWrite(leds[2]);
}

```

too few arguments to function 'Void digitalWrite(uint8\_t, uint8\_t' Copy error messages  
too few arguments to function 'Void digitalWrite(uint8\_t, uint8\_t'

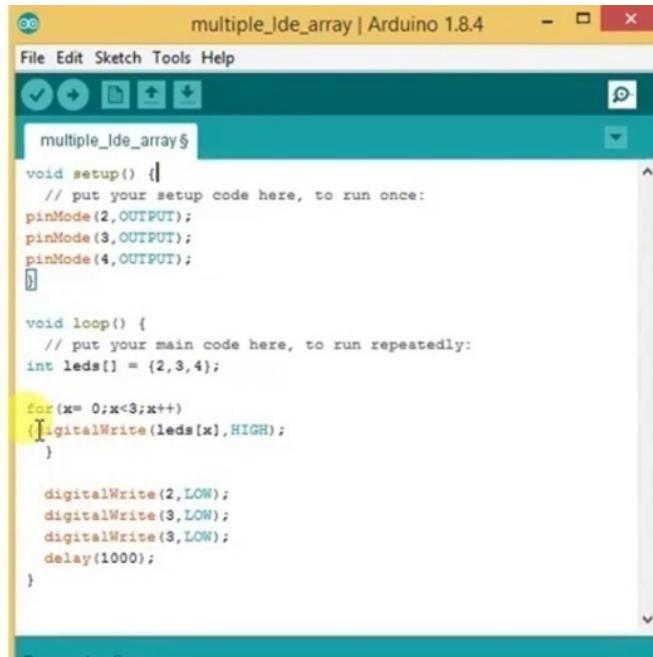
This is a simple mistake for dinner. Okay, let's just do it. Once again. Okay, everyone is my fault, I don't do a value that must be placed

low and give value now you can see that the number one LED will shine.



Let's just see. Okay, you can see that the LED connected to three pins will flow. But what about the pin four if you want to test it, you can do it. Just go there and say on the array through it means the third, the third, you are connected LED number four. Okay, let's just say. Okay, you can see that the fourth led shines. But what if you want to call them in sequence and give us a simple example and give a simple example of a program to work in a sustainable model? Okay, let's just try it. Remove the line. You can use for our loop. Whether parentheses, say x is zero, as we do to loop, give you a semicolon, say x six is smaller than, say three. X is smaller than three. Do brackets, x plus. Good? Now do two curly brackets. That is easy. Just write it. Good. Digital rights E D S Le Days do two parentheses. Be careful by writing the bracket that I have done a lot of mistakes here and the value we did was X that we provided X and the statement said that it had to end with a line. Now another thing is something you should pay attention to so that the LED is blown, they can't turn off you to mention that to turn off the LED okay, come here, copy the line after completing weaving, write three lines and there is in it our three lines say that the LED value Must be fine, let's do it. Get rid of a second just wait. And we have to return the height as low

because we want to stop we want to set it at a low speed for a while. Okay, let's do but for how much it will be done in a fraction of the milliseconds. So we have to delay saying the second we give a thousand milliseconds. Good.



The screenshot shows the Arduino IDE interface with the title bar "multiple\_ide\_array | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with various icons. The main area contains the following code:

```
multiple_ide_array.cpp
File Edit Sketch Tools Help
multiple_ide_array.cpp
void setup() {
    // put your setup code here, to run once:
    pinMode(2,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    int leds[] = {2,3,4};

    for(x= 0;x<3;x++)
    {
        digitalWrite(leds[x],HIGH);
    }

    digitalWrite(2,LOW);
    digitalWrite(3,LOW);
    digitalWrite(3,LOW);
    delay(1000);
}
```

Let me explain. So, first of all, we have called pin mode causing three-four output. Now in the blue world, we have declared an array of 234 which is the value we call here. For the next line, we have made a circle that says that LED X will be high to say for a moment, let me fix it. I will frown for a small version of Lola just need now, let's try it once again it is a mistake in the program I say led three times so let me fix it and now let's see 123 stop all women. So, if you want to see it in the serial port, it can also be done here just let me call the monitor series is E R I L Dot starting. We can use different initial values, such as 115200 but there is a higher baud level and we will discuss it later. In line, you can call it serial dot print In value x. Value says x. Now we don't use the value we use the x x value selected for the second okay, again

The screenshot shows the Arduino IDE interface with the sketch named "multiple\_Idc\_array". The code is as follows:

```
multiple_Idc_array | Arduino 1.8.4
File Edit Sketch Tools Help
multiple_Idc_array

void setup() {
    // put your setup code here, to run once:
    pinMode(2,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    Serial.begin(9600);
}

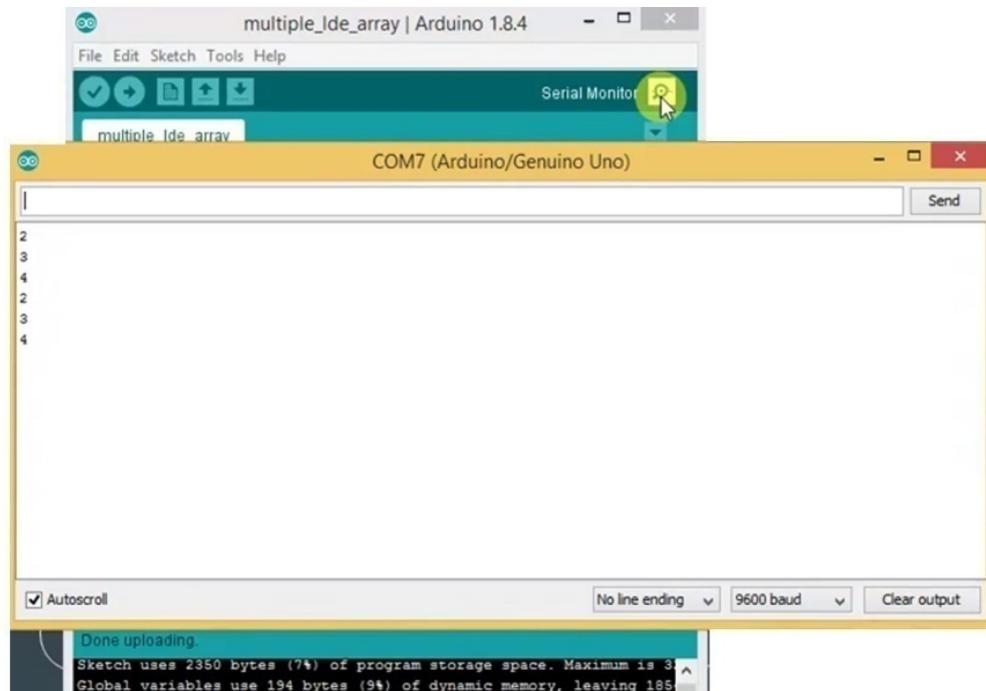
void loop() {
    // put your main code here, to run repeatedly:
    int leds[] = {2,3,4};

    for(int x= 0;x<3;x++)
    {digitalWrite(leds[x],HIGH);
    Serial.println(leds[x]);
    delay(1000);
    }

    digitalWrite(2,LOW);
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    delay(1000);
}

Compiling sketch...
```

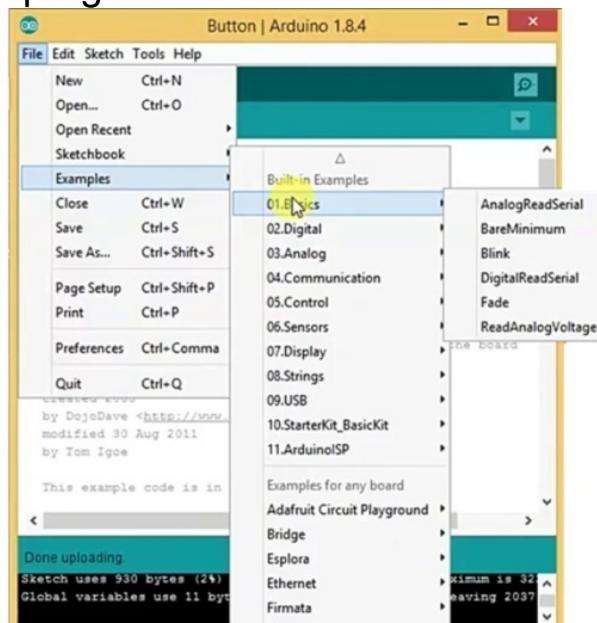
Uploaded to the board there are some corrections needed dot series starting there are two ends corrected uploaded to the board. And after uploading, the big book, open the serial monitor, the program will restart and SR 234 closes at 234 seconds and once again will be closed with the thigh method



We can connect and we can control the list of digital pins that are connected to other values randomly as possible in the form of a pin like 459 etc. It can also be used to control analog pins and it is very useful when it will contain a list of sensors and motors.

## BUTTON INPUT AND LED

We will discuss the input like a button and give it output like Lu Lu Lu every time we press the button to declare the LED height will shine every time we release the woman will return to the offset okay now, first I want to tell you that the thigh is the type of program Most fundamental planting in physical computing is just a simple circuit with the help of the button on the button you will get LED on and off okay. But what's important is that in the thighs in our thigh program using the logic of microcontrollers you can say that when the microcontroller sets one high pin it will do a logical operation, it will send a digital pin output connected to a high cause so the thighs are done by logical operations to lead to proud , Bright, okay. First of all, let's try writing the program.



This program is available in our Arduino example in the details of the parts mentioned as buttons but we want to make our program our

program, but we want to make our program so we did it from the beginning. Okay, open a new tab in Arduino. I write pin mode. In the case of thighs, we need two pins for input and one for output. Okay, let's just do it. Copy syntactically for a moment. We copy pin mode. Good. We have provided two-pin mode that says that input means that the signal is needed from it read the signal for high or low status okay and we choose three LED pins as output or pin where the lock led is collected.

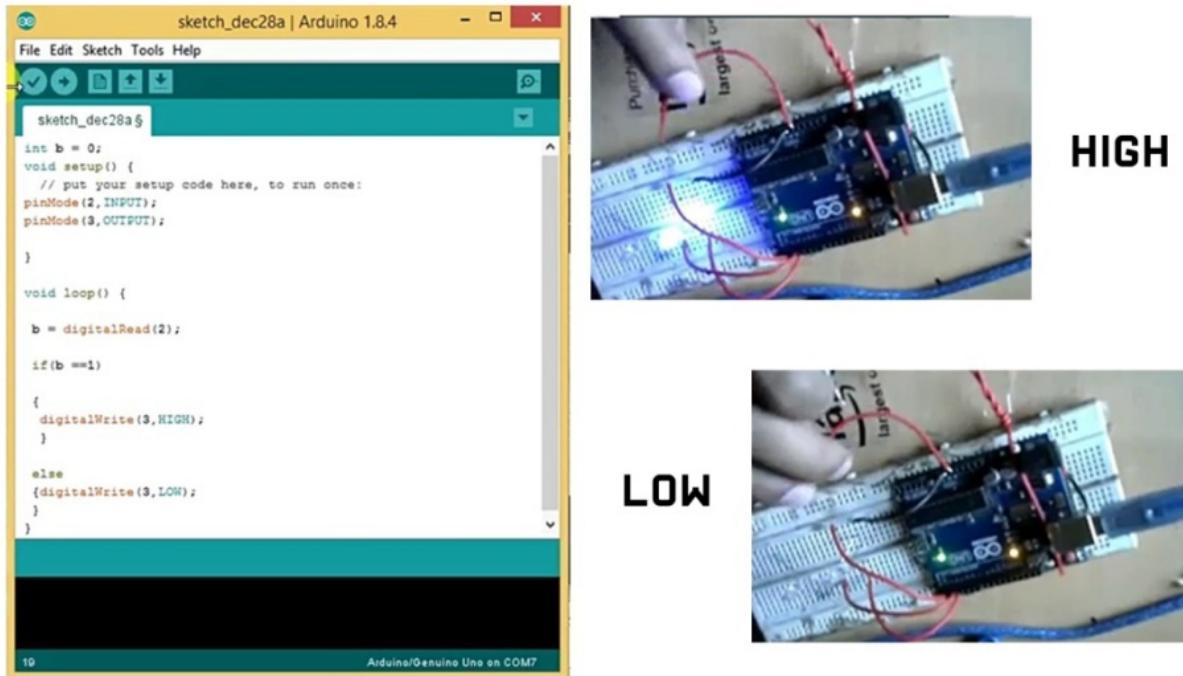


```
Button | Arduino 1.8.4
File Edit Sketch Tools Help
sketch_dec28a | Arduino 1.8.4
File Edit Sketch Tools Help
sketch_dec28a§
void setup() {
    // put your setup code here, to run once:
    pinMode(2,INPUT);
    pinMode(2,OUTPUT);
}
void loop() {
    int b;
    b = digitalRead(2);

    if(b ==1)
        {digitalWrite(3,HIGH);}
    else
        {digitalWrite(3,LOW);}
}
```

Now here let's declare integers saying int before what I am and okay we have stated in an integer named v. Let's give a simple example to read the details. We need V to store the reading value done when two digital pins are lower or low. So it is a simple writer in DIP L modal R e A. D digital does not turn into orange P number two letters in our case it's two pin okay, so the LED is a LED pin. So integers we will save digital print value into it. Okay and just writing a simple logic the same as OneWe can say it's right excavation digital P number three high high comacas we also need an assistant. At first, we have stated two pins like a two pin as input and we write pin three because the output of two pins will be used to read the value and three pins will be used to read the temporary value of three pins connected to the LED to the US and output. Okay and here we have asked anything for better to improve our practices always trying to

declare integers above okay word settings, we have made it here okay we and said our initial value from Zero Integer will be a full repair of the program and Improve okay performance, the first value of the integer we will zero but after entering a void loop or b, we will read the value of v number two as a digital number two runs high, we will be one and if the second pin runs low will zero. Now, for a statement if the condition is equal to one or we can say height, the right retail operation number three will be high or will blow the LED and then for assets when we have to give a statement that if the value is not the same as one then the light will die or you can say number three will be below. Okay, let's compile.



We compile and give the name we use on the okay button compilation just waiting for a while. Okay we upload it to the world and upload it well, let's just take it. Here you can see wire connected to five volts and thighs are two pins. Every time we connect the thigh wire it will change me but remember one thing because the garbage will use the LED in a high turn for a few more seconds that we must connect a high pin to the ground to get it. Look, you activate it. To change it low, you change it high, you change it low, it's better to use the cable on the button for testing because on the button you can get the value of garbage or you can say the micro value is initially lower

and lower. Transient who will provide high and low random values What will make the LED for thigh flicker caused by microcontroller processing speeds because in milliseconds so immunity removes wasin directly. Let's see once again the thigh because the value of the trash is now SR lights up and, on the terms of thighs okay is the use of the value of the button and the button will be very useful for many applications because in many cases there are no buttons but there will be sensors there is a digital sensor they will give high value or low and for systems that provide high or low value you need a simple logical program like returning here to get value and do retail operations and do operations such as the right details in high you can use other motorbikes or anything other than in our case , We have made it the simplest format like growing okay LED code will be served at the beginning of PBT.

## **INPUTS BUTTONS**

Arduino and what I did about what was the last time I would explain how to connect Washington to Arduino to control the requirements was a computer in Arduino's laptop microcontroller. You talk to the city and adapter. If your microcontroller doesn't have our vibart according to us via cable. But I will not switch minutes and need to be connected to replace the push button and the contact switch for a moment so we know is a straight step. I hope not what simple devices can cause the single-tip circuit button that is connected to us is usually a low voltage where 5 volts and Arduino are ideal.



Others are connected to your way, Dalman, when the switch is played pressed to activate the circuit. Is it open or closed? Digital bin only returns if there are 5 volts. Or is it over. Now in the following similar programs that can show you publish what is used to control tin. Is this we? First, we start by initializing the global variables, the event button is equal to seven semi-two in which leads from the Bush button are connected. Do you know how you use your number 7?



```
File Edit Sketch Tools Help
push_buttons

int buttonPin=7;
int ledPin=13;
int status=0;

void setup() {
  pinMode(ledPin,OUTPUT);

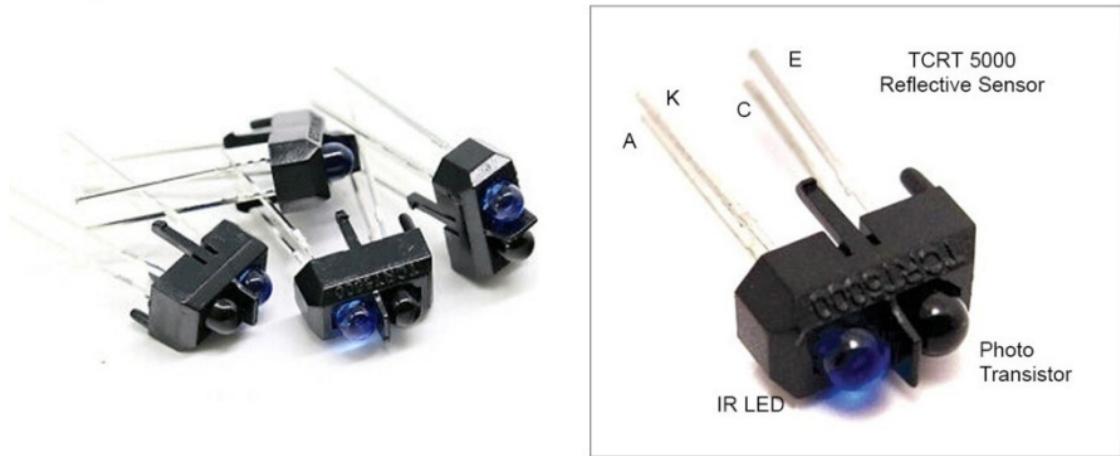
  pinMode(buttonPin, INPUT);
}

void loop() {
  status = digitalRead(buttonPin);
  digitalWrite(ledPin, status);

}
```

Finally short for integer. The fact that being in the bottom pan is used. Makes it easier to see that it's two words. Because the space cannot be used as well as the line must end with semi-coma two as shown and then equal to 13 semicolons where lead lead is connected. Here I have chosen the best status and tin equal to zero. Sénéchal at I givat the initial value of zero to release it under the emptiness. We write mode but don't realize that Ben has output rather than mod. Place the pan and purchased to be initialized. Published in Ben as unclear he and groove. We are this step. So he embed a Read button Chua Digital to McCollum used to read the initial pushbutton and to start status. Then the digital line is correct. Ben semi Colone status since yesterday. We both close high and if it is law then it will turn off depending on the status variable. And now you are good to go. Connect you are doing the board using a USV and unlucky cable. As I show my previous lesson. Thank you for watching this lesson. If you have questions, please ask. But you and I are connected to Unix.

## IR SENSOR INPUT ANALOG



I might be careful and I just don't censor at the front distance and so on it's useful to measure the distance without really touching the surface into three cables that stand out from a distance and not far evolved. In most cases, this soil signal is almost always encoded with black as a red base as a vault and white as a signal. If you are remotely, so don't come with me why you and I need to find it.

```

Renesas | Arduino 1.8.5
File Edit Sketch Tools Help
IR_SENSOR
Int x; i
void setup() {
    // put your setup code here, to run once:
Serial.begin(9600) // initialize the serial communication at bude rate of 9600
}

void loop() {
    // put your main code here, to run repeatedly:
x= analogRead(A0)
Serial.println(x);

}

```

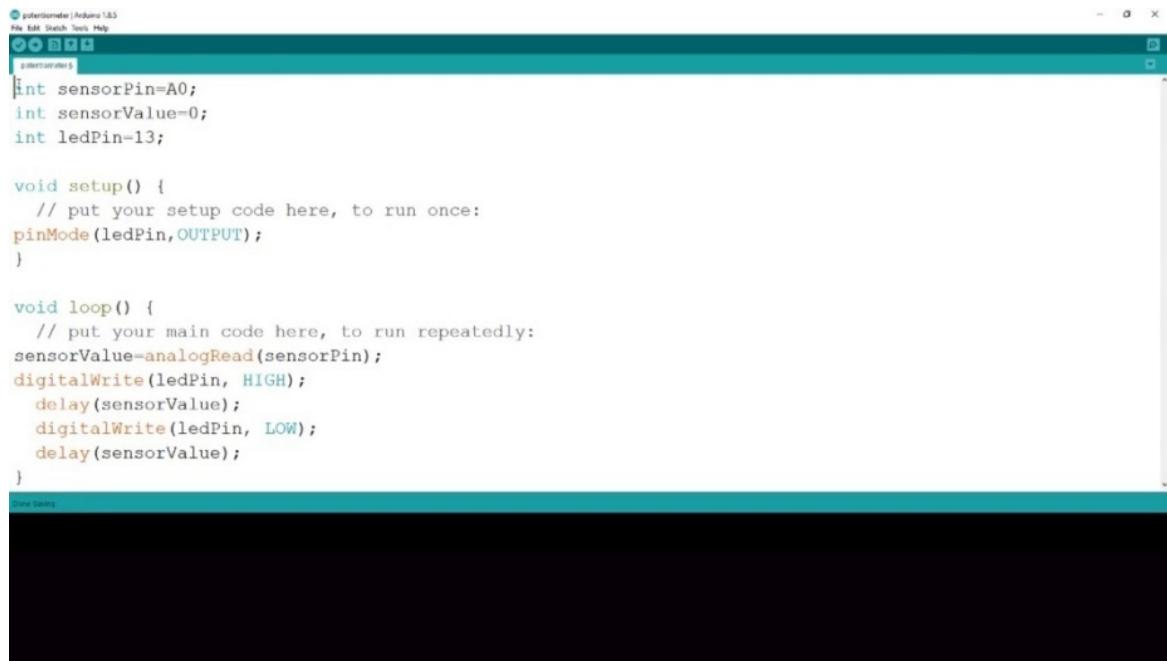
I could be a connector or so that the cable directly to the lead ensures the turn, soledad. No, do not contact one. So you can attach

a nick cable that it can evolve on what you know to connect the black cable to Arduino floor. I used what you know in this case, it's like Shawn on the picture. Because the sensor is connected to the game tray we know that the code will be like that. We started by initializing the initialization of the Semicolon variable that was initialized in an entire variable x and wrote it that we wrote c and started nine thousand six hundred semi-semitone initializing serial communication in world trade Nine hundred and six hundred. Under why we write X approved in parentheses the seven points points that must argue the data of the same connected store. The place as usual, we are right at the edge of the cliff. And among the support X, the COMA point which is the under-rental base on the screen is not different from it and pass to the thumbnail when you move the front of the distance and so on Solid objects. What values must change between 0 and 1000. 23.

## POTENTIOMETER INPUT ANALOG



You ask me to say the adapter if your microcontroller does not have you because we want everyone to be a cable showing me a few minutes, then it is the type of meta that this circuit will look like this. Connect the 5 Volt Tobin counter and their land indeed. Adakah who took Nick in the trash and the tension via an orientation turns two other bands. It's OK. Now, I will show you and explain all the lines of code needed to overcome this lesson, we will start by initializing the variable as defined.



The screenshot shows the Arduino IDE interface with the title "potentiometer | Arduino 1.8.5". The menu bar includes "File", "Edit", "Sketch", "Tools", "Help". The toolbar has icons for file operations. The main area displays the following Arduino sketch:

```
int sensorPin=A0;
int sensorValue=0;
int ledPin=13;

void setup() {
  // put your setup code here, to run once:
  pinMode(ledPin,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  sensorValue=analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
}
```

First, because I became easy semicon later and our people are chosen only to describe what value I want to attend that the type of human is zero where zero is not zero on Arduino itself . Zero is not the ordered term. However, when used in the context of the system recognizes it as zero. The line must end with a semicolon. As I said before declaring a variable in this tab, you can use the term in this case is a kind of pattern throughout the code. And instead because you will have two main advantages for this one. It makes the descriptive code. If you want to change the value of the variable, simply do it in the same place.

$$3/5 = x/1024$$

$$x = (3 * 1024) / 5$$

$$x \approx 614$$

And

as the value is equal to Zero Term of the semicolon, because the value is not reserved, a term term leads the same as Sénéchal. Once again, Liblin is not a word provided to Arduino. It has been chosen to describe the trash that is connected that the value of 13 is a normal value. But as zero when used in the context giving reasons to spend 13, but you will need the following code and other emptiness so that the output used to indicate that the lead has an output, finally, we read the void That will be repeated without time period because around you agree because of me, it is cynical on the lines of use of this term and I like to read the AM tension and I know the most used microcontrollers. And I see the digital conversion where the conversion of the voltage caused by a solid power the same number is equal to 1024, therefore, a zero voltage according to the zero value of zero five voltage depending on the added value of 1024. Therefore My value has evolved well according to the value of Omni that can be counted as this.

The screenshot shows the Arduino IDE interface. The title bar reads "Serial\_Port\_Potentiometer | Arduino 1.6.2". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar has icons for file operations like Open, Save, and Print. The code editor contains the following sketch:

```
int sensorPin=A0;
int sensorValue=0;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600) // initialize the serial communication at bude rate of 9600
}

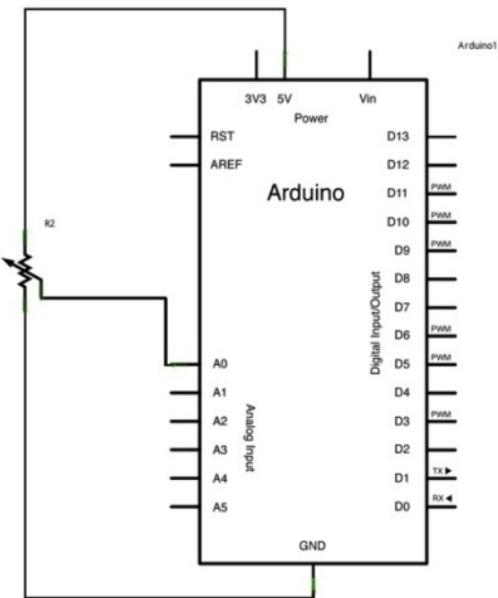
void loop() {
    // put your main code here, to run repeatedly:
    Serial.println(sensorValue);
}
```

The status bar at the bottom displays the message: "The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers and be less than 64 characters long." A blue selection bar highlights the number "64".

I wrote the equation. You can get the inside by making three divided by five equal X divided by 1024. We are X on the one hand. So we tried three multiplied by 1024 all divided by five. The answer will be about six hundred and fourteen. Then we die. Do you do it well? Hi. Which has been postponed and. Delays must take the delay time of the sensor value that stops the program for the number of seconds of this interval you are digital. Ben knows that the Ben destroys the delay has been taken from the value of the sensor? It starts with a program for millisecond sensor values. As I said after connecting the counter button. I bought this sketch on the board and I became a series of series when you read the cursor of the notebook or cursor. The volume must change between 0 2 1023. In the same way as the valid will flash with faster or shorter delay. You can now read values and use it with your account and then your function used here is and I do not agree. Where it was selected number 0 if you use it. Then, number five, you must modify the code to play and, given about 5 if the system does not check the syntax and make sure God Blonsky gets the next check of the potentiometer connection of the correct shooting and d. 'another. . If you buy very cheap on all this and that the meter has a chance that can be mechanically defective. You can test this using a mustmetimeter and extends to the model. I sit several meters to read Jesus standing. The resistance changes

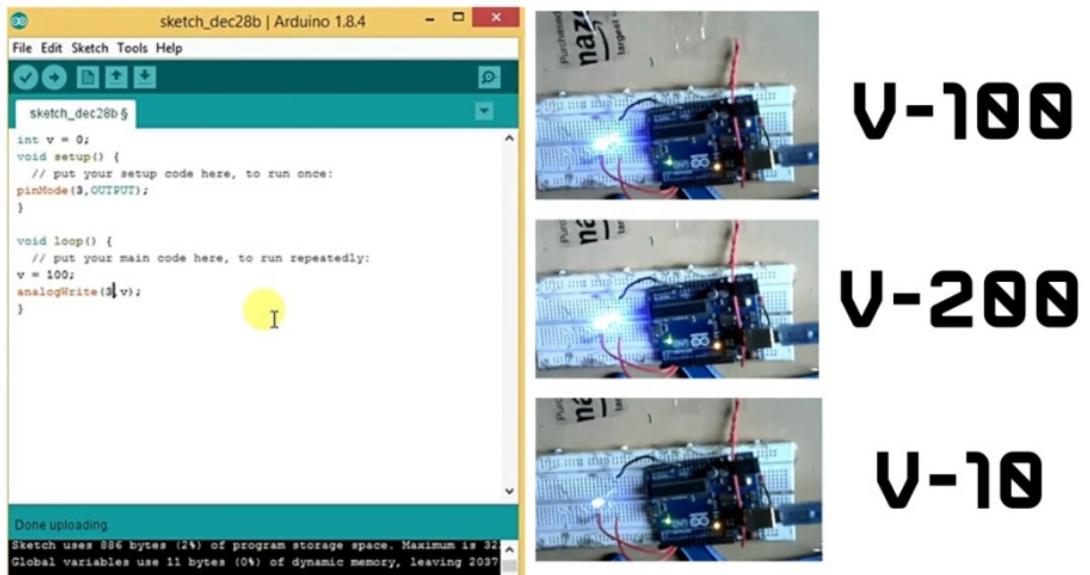
slowly but it works. It's just a fool fool. You need and you can not be wrong. Now, and if you want to see his own value. Look at the code I showed here and it's a new sketch before initializing the example of variable or adding in the same way that zero that chooses later, but for the Shimit button later in the voltage value equal to zero that is a variable for storing the value. which comes from the sensor. But some of your functions will be used now under vacuum. The voids arranged under the void tab that we wrote to see Seeing starts between parentheses 9600 semicolons initializing the CNN communication in Aldred or nine hundred six cents. Then, under the empty newspaper, we write C and D to print and in parentheses since the value rather than a semicolon. Because the value contained in the variable because the value of C via the United States becomes empty and digital is free at that time and I am not discouraged in Arduino. After finishing on the glass battle at the top right of this ridiculous window, I want to add a communication monitor to start sending and receiving by Arduino here. You must check the gold head and nine thousand six hundred. You will see the garbage. Thank you for watching this lesson. If you have any questions, please ask the queue and air. We see each other next lesson.

## **ANALOGUE INPUT WITH EXAMPLE**



The analog input is a method used with a potentiometer to show the voltage through the card and say that the word can display a voltage on the serial monitor, but we will learn here that you can manually write an analog value in the program and give American output on LED brightness, so if a value of 255 LLC will increase optimally, but if we receive a value of 150, the LED will increase up to 50%. And if you receive an analog value on the right 50, the LED will shine very deeply. Okay, just try with Elven's example. OK, first, open Arduino. Let's start by writing our sketch, we do not want only one value as a three-pin connected LED spindle. So we say the spindle three as exit. Remember one thing that must be very clear that the analog output can only be produced on a pin that has a special type of symbol on it. Well, you can see here that the pin connected to analog has a special type of symbol on it that is named as a PW, because an analog writing in Arduino is performed using the pulse width modulation . Okay, write the program. First, start with mode. We make PIN or PINTHREE mode as output number of the output output we want us to want an integer that we will record the value of the analog we say is an integer V equal to zero value initially, of Agreement, okay, no statement for your identical to less than 100 V of value given for integers as one hundred. Now, write in odiol capital with an analog right thigh will call the PIN code of the ANA three

connection and will give the value stored in the variable that we see the syntax indicating the integer that calls the output of the penis gives an entire value and write The analog format on the three pin with an integer value v.



let it simply download it on the board and see here. You can see the growth of the light, but if you wish if we give you values from 100 to 200, refer to the brightness of the brighter rear LED 100 and see where the LED LED is low, but if you Want to see the actual performance starts 10 See the value of light brightness is very low. So, if I affect our method can directly provide an output to the analog pin to perform the occurring operation, write the variables correctly, we modify the constructor task cycle, which is activated on the digital converter in analog, which is activated. Is a little in the three temporary for this reason, which is used in Arduino Uno will provide a task cycle that can be duly divided 255.

The screenshot shows the Arduino IDE interface. On the left, the code editor displays the following sketch:

```

int v = 0;
void setup() {
    // put your setup code here, to run once:
    pinMode(3,OUTPUT);
    Serial.begin(9600);
}

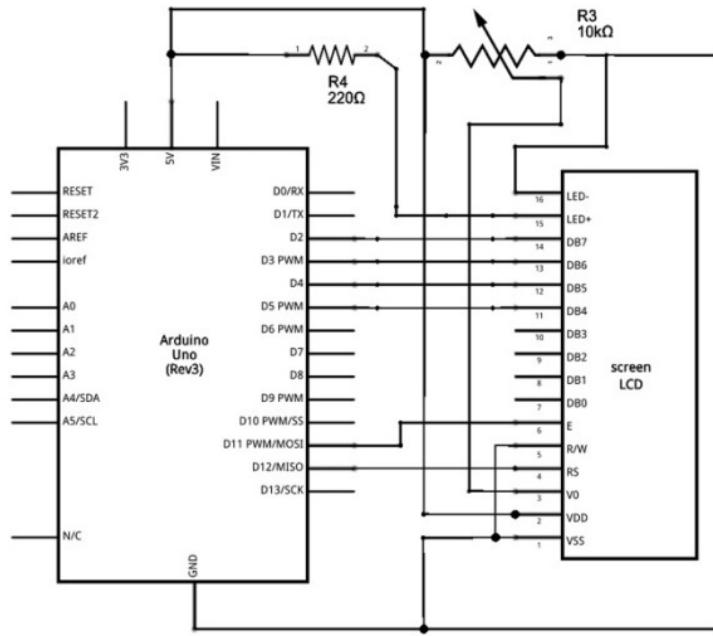
void loop() {
    // put your main code here, to run repeatedly:
    v = 255;
    analogWrite(3,v);
    Serial.println(v);
}

```

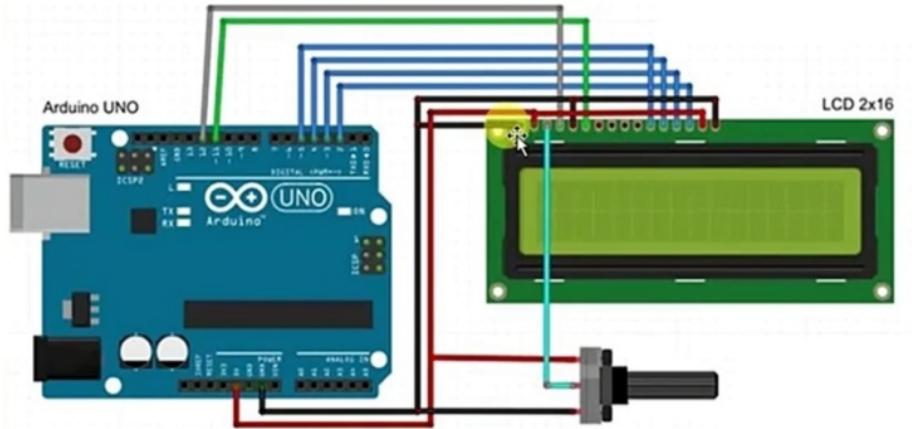
On the right, the Serial Monitor window titled "COM7 (Arduino/Genuino Uno)" shows the output of the sketch, which is the value 255 printed repeatedly. The monitor settings are set to "No line ending" and "9600 baud".

It will be a task cycle. Once again, Mega 255, they will be the highest point of truth. See the head will be correct. So, if we want to see how the value is explored or the number of people are returned in the interest, how many values are stored in the variable that you can see here say 9600 9600 here and down. Write a printed value of dot value. We store values in V then we are writing here change the download OK shiny, okay. See you later. The value is 255 donated by analog well now allowing you to save 25 women growing with a very benignant pace and an analog level value is 25. And if you want to find the task cycle, 255 simple divisions representing about 10% of the work cycle of the thigh. is a method for finding the task cycle and with the thigh method, we can control analog or you can say an analog output using simple programming.

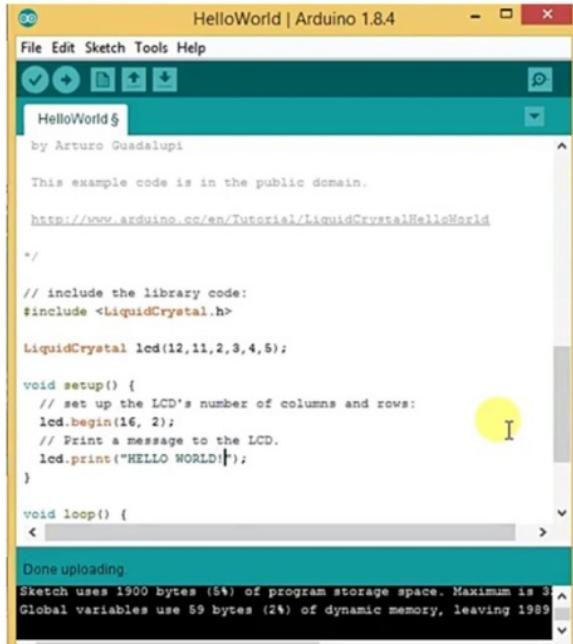
## LIQUID CRYSTAL DISPLAY



Well, the LCD screen is an important part of Arduino because it is the main thing that can give you a program view or you can say that the variables are in the standard Arduino, so it does not require computer, as in a series monitor every time. You want to give a screen, we need a computer when using LCD thighs here. You can get all views on the LCD screen so we can create Arduino and complete the system in the box and can place it anywhere without using a computer and we can see the variables it can Be used to measure temperature, moisture and you can make special molds on the LCD screen.



Okay, let's look at the LCD block diagram. Well, the LCD scheme remembers the important things of the thigh means for light of light thigh. Here, the thigh of the penis is important for the analog display. If you convert ads to rotate calls, this will give you a value that will make the LCD or Sun LCD. Okay, I will show you the same thing that you can see that with the rotation of the rotation of a potentiometer potentiometer, you can see that there is a display of LCD changes and which gives us a dark or ally appearance . Okay, here is this scheme, here are the four foundations that came to come here are four basic software used to convert the entire built-in program written in Arduino according to the LCD screen. Okay, let's start now writing a simple example. First of all, remember one thing in Arduino is a so-integrated LCD library and a simple LCD program is also written to Arduino.



The screenshot shows the Arduino IDE interface with the title bar "HelloWorld | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, refresh, and search. The main window displays the "HelloWorld" sketch by Arturo Guadalupi. The code is as follows:

```
/*
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/LiquidCrystalHelloWorld

  */

// include the library code:
#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11,2,3,4,5);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("HELLO WORLD!");
}

void loop() {
<
}

Done uploading.
Sketch uses 1900 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 59 bytes (2%) of dynamic memory, leaving 1989 bytes free.
```

We will see here the program. First of all, we need to call the library of liquid crystals so we would call the liquid crystal word. Edu gives close support now, we have to declare a liquid crystalline element. We declare the LCD liquid crystalline element and we provide a spindle like a good pin that allows 11-pin display and spindle 2345 for the variable to print on the screen. In the word game, we will call the series LCD. To start the text parameter, we will call the serial point function to start it set to 16 pins to do so that it means that 16 columns and two lines are enabled on the LCD screen and in the following line, we will print . Halo world. The thighs are words that we can print directly on the LCD screen, but simply by calling the word in parentheses in Word Lu.



The screenshot shows the Arduino IDE interface with the title bar "HelloWorld | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Open, Save, Upload, and others. The main window displays the code for the "HelloWorld" sketch:

```
// include the library code:  
#include <LiquidCrystal.h>  
  
LiquidCrystal lcd(12,11,2,3,4,5);  
  
void setup() {  
    // set up the LCD's number of columns and rows:  
    lcd.begin(16, 2);  
    // Print a message to the LCD.  
    lcd.print("HELLO WORLD!");  
}  
  
void loop() {  
    // set the cursor to column 0, line 1  
    // (note: line 1 is the second row, since counting begins with 0)  
    lcd.setCursor(0, 1);  
    // print the number of seconds since reset:  
    lcd.print(millis() / 1000);  
}  
  
Done uploading.  
Sketch uses 1900 bytes (5%) of program storage space. Maximum is 3  
Global variables use 59 bytes (2%) of dynamic memory, leaving 1989
```

We will define the cursor on the zero pin and the one that gives the bottom column here you can see the column of the thigh in the first part will get a point of variable and we will write here variable variables that are milliseconds and we Print, download it. In the world, these downloads are waiting for a moment and wait for a minute and you can see that Helloorld is written on the screen and calculates the time. Allow me to specify for you. Hello the world and counts the second time in a millisecond method with the thigh method, we can connect the LCD screen to Arduino and get the desired view itself. Okay if you have any questions, okay, but anything written something, we will write something acid here. And when we come, the author says, the author says something ah, that is, we will write value in the second row and the first word of agreement of the first part we did to download in the word. Let's see now where we can see the top written here.

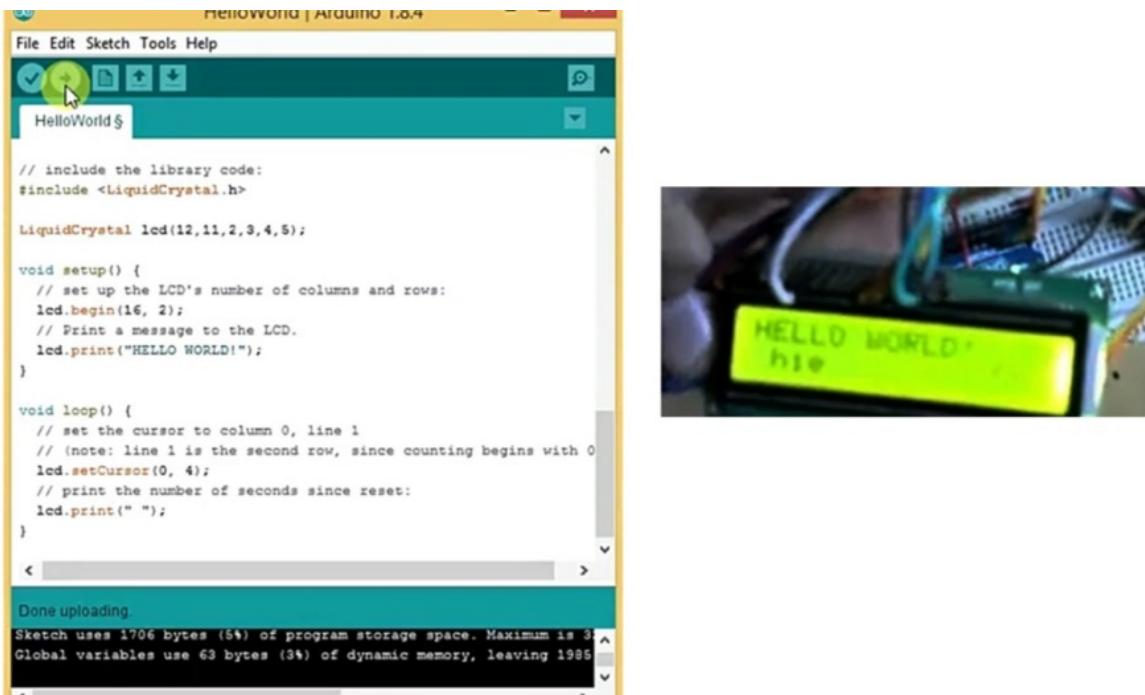


The screenshot shows the Arduino IDE interface with the title bar "HelloWorld | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, save, and other functions. The main area displays the code for the "HelloWorld" sketch:

```
// include the library code:  
#include <LiquidCrystal.h>  
  
LiquidCrystal lcd(12,11,2,3,4,5);  
  
void setup() {  
    // set up the LCD's number of columns and rows:  
    lcd.begin(16, 2);  
    // Print a message to the LCD.  
    lcd.print("HELLO WORLD!");  
}  
  
void loop() {  
    // set the cursor to column 0, line 1  
    // (note: line 1 is the second row, since counting begins with 0  
    lcd.setCursor(0, 1);  
    // print the number of seconds since reset:  
    lcd.print(millis() / 1000);  
}  
  
Done uploading.  
Sketch uses 1900 bytes (5%) of program storage space. Maximum is 32256.  
Global variables use 59 bytes (2%) of dynamic memory, leaving 1989
```



The thighs are a feature of the program here. But what happens if you want to read something else on another way, give a value for one hour to write something else again. Let's try with H ie IME HIGH Clean the room. Okay, we will print it in another line that will wait for some time and we will reset Arduino. It will take a second and see that we call my mistake. I wrote it to the fourth row and we will write in the first row and the fourth column. Okay, we will improve the program and wait a moment. Now you can see that the high word has more than four columns on the left.



Okay with the thigh method, we can write words on the LCD screen, we can print variables and we can get real-time data on autonomous Arduino without using a computer. The thighs are called LCD and the thighs are the LCD modules I use here. It is produced by a high tower and it is a very stable CD and they are also CD providers who come in four out of four, they have differences when diagrams, but they work mainly in sin, Then need their special library.

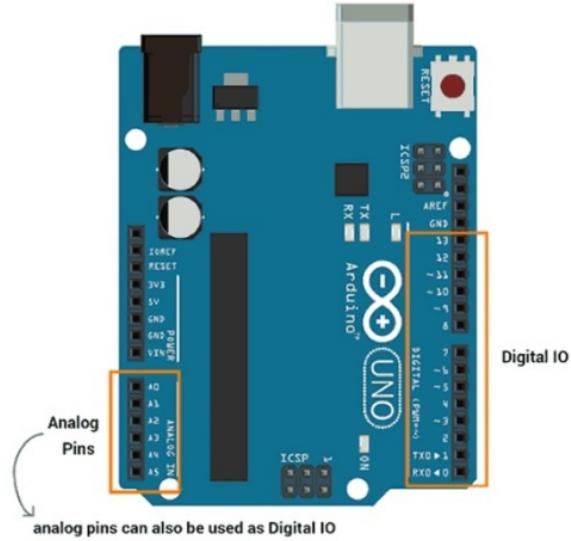
## QUICK SUMMARY OF THE MAIN DIFFERENCES

The male or clear difference between these three controllers. Like my Arduino and Rasberry controllers, before we get into the details, let's get started with the Big Mac controller in its name. It is a microcontroller-based device, meaning it is in seclusion in controlling small devices. There he is. It is only used for projects such as doors for motion detectors of heat detection devices or even for small RC robots. You'll also hear people talking about old boards if you talk about Big Macs or not. When a project is tested on board and engineers make sure it is working,

they move on to a more permanent prototype called a circuit board. As you can see in this image, it is a circuit board. That's great, my control is active. This is an electronic item. It is connected to the IBC so you can easily control it via B.C. E. or a controller can control the whole process. This is a predecessor as you can see. That's the result. This LED illuminates a rotating motor. And this puzzle will send a sound in Arduino terms with this great microcontroller. The Arduino, for example, is my controller, which means it is in seclusion when it comes to powering small devices like sensors, motors and lights. It's like a big controller but present on board as you can see because the controls are on. I see this board is an animal control and you are not buying well, you are not buying this controller yourself. For example, suppose you buy an Arduino board in combat or as a complete product. Because of this, the applications we know of are best suited for projects such as development. I wake up to a motion detector alarm or even a small alarm, but you'll also hear people talking about prototyping with the Arduino, the process of rapid prototyping electronics. We don't need to connect such wires or solder elements. And my controller, if the prototype is successful and the device works, it can be built with a printed circuit board on a larger scale, like my controller. So you will know that you know my front runners are similar. They have a lot in common in many areas. We will talk about disability and you will talk about the time to choose. Arduino - all our best mac you need at all times. My control over the Arduino. But now let's start with that information and move on to the Rasberry from time to time. On the other hand, it is under my control and not designed to control sensors and the like. It is an entire computer with an operating system that can be used as a unit. The operating system is so minimal that you will need some programming skills to make the most of it. But it's one of those great things to help people learn to code. It's also very good to act as a server. It can communicate with other computers as an alternative to Chrome's pricing or provide log information and data. This was seen before. When I'm tired of it has eyeballs, so I mean the computer has a Sibiu specially designed for the Rasberry. Thanks for watching this tutorial. Hope you've enjoyed the information in this tutorial so far. If you have any questions, you can post them on Q & amp; Get on if you want to live. I'm not alone or tall. I can not. Everything is alright. We have courses that teach you how to assemble on a circuit board, how to program an Arduino, or how to program Big to build a controller.

# HARDWARE POWER AND CONNECTIVITY

We'll discuss the equipment contrasts between my Macon our Arduino Andreas Billyboy when you take a gander at Big Mac from close to us we know close to our soul by it's exceptionally evident that the equipment of the telephone is a lot between them. We should separate it. As to the Arduino Bauer subline, necessities are basic. You can plug it into your PC or a battery back and it will begin running code right away. If the bar is detached it will stop. There is no compelling reason to run a closure cycle. Beckmann control is fundamentally the same. Large Mechelen choke body supply prerequisites are significantly more image of volleyballs supply will manage their work and got it just to close down interaction. When done unplugging them supply and that is it the Rasberry by then again because it has an all the more full-highlighted PC framework or processing framework set up ought to be closed down like a normal PC and can be harmed by Balatka Big Mac. Arduino and that is the soul by having an exceptionally low Bordereau and can be around for an extremely prolonged period without utilizing a lot of power. Alright, presently we should discuss availability that raspberry by comes fit to be associated with the Internet. It has an inherent Ethernet board and it's exceptionally simple to get. You inquired as to whether I don't will give it remote availability also. You can see and this is the reason. Dongle. So this respited bicycle be associated with the web 12 lessly this is one reason why they purchase a gadget of decision for things like base on a web worker blender worker and VBN. Been then again doesn't have any implicit capacity for network. On the off chance that you need to interface it to the Internet, you should add a piece of equipment that incorporates an instrument board. On the off chance that you need a qualified dongle or Arafa availability, you need an alternate piece of equipment. Again because the Arduino is intended for equipment projects in the condition of programming ones it needs a digit of fiddling or dabbling to get it associated. Like no enormous Mac doesn't have any underlying ability for the network. If you need to associate it with the Internet you should have an additional piece of equipment that incorporates an occurrence board. If you need Wi-Fi availability you will require an alternate piece of equipment. Again if you need to utilize an SD card or you should be driven with my control you need to fabricate an interface circuit. So that is it for the equipment. Concerning and availability next, we will discuss info and yield twists on additional intriguing themes.



We'll

discuss the equipment contrasts between my Macon our Arduino Andreas Billyboy when you take a gander at Big Mac from close to us we know close to our soul by it's exceptionally evident that the equipment of the telephone is a lot between them. We should separate it. As to the Arduino Bauer subline, necessities are basic. You can plug it into your PC or a battery back and it will begin running code right away. If the bar is detached it will stop. There is no compelling reason to run a closure cycle. Beckmann control is fundamentally the same. Large Mechelen choke body supply prerequisites are significantly more image of volleyballs supply will manage their work and got it just to close down interaction. When done unplugging them supply and that is it the Rasberry by then again because it has an all the more full-highlighted PC framework or processing framework set up ought to be closed down like a normal PC and can be harmed by Balatka Big Mac. Arduino and that is the soul by having an exceptionally low Bordereau and can be around for an extremely prolonged period without utilizing a lot of power. Alright, presently we should discuss availability that raspberry by comes fit to be associated with the Internet. It has an inherent Ethernet board and it's exceptionally simple to get. You inquired as to whether I don't will give it remote availability also. You can see and this is the reason. Dongle. So this respited bicycle be associated with the web 12 lessly this is one reason why they purchase a gadget of decision for things like base on a web worker blender worker and VBN. Been then again doesn't have any implicit capacity for network. On the off chance that you need to interface it to the Internet, you should add a piece of equipment that incorporates an instrument board. On the off chance that you need a qualified dongle or Arafa availability, you need an alternate piece of equipment. Again because the Arduino is intended for equipment projects in the condition of programming ones it needs a digit of fiddling or dabbling to get it associated. Like

no enormous Mac doesn't have any underlying ability for the network. If you need to associate it with the Internet you should have an additional piece of equipment that incorporates an occurrence board. If you need Wi-Fi availability you will require an alternate piece of equipment. Again if you need to utilize an SD card or you should be driven with my control you need to fabricate an interface circuit. So that is it for the equipment.

## COMPARISON OF SOFTWARE

Now after we put the difference between Arduino Spirit hardware by the Big Mac Controller we can talk about software to understand when you want to use one board or another you need to know what can be done and a little depends on the software that complicates or for Gather this problem. Arduino is not equipped with any software Bursar or Blissett has a very basic capability for Brettthe the code that receives and the Alt-Tab hardware function connected. But the board does not have any operating system or interface besides the ideal integrated development environment. And the right side. What this means in practice is that you have to make software running in Arduino using your ID will make a series of commands that will be interpreted and act as simple instructions say something like 10 that lights up for three seconds to turn it off, the green light in the fortress seconds turn it on and repeat. You can do things that are far more complicated but you still need to make your program. Fortunately, there is a large Arduino community that stretches the whole world which means that if there is something you want to do with Arduino someone might do it. You can see their code modify it and make your Arduino do exactly what you want.

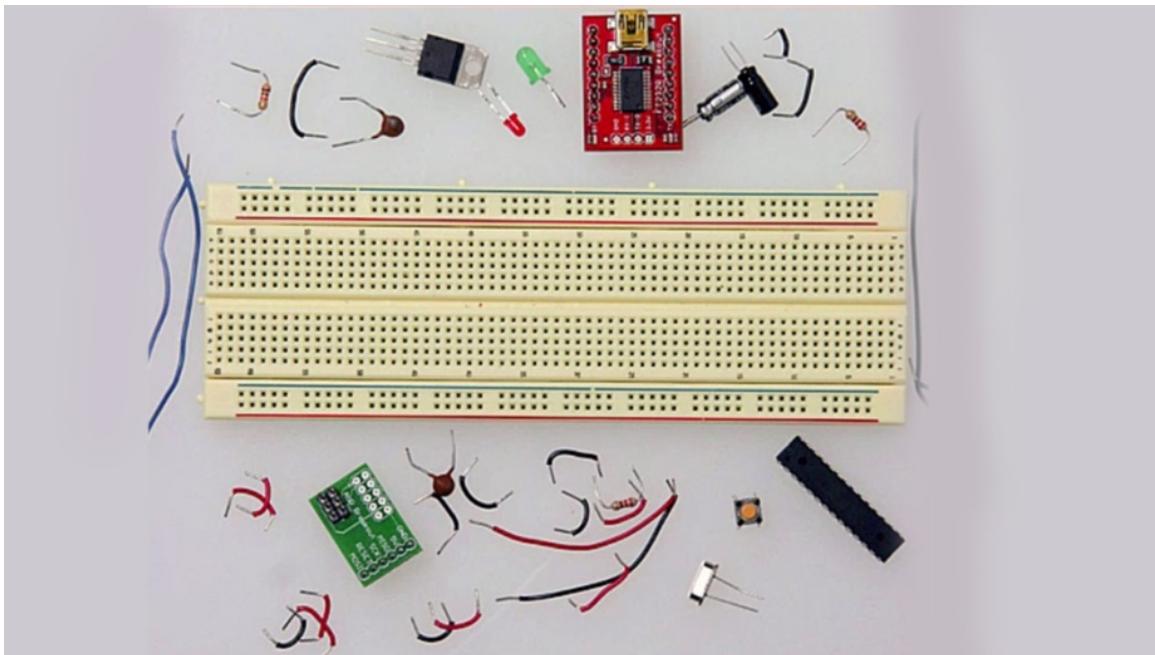
This is a great way to learn the principles of coding and prototypes too. That is why Arduino is a suitable choice for anyone interested in electronics or beginners and any children who want to create something or love to create something. The same thing with my controller. It has a crazy idea that allows you to write a set of course that tells the controller what should be. Instead what I spent by coming was stacked with a fully functional operating system called the recipient. This is for us based on Debian Linux and made specifically for that body. There are several other operating systems that you can use with balls that are mostly Linux-based. But Android can also be installed on what is called the trust built by Rosby on the desktop. But I think the system for the Billy-Boy Boy beginner system is not the only software that belongs to the Lund person. There are also some useful applications that you can use to complete different tasks. One of the most common uses of spider bite is as a media server that provides the code and Blakes are a

popular application code. And then, you can download the game server application calculator and even I buy Office Office You know the document, of course, you can't write your book for us too. That's one of the best reasons to get it for them to decode by telephone. To lead the code and programming language by telephone is the suggested language. But C and C ++ Java and Ruby are all installed before on the board. While Arduino can be tweaked into other Subodh native languages is the best choice..

## **EXPAND THE NEXT CAPABLITIES**

So let's name our big Arduino Mac frontalis and it was built by our very capable Littlemachine that can help you learn and do other things. But at a certain point, you might want to move beyond the basics and try something little more forward. This is one place where we know and shine a large microtonal. Hundreds of chips allow you to expand stock board capabilities with things like connectivity and connectivity Life-I motor speakers control and microphone capabilities or touch screen cameras radio transmitter graphic processing and almost all other things you can think of \$ 20 to \$ 40 Arduino or Beckmann It becomes something else, the chip is called shivering and very easy to install. What you do is just bless them above you and we know and in some cases, large microcontrollers are sold with money and money can only sit on making an IPS installation as you can see. And this example is a shield on Arduino boards supported by more independent boards and does not have the same expansion ability as my big controller. There are several hats available that add additional hardware to buy. However, it gives you some very interesting possibilities. For example, you can add capacitive because all this G-B as R. G. Touch screen panel and even 3D you have the self sensor. Then you ask. Also if you add functionality with a dongle, for example, get a wife on connectivity everything you need to do is a plug on my wife does not understand. However, even with these options, the spirit body does not have many choices to add functionality. Not to say that purchases are not capable. You can still do almost everything you want with it. You might need to get a little more creative or switch to Arduino which is also possible.

# BASIC PARTS REQUIREMENTS

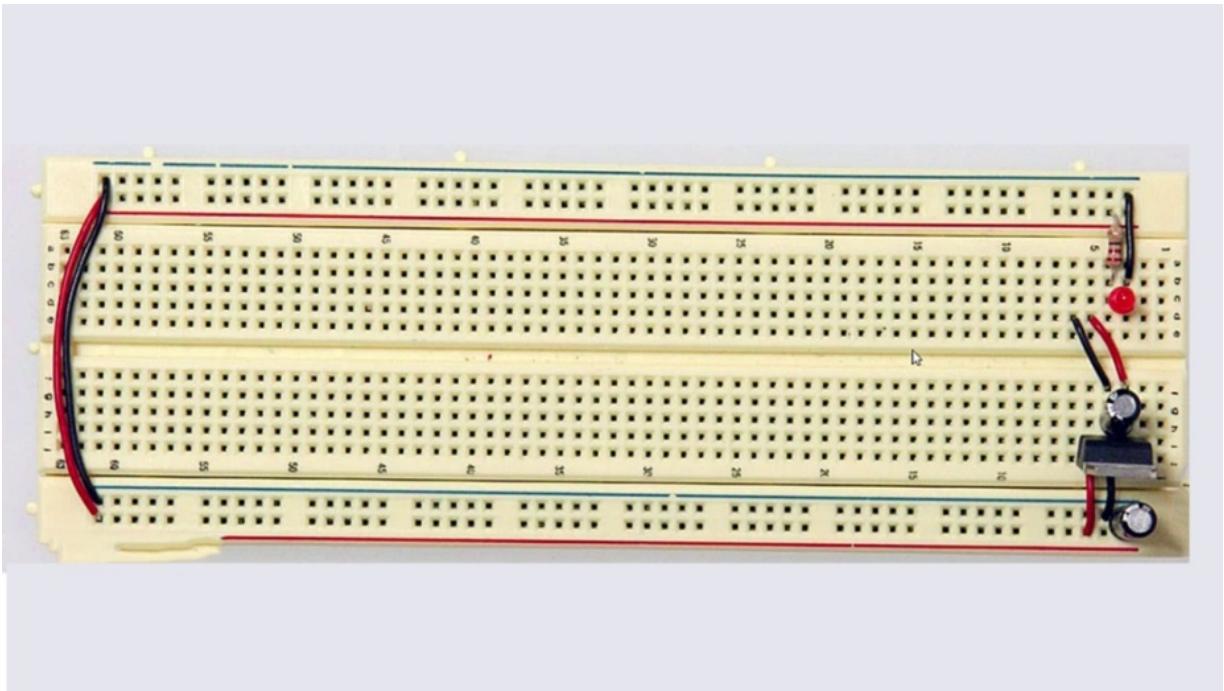


Eight hundred sixty-eight three hundred twenty-eight. If you are making a controller and FDDI breakout board from sport fun or any other store you could also use the new US ASBI Mini will clarify all of these components next. Now let's see the bars that are required for you to make your own Arduino board at home. To do this you need basic bars for wedding bands we know such as a billboard which is this one entity. W. G. As you can see. These are the whales. Or you can simply use it anywhere that is available for you. It's a 0.5 voltage regulator which has a 5-volt regulator. This one this is the item 5 will trigger it or takes 12 volts or mind volt provides 5 volt built 5 volts voltage for boarding up Arduino 2 LEDs. These are the toilets for the boiler and the ablation to 220-ohm resistors for the LEDs. One thing on resistor for the reset button to 10 micro odd capacitors for that Kristen. These are the capacitors. As you can see 60 megahertz clock Kristen and 2:22 Beco Fraud capacitors. We need also a small that is normally ORBIN off button. For example, in the Omron type B-3 path, all of these components are available online and or flight stores. They are very cheap components. You will make a very reduced cost.

# **THE PART NEEDED TO COMPLETE THE WORK**

Old people are bored at home. Now we will talk about more advanced. It must be available to you to beg the board. First, you must be on the serial communication board. You will need 50 to three to become a breakout board. There are two options available from them from spark fun or other online stores. There are after two or three hours two hours, you we see a real breakout board. There is an art we know. Seriously you have to get bored. If you plan to use that option and not Soledad Haydar to the Breakout Board will now be the right time. We will clarify this option later in this course. The second thing you need is the boot loading yours on the Chip Maigre. There are several options for clothes or mega chips. Some of them are included in this tutorial. If you want to boot the chip you use a breadboard and additional balls you will make your life easier but not necessary. Ever. Our programming adapter from a fun spark or another online store can do the job. We will talk about boot loading and why we need to boot or make CEB in more detail in this course. But for now, you need to understand that. I see where the ball curse must have a code that will help but realize that it is used because we know the brain so we will burn the code for this. I see that as some options we can use to burn the code.

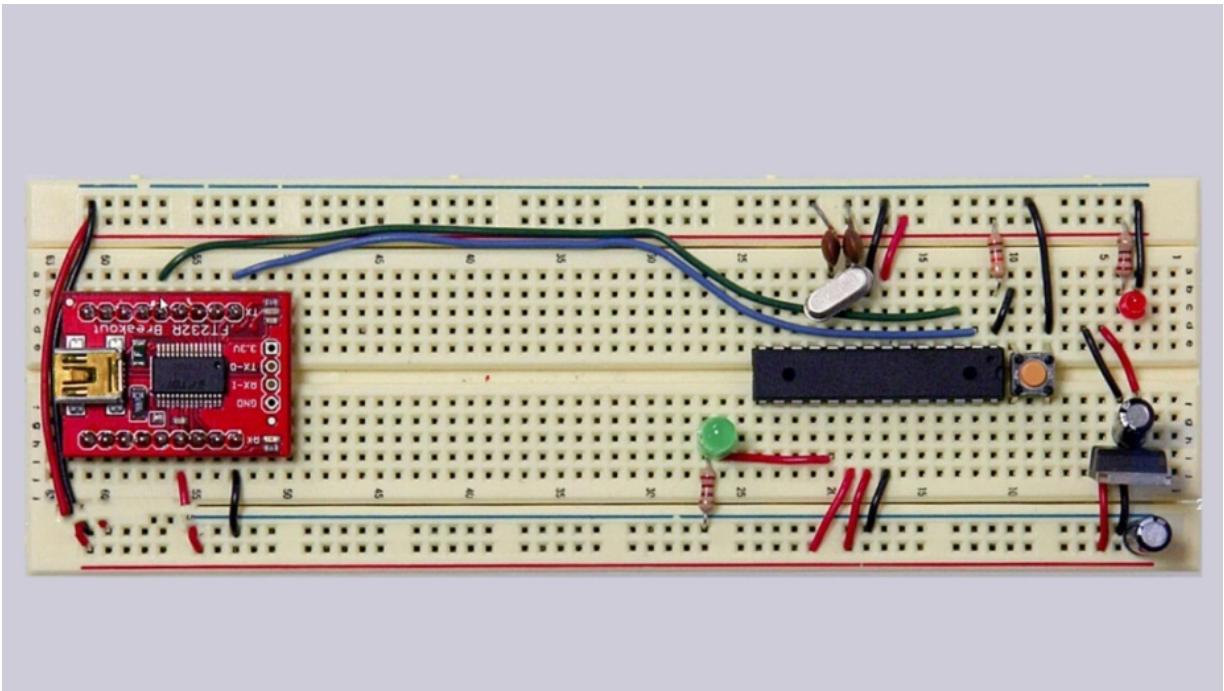
## **POWER SUPPLY WIRE**



If you have worked with a microcontroller, chances are you already have a preferred way to install the supply to your board. So please and do it like that if you need a reminder. Here are some pictures of one way to go about Fanjul. It is a supply of five vault balls. Let's start by doing this at Bauer and Ground Words for where your voltage regulator will. As you can see this is where our voltage regulator will. These are the two main board lines. This is a positive supply and this is the supply of land or negative. Here we will put our regulator. It has three bends. It will take nine volts and produce 5 volts for our bread cutting board. So here we have two of our Tobler lines. Going forward because you can see our underground cable at the bottom of your board connecting each rail. But on the power line, as you can see, we need to connect these two lines with these two lines of this blade so we need to add two lines to make a bridge between regulators 7 and 054 and lines to the board. Here we add regulators and we add two capacitors. The regulator is a Z or 220 will be. Where is the input from as far as our supply runs on the left as you see here? The ground is in the middle and output 5 volts are on the right next to the front of the regulator and power out and the land cable that is connected to the right and leaves the breadboard rail also acts like a Wreford capacitor between the final regulator and land and I don't make it far in the trail Exactly between the

Underground Bouwer as you can see here. And here we are at the bar on the ground between the two underground silver strip on a copper star signalling the land foot. So, you need to make sure that this silver leg is a land leg and connected to a black wire here. You must ensure the same cabin silver with black wire. This capacitor is used to make more say noise-free voltage and pale and 220-ohm resistor here on the left side of your board opposite the voltage regulator attached to the bar like this is a great problem-solving trick. You will always know when your board is purchased and quickly knows whether your board is abbreviated. And Blackwells on the left as you can see to the left of the voltage regulator here is where your bar supply will be plugged into. So say that you have 12 or nine full board supplies. You must add it to these two trash cans. This is positive. This is negative. This will go through all the capacitors to the regulator. The regulator will produce 5 volts and this 5 volts to the bar here. Now the exit is for boards and black wire for the ground. Be sure to just attach the supply of bars between seven and 16. All and know is and you will not get your 5-volt regulator higher and your regulator can be damaged. So save between seven and 16 volt 9-volt batteries or attentive DC supplies or 12 volt DC supplies suitable for this case. So as you can see here and here. Seven to 16 volts and here is soil so now the basics of the Board are finished you are ready to load on the chip. As you can see this board where you cut bread.

## **ARDUINO IS READY**



Now we will add ASBI to a serial breakout ball to our Arduino Breadboard circuit. If you haven't added a lot of headers to your Breakout Board, you have to do this now. Connect VCC from brick with the brick bar on the ground to the ground. So as you can see here we have to connect these two dams and this land. After you beat the Syrian border now the board. This is the bend as you can see, we just connected the Underground VCC in previous lessons from the previous lecture. As you can see here, this is the underground reason we connected this is a corner. Curious about what all bends for splash fans. After two-three to break up or just turn it on. In this situation, we will use VCC to supply 5 volts as you can see here from there. Being more to your board and x and x as you see here is that x and here are x we will use both to allow us to become serial communication. Having come out of the splash of pleasure from the break is this. So you can know what every bin does. We will only use these two pieces underground to our surroundings, we know T S D and R S to make serial communication through PS so you don't need to take off the ball. I see or chip every time you want to program it. No, it's time to get the speed of the serial offboard. Talk to your new Arduino settings Connect the DOOR X which has been two in your Mega chain here number two to x from the SB serial board as you can see. This is number two. It runs to ex-Penn tea here and our board and is connected to x bin number 3 which is one of you made the chip to x from US B moved which did not connect Lot X here from USB to the series board. So we connect T x and x and in the previous lesson, we can let it slide connected. He said he was on the ground. So now we have our reading and then he will be ready to be plugged into. Bob and programmed. But wait, there is another step back. If you build, you will remove the chip from your underwear. No, it's most likely to have been dropped several times

alone and therefore has been purchased so you don't need to move further in this story and you will have a fully functional breadboard that has a fully functional ARDUINO USP functioning up. However, if you buy some extra MAIGRE 328 at Chip Maigre 168 from an online store, there will be no boot with Arduino bootloader. Except for fruit trees and dust if you buy this in the food industry maker they already have bootloaders on them. But if you buy this ice elsewhere, you have to make what we call the boot loading. What this means you will not be able to program your harass because it uses a serial break outboard and hardware and software. So to make your new chip useful for us, we know you have to boot them and have to do a boot loading step.

## **YOUR BOOTLOADING CHIPS OPTIONAL**

It's optional because we clarify why you might need to load booting in the previous educational loading option. There are two options to load a good chip. The first is quite easy and the other becomes a little more complicated we will cover both so it will be at the safe and good side bootloader at the Mega chip using our board and on our program. This is the first option. The second option is boot loading. You make a chip on your newly prepared breadboard with and if they are programs. So let's talk about the first option. Which is good butadiene using a debatable board and on our program, before going further you have to know that there are also various types of strange diamonds but the two most commonly used



AVRISP mkII



USBtinyISP

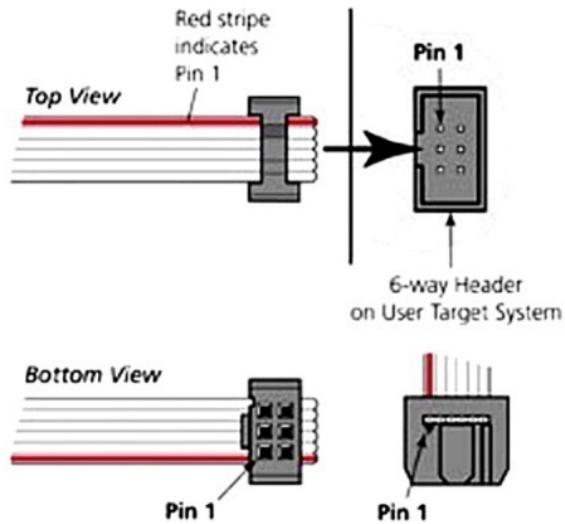


ArduinoISP

Fun is very useful because you can see it is a very young breakout board every hour. This adapter breaks the enampence of arms to six inline trash cans for attachments that are easy to breadboard. All trash cans are also labelled making it very easy to connect to your chip. Because you can see Volt Ground 5 on my ass or C K that I say or me. This is one of our programming adapters and is very easy to use the adapter. Very cheap. You can be birched anywhere. One or two boxes. Now let's continue and talk using Breadboard we don't want it if you don't have an adapter running Arbel that you can always still evaporate without it. However, it will be more. I have an urge to set two images left to left or right. In this case, it is a good reference when connecting to an ATM that makes CEB without an adapter board. Images tell you what all holes and sixpence for our blog. And you only need to attach the cable to the end and drown it to make a chip. Because you can see the top. It's about all of you.

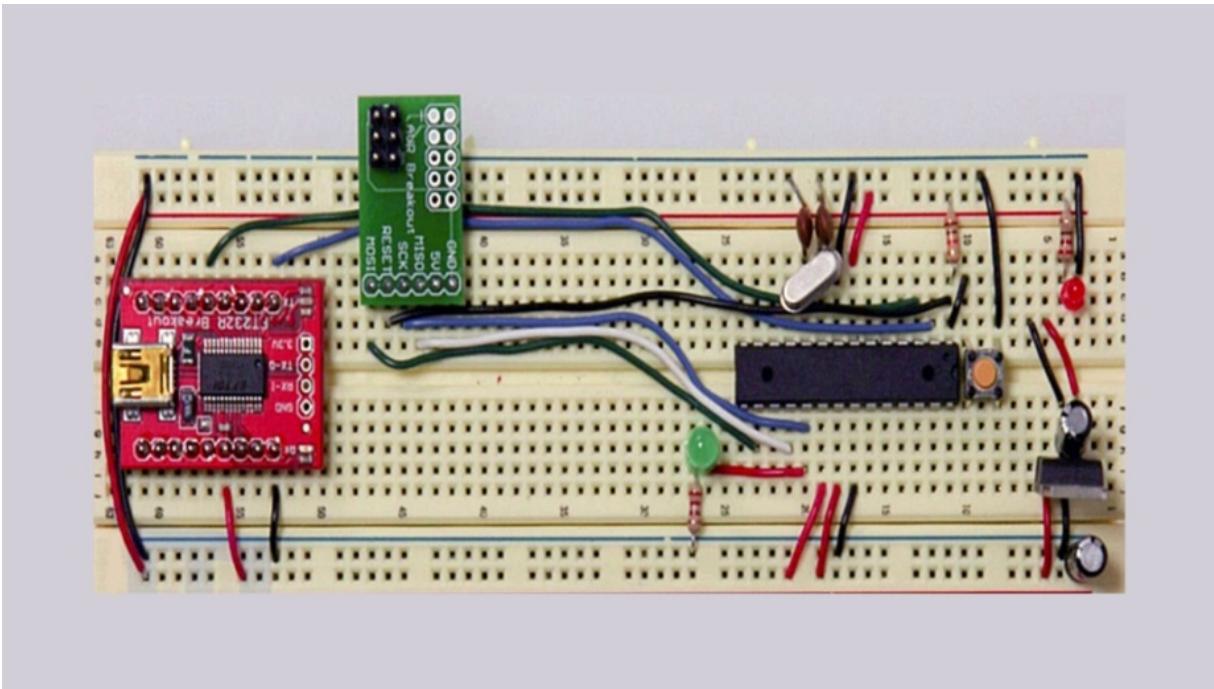


Fun is very useful because you can see it is a very young breakout board every hour. This adapter breaks the enampence of arms to six inline trash cans for attachments that are easy to breadboard. All trash cans are also labelled making it very easy to connect to your chip. Because you can see Volt Ground 5 on my ass or C K that I say or me. This is one of our programming adapters and is very easy to use the adapter. Very cheap. You can be birched anywhere. One or two boxes. Now let's continue and talk using Breadboard we don't want it if you don't have an adapter running Arbel that you can always still evaporate without it. However, it will be more. I have an urge to set two images left to left or right. In this case, it is a good reference when connecting to an ATM that makes CEB without an adapter board. Images tell you what all holes and sixpence for our blog. And you only need to attach the cable to the end and drown it to make a chip. Because you can see the top. It's about all of you.

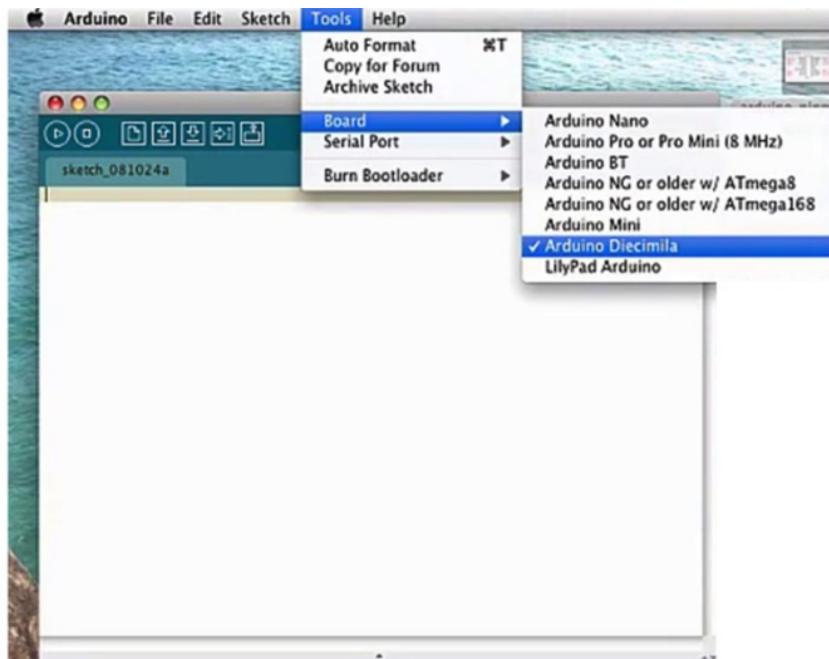


It is six directions and the use of a lenguk cable head as you see in my ass. Oh, I see. Care for setting VCC Ground. Or US II. This is Sixpence. This image is viewed from the problem below and the hole label. Watch the square about your cable orientation and as you can see this will show you orientation. This box is here so you will not label it the wrong way.

# COMPLETE AND BURN BOOTLOADER



Now there is a drum adapter that sounds like you see here to breadboard with beef that matches the ground why you just thrown at five times. Has chased behind with a bore. Wow, it just from this is an adapter that we use moving forward. And this will lead to the last four cables needed by what has ever existed. Our programmers for loading robot mode. Be sure to refer to us to help make this easier until it's stupid. The adapter will go to 19 or and we know there are still 30 mega chips you say Ben from your adapter will go to one of your Nigger chips or us. I have some adapters going to pin 17 or there are still 11 of you, I can make the chip now. This is a cable that is connected to Andrina chip or makes the chip almost just a cable or you might be a breakout board here and plug in your six Ben blogs. Or if you are an adapter, the black knob of six trash cans must face up towards the mega chip in the next table will show you how to use the software device box to burn your bootloader. So this is a simple Sibyls without using Arguido boards that only use every programmer and breakout board.



Time to burn Arduino API and then go to the tools as you can see then select your board to choose the type of void you want to use the effect that you will do on your chip. Most commonly you will use that this says the area of the smell or the latest version of Arduino for that might be an IAB. However, if you want to boot Arduino Linley Bad Arduino a lot of Arduino or an older Arduino version. Choose the right board from here. We recommend this option. Don't argue with seniors after choosing to go to the Brande bootloader tool and choose a program or you will use this class we use every hour I will be fine. We only need to choose it. After you choose your program from our program or we will start loading good items I can make chips and messages will appear in the status bar here. Which leads to burning bootloaders to enter the board. This might need a little light to blink on your program and then you will see this message done by Bender Manbootloader. When finished with the status bar clothing will be updated with the message made and bootloader. Your chip is now ready to be programmed using the software. The bar cycle is congratulating your Arduino and the new Mega chip you will run a similar Blink program in 13. If this is not the case, try and get it with one. If this works most definitely successful and you just created your own functional Arduino board, not on occasion. And I made a chip with the M-K ISP to Celtic and squeeze Nungarrayi a long period used when it only takes a few minutes. And there is an ISP Tinney fish much faster. However, there are times when after 5 to 10 minutes still seems to be loading. I found this or habs. This is terrible checking the data flow and after giving it a 10 minute time or more I usually revoke verbal or just to find the building blocks must be successful and have ended for a long time. I can't this enduro. So and you take all the responsibilities for whatever you have with your chip. But in my experience, it has been successful.

# ATMEL ATMEGA 328 DETAILS

Atmega three to eight details are good we have talked about before the profit is the black dollar chip on the board. Thighs are known as Atmel Atmega three to eight P P P, P Q because it runs on five volts and it is a power-saving chip.

## ATmega328 Features

### Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory
  - 256/512/12/1K Bytes EEPROM
  - 512/1K/1K/2K Bytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
  - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
  - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
  - I/O and Packages
    - 23 Programmable I/O Lines
    - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
  - Operating Voltage:
    - 1.8 - 5.5V
  - Temperature Range:
    - -40°C to 85°C
  - Speed Grade:
    - 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
  - Power Consumption at 1 MHz, 1.8V, 25°C
    - Active Mode: 0.2 mA
    - Power-down Mode: 0.1 µA
    - Power-save Mode: 0.75 µA (Including 32 kHz RTC)

Okay, the frequency is well has a built-in clock that runs at eight megahertz, but by adding crystals on it, you can see that here the thigh is crystal and after adding crystals on it, we can run it or something Stain Megahertz and all Arduino boards With Atmega three to eight workers at the processing speed of 16 Megahertz. And the next section is flash memory or you can say it's just microcontroller violence. It is 32 kilobytes, and I invest 32 kilobytes it means that it cannot store photos or other data but is made only to save the program written on the chip. This program is written on the chip in hexadecimal form, but we have to program it in Arduino's idea in C++. Will Will Arduino is a special form of C++ which is very easy to use and can be used for non-professional sounds about the PIN. It has 28 physical pins, well you can count here it has 28 more than 28 pins because the land code has been given twice and also three to 313 power adapter is also added to the board and three pins for input-output reference. Good? The next point of digital moulds well on board you can see that it has 30 straight digital pins. Well, I

want to tell you that first of all considering that zero print and pin one or should we use it for processing because if we do a serial connection and if we use a zero pin and word pins there, then there will be a mistake in communication Serial so we have to choose that we don't prefer zero pain and one pin for our programming and other purposes. It has five analogue pins like Zero A two fives well all analogue pins can also be used as a GPIO digital pin bus, but not the point that zero pins must be written in detail with 14 and one digital pins for PIN and so on.

## **DIGITAL AND SERIAL PORT**

Okay, now, about digital signals. For that, we must set the Arduino pin as input so that it can read it. So can read the voltage again, we have to open a serial monitor we have to initialize so we can see the light conditions? Okay, let's go to Arduino okay, let's go to Arduino Open the Arduino panel Okay, go to exemplify certain basics and select the retail level cereal, okay, in the thigh program it is written that first the value of the pin that we have to read it is said to be a two-pin so go to pin two and connect the wire on it okay, in the SR world setting you can see that in the word settings, you can see that first they have initialized the series of monitors by setting up to 9600 boards and they have Pin mode or encouragement is done You can avoid writing data buttons, just write the PIN like a two PIN and setting the input so that it will be read. Okay, now go to Void Low.

File Edit Sketch Tools Help



```
/*
DigitalReadSerial

Reads a digital input on pin 2, prints the result to the Serial Monitor

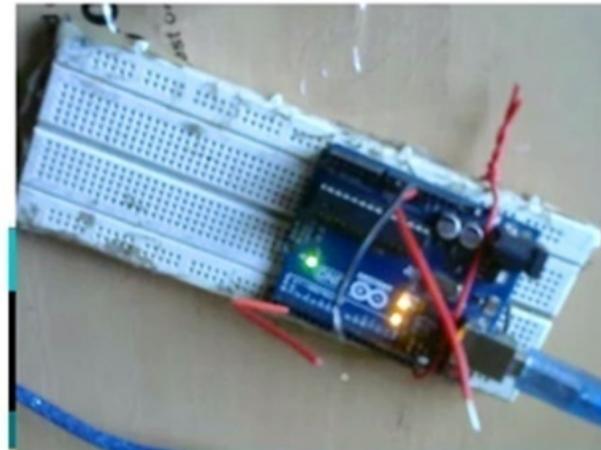
This example code is in the public domain.

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

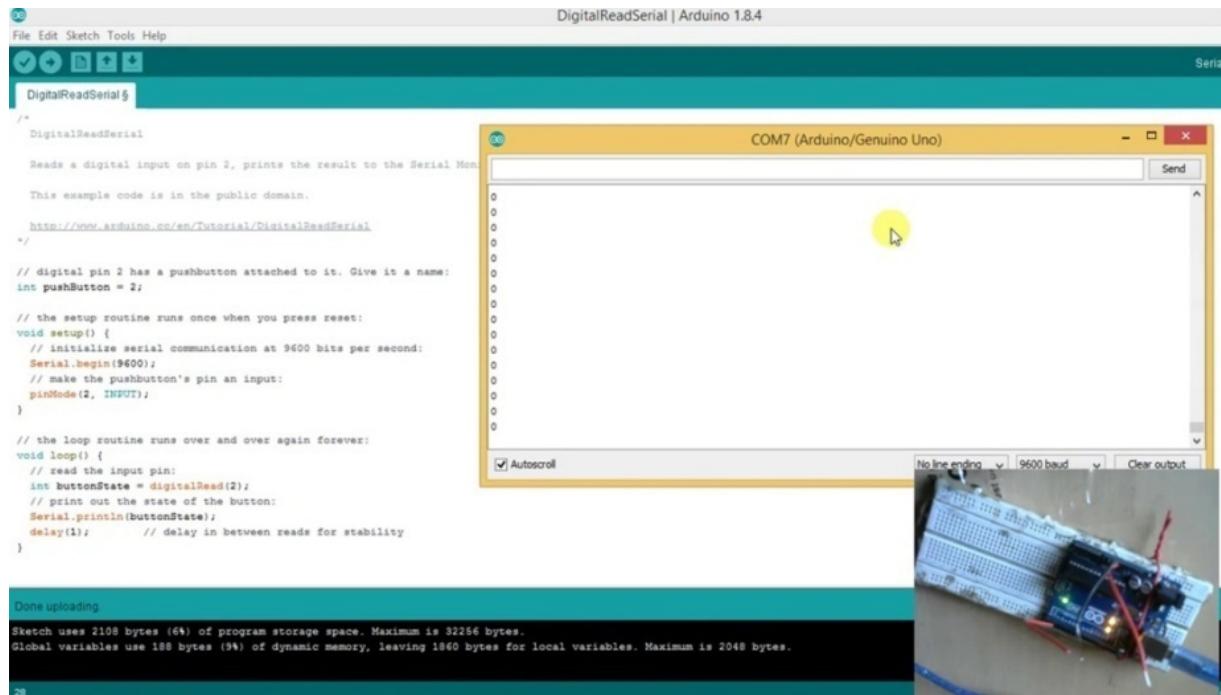
// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(2, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
  int buttonState = digitalRead(2);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1);          // delay in between reads for stability
}
```



It is said that the state of the old button will be read digital push button Push Verdun capital and write value for your better understanding and clear knowledge about things in the world loop Program in the Void Loop This program says that the state variable button will check the state of the data into each time The state of the two data pin will be changed from zero to one or otherwise. This will provide value on the condition of the button and the button status will print it and the next line you can see that it can check the microcontroller in each of my milliseconds. Okay, first of all, upload code into the microcontroller. We will upload it to take a few seconds to compile it and then converted into hexadecimal and download it to the board and now converted to open a serial monitor. OK, now we can see the value of garbage or changing values from zero to one do not mind that it is only the garbage produced by the microcontroller.



Okay, here you can see three cables. First of all the wire connected to a digital pin for two seconds the red wire is connected to five volts and black cables that are connected to the ground. Every time you connect a digital pin to a black wire, it will display zero fixed value. Okay, the thigh says that the data value has a low value or connected to the ground. Every time you connect it to five volts, it will display one more value allow me to show the day of the pin connected to a two digital pin connected to the ground. This will show zero fixed value. You can see that garbage after trash you can click on it, you can see that it has a fixed value unchanged because it is set to ground or set to low. Okay, the other time the latest is connected to five volts and you can see it has a fixed value. The moon will cause trash we do not want garbage but every time we connect it will also improve the pleasure of value. Well, thighs are programs for PIN and data thighs for digital readings on serial monitors.

# MOTORS TYPES

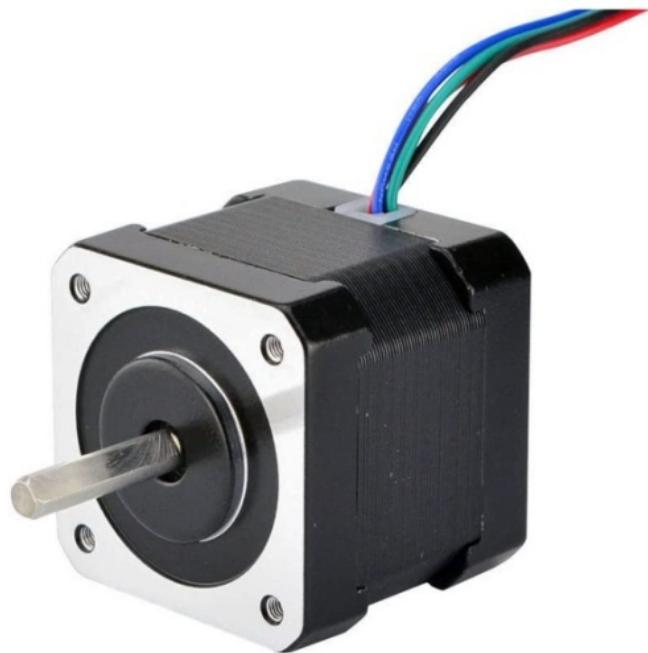
We will learn about the engine and how to control the engine using Arden, first, see the content or content is a type of engine. It can only the following content will be a single DC motor, a DC servo motor, then once we talk about a step by step in the last three tutorials we will learn on the weather engine controlling and we will experiment control the engines. Okay, let's start with our first topic. Before starting the type of engines, I want to tell you that an engine is an electromechanical instrument that takes electrical energy as an input and gives in the form of mechanical power, there are two types of the motor in the World The DC engine AC engine in Arduino, we are mainly talking about low voltage CC engines.



Okay, simply start CC engines and their types first of all the CC engine, is a simple engine with two wires, you can see here that two wires. If you want to change the direction, simply change the polarity of a positive connection and another negative and if you want to change the polarity, reverse the wiring and it will change the correct direction of the thighs is a CC engine. Ok, then what is our stepper motor? Well, first of all, it will be the servomotor.



A servomotor is a special type of engine used to give the values in degrees. It is not made to rotate continuously in a continuous way 360, it is intended to rotate during a specified degree, if you set the engine to operate at 30 degrees, it will be locked itself or 30 degrees, it is will connect to 30 degrees and when you try. To change the engine direction if the program is running, it will arise to such a degree. Very important motor thighs area for the manufacture of red toy aircraft because the thighs are a very important engine for the manufacture of toy aircraft because for rudders if you want to define the value of 30 degrees and that it will cause an Another effect on deletions while the engine is running while the engine will turn. Back to 30 days so that it can move to the urban Verizon, the third type of engine we are talking about is here is the step by step, the stepper motor.



The third type of engine we are going to talk about here is not motors. Stepper motors are two types, they are a unipolar engine or micrometre. They are specially designed motors that can operate on computer controls such as the row motor, but they are used for more than 360 degrees of execution, they are useful in the long term, especially in CNC printers and 3D, they can be very high-speed motors, they can be a very low or very basic engine, they have a simple execution mechanism that is set as a coil inside. It is a magnet that magnet is rotated with the changing value of the direction of each coil that activates the movement of the magnetic coil is with the help of a pulse during its driver. We are used to amplifying the computer pulse in high voltage pulses to execute the engine, OK and the last type of engine we are going to talk about the DC well The last type of motor is a DC motor without Broom or a blocked motor.

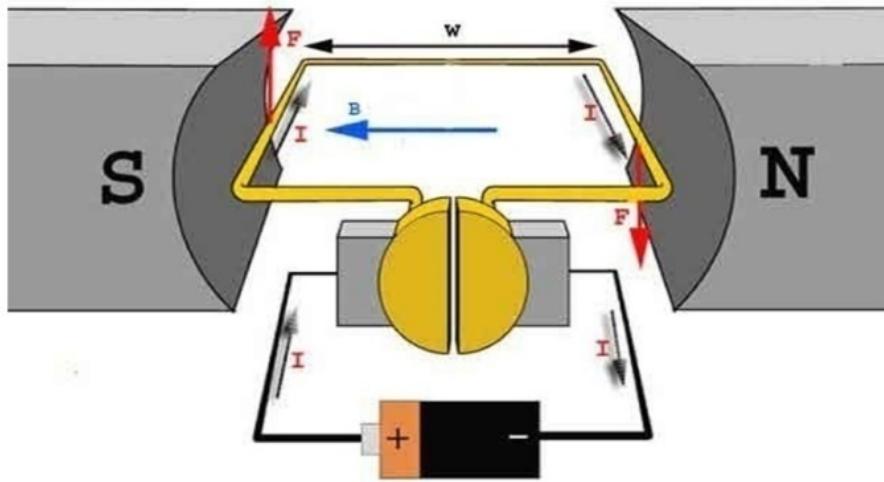


The DC engine again is similar to the step by step, but the basic difference between it is the basic difference between it indicates that it is a free-bowled DC motor means that the effect Functional engine is similar to the CC engine. The mechanism is similar to a stepper motor, but it is used for very high-speed operations, such as quadcopter rotors, all the rotors of the quads are only connected by the block engine because they are the lightest 'They are the most powerful who are the most powerful. speed and they are the simplest types of movement. They are the simplest type of engine for high-speed operations, they can be controlled using the pulse width modulation and specially designed drivers that you can say that electronic speed controllers in the long term. Quadages are also linked to the micro-mantra mostly. are 32 bits to control their operation.

## **SIMPLE DC MOTOR**



We will learn about the engine and how to control the engine with the help of the R & D on the part of the RD of all the contents man of the legislator or the contents is a type of motors that can then contain a direct current engine, A simple driving motor servo engine, then after we are going to talk about the step-by-step engine and in the last three tutorials, we will learn to the controlling nature of the engine and we need to experience controlling the engines. Okay, let's start with our first topic. Before starting the type of engines, I want to tell you that about an engine is an electromechanical instrument that takes electrical energy as an entry and gives in the form of mechanical powers that there are two types of Engine in the world The engine AC in engine DC in Arduino, we are mainly talking about low voltage CC engines.



Okay, simply start the DC engines and their types first of all the CC engine, it is a simple engine would make the reason you can see here that it has only two sons Whether you want to change the direction, simply change the polarity connector. And another negative and if you want to change polarity, reverse the wiring and it will change the correct direction of the thighs is a very next DC motor What is a stepper motor? Well, first of all, it will be a servomotor is a special type of engine used to display the values of degrees that it is not made to run continuously in 360 degrees, while it is intended to rotate for a specified degree. If you set the engine to operate at 30 degrees, it will be locked itself or 30 degrees, it will connect to 30 degrees and we know that you try to change the direction of the engine. If the program is running, it will take place except for Saturday, the AREETHE Treethe thighs very important for the creation of Toy Aeroplanes in red. Because the thighs stop a very important engine for the creation of toy aircraft because for rudders if you want to set the value for 30 degrees and that will result in another automatic effect repressing until the engine turns on 30 d so that he can move the urban very easy. The third type of model we are talking about is here is a stepper motor, the step by step, we are the third type of engine we're going to talk about here is the step by step, the stepper motor is two Types, they are a unipolar engine or

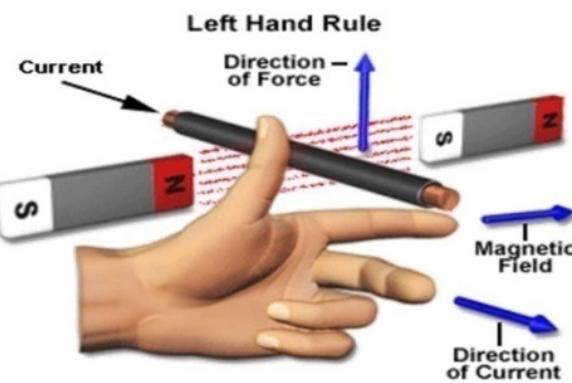
vibrating. They are a specially designed engine that can run on computer controls such as the automatic engine, but they are used for more than 360 degrees, they are useful in the long run, especially in CNC and 3D printers. They can be very high-speed motors, they can be a very low or very basic engine. They have a simple operating mechanism that is said as a coil inside, there is a magnetic magnet is turned with the changing value of the direction of each individual coil that activates the movement of the magnetic coil is at the Help of a pulse while the IT driver is used. To amplify the computer pulse in high voltage pulses for the engine execution and the last type of motor we are going to talk about the CC well and the last type of motor is free-bowled DC motor or BLDC engine. Well listen to the engine similar to a stepper motor, but the basic difference between it is the basic difference between it indicates that it is a broomless DC motor means that the functional effect of the motor is similar to the CC engine. The mechanism is similar to a step by step, but it is used for every secondary school operations such as quadric capturing rotors. All quad copter rotors are only connected by the BLDC engine because they are the lightest, they are the most powerful that in terms of speed and are the simplest type of motion. They are the simplest type of engine for high-speed operations, they can be controlled using the pulse width modulation in specially designed drivers, you can say that electronic speed controllers in the mandate. Quadruples are also linked to Micro Montura. Most of the time are 32 products to control their operation.



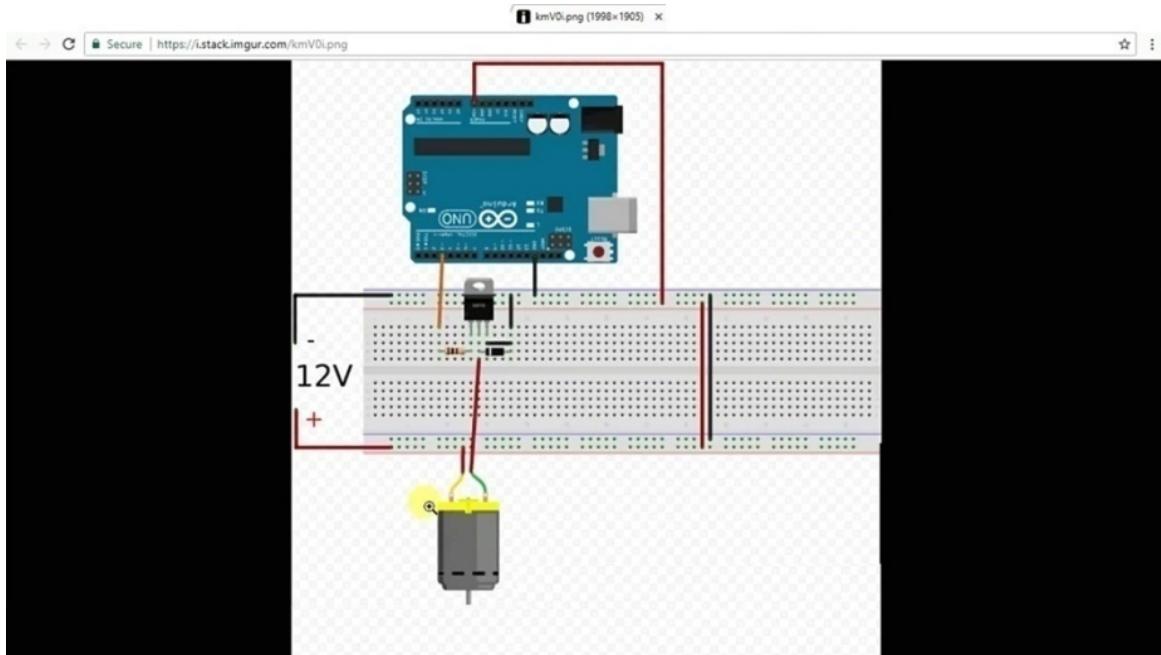
Simple DC engine. It works its diagram and how to connect it with Arduino to get the motion. Okay, first on the CC engine, I want to tell you that their CC engine is the simplest type of engine you've seen because the thighs are a very small device. You can find in the toys that you have seen here the engine of the photo thighs has been taken outside the toy. And here is their CC engine, you can see the B engine. They both have different design, but the thing is that they both run on the same principle. When the power is sent in one direction, it will turn clockwise and when the power direction is changed, it will work integrable. First of all, the CC engine can not be connected directly to Arduino, it requires the execution of a specific driver or transistor. Because if we connect the CC engine directly to Arduino, it will attract a lot of power that can eventually err and you know, sometimes, the laptop can also be injured because of the thighs. So, to control a DC motor, we need a driver or a transistor configuration to execute it. Okay, just see the CC engine. Here you can see the operating principle on the DC motor is a DC motor outside the body, there are two magnets such as the North Pole and the South Agreement and inside the coil every time the load is passed through the coil or the contact centre, so the engine here you can see that these are the pressures of the context and the thighs is called that the suburbs are a copper ring divided into two Parties,

whenever the current is paused here, it will force the engine to it. will force the wire of the coil to enter a thigh direction is because of the law, the thighs are because of the rule of right of the engine where it was motivated by the fact that it was given by thighs ok OKS are known as the rule of the Right thumb of Fleming. You can see here let me make a correction for you. The thighs are the correct thumb rule. The three fingers are in other directions. Okay, let's just see that while working. Here you can see that whenever the loads pass the engine will turn in one direction or when it comes to the centre point here. So, contactors change direction and thighs work as a mechanical rectifier for the CC engine to continue its motion in a direction. The important part, the Sleep of the split ring is a cooperative divided into two or more parts. The thighs are the main reason for motor movement in one direction because when it happened here 90 degrees, it automatically changes a height installation as a mechanical rectifier and the thighs will give the engine a continuous movement in any Which specific direction that the current has been controlled in the direction of the engine direction, can we change the equation here, less start and it will be given here more than this will automatically change the direction from there.

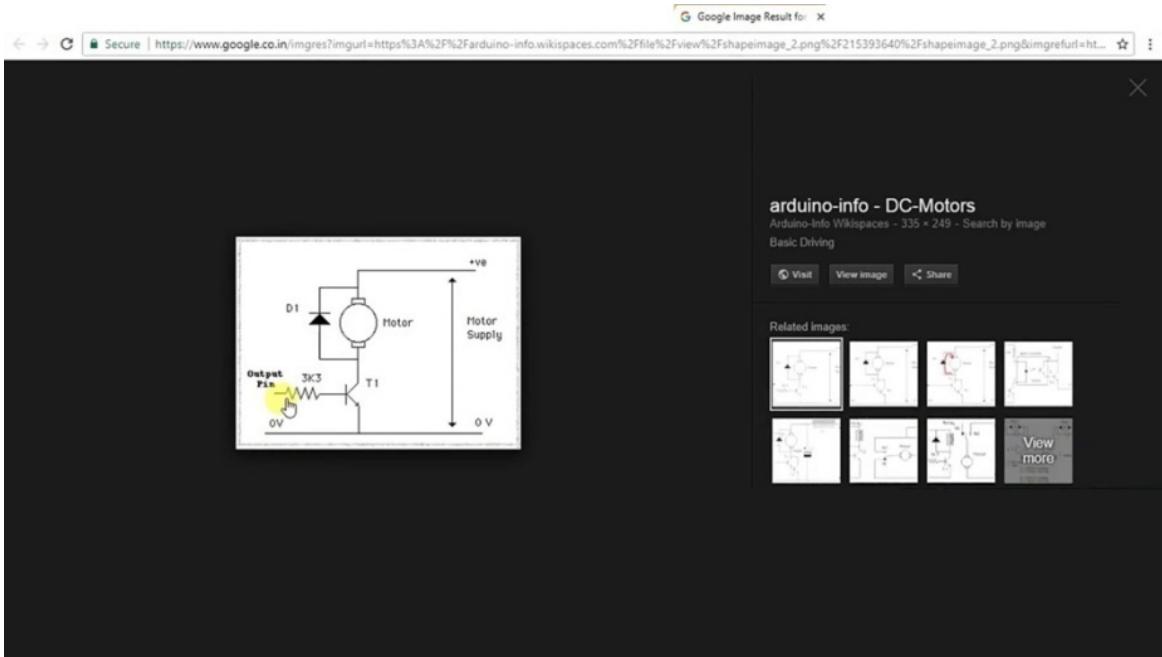
## **Principle Of Operation Of a Dc Generator.**



Now let's see how to connect an engine to the Arduino here you can see that it is running on the rule of the left thumb, but I showed you that it works to the left of the room here, the thighs will cause The engine leadership. Just see on the drivers here you can see that the thighs are the driver we will use here to manage an engine, okay, you can



Find out that the diagram connected here is the diagram of an Arduino and a motor with the help of more than 12 words outside the source. Here you can see that when the transistor is connected, it's an NPN transistor and the wiring, first of all, activate a transistor as I told you, you should be able to give it a value digital aloud and we should give a value of five that it will make the thighs true transistors to assemble and it will eventually connect these two circuits which give it a closed loop and that the closed-loop will take the power source of the power supply of the 12-volt battery and giving the engine a specific execution output. So, by the thigh method, we use the engine with a 12-volt output, but the motor control is performed by the Arduino of the interface between them is the transistor, because the transistor controls the direction of the power supply. Or you can say it's like a power wall functions like water.



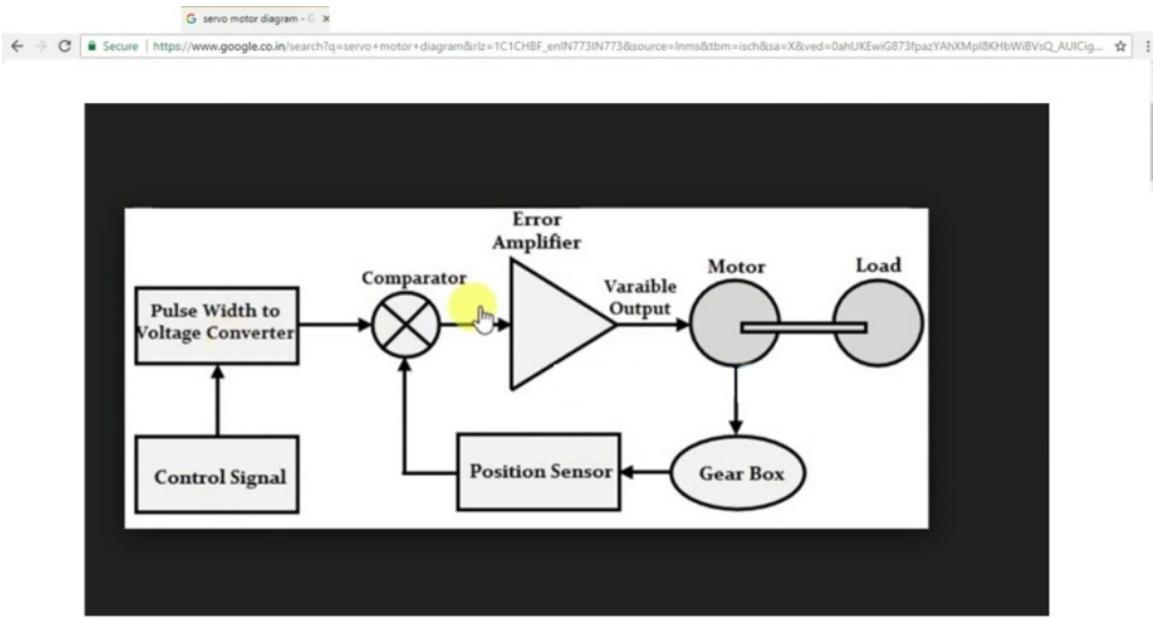
So the transaction controls the water of the transistor controls the direction of the current or flow of the current to give the rotation of the motor, but the motor of the thighs method is the motor can only work in one direction and to execute the Engine in the direction that you need to change the polarity of the chord engines and the thighs were the description of the CC engine if you want to see that I will give you the diagrams here is the diagram and here the connection to the transistor and the side of the thighs will be Connected to Arduino. The thighs side will be connected to the power supply and the thighs here are the engine that is the engine terminals. Okay and these are the basic theory of engine operation. Well, I think the thighs method is very useful for managing an Atlantis, but for that, we will talk in the last section or the 10th lower heavy engines by the thighs method.

## SERVO MOTOR



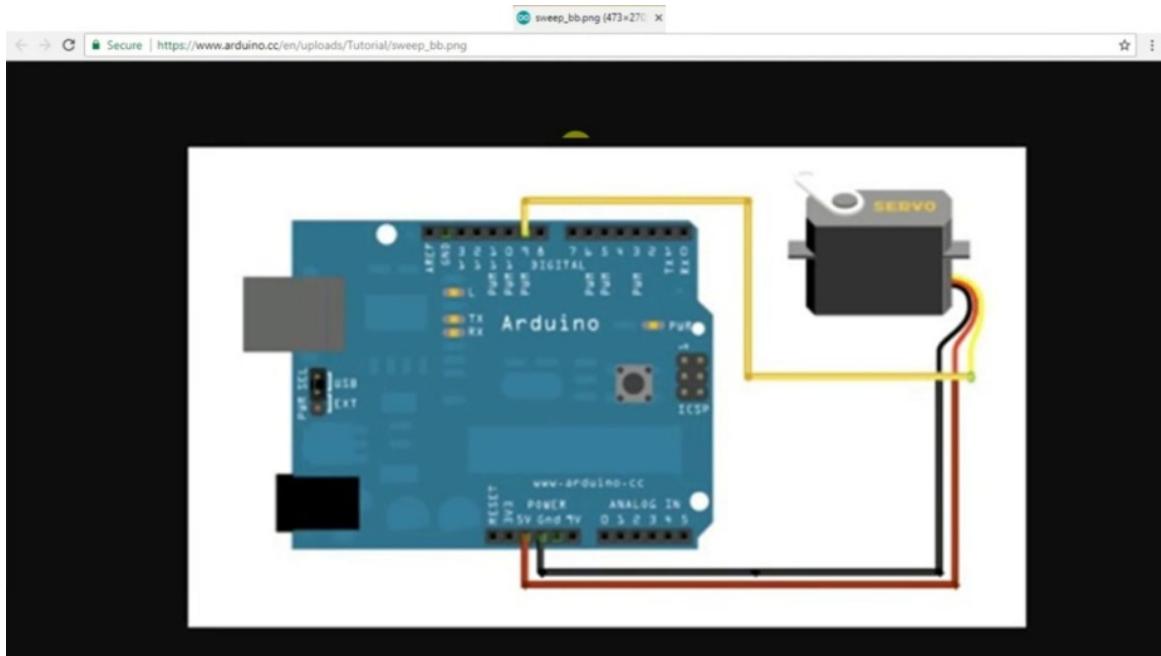
We will discuss the servomotor. Well, what is a servomotor, first of all, I want to tell you that the servo motor is not similar to a special does not look like a simple engine running in a progressive direction while on the other servo The engine is a kind of engine. This is fixed for a specified value, it can only turn from zero to 180 degrees and it is not possible to rotate it more than that in many servers? While some servers can rotate 360 degrees complete in continuous rotation. But the main advantages the main use, but the essence that the servomotor is used that the servo motor can define a specified degree, if we define a servomotor to more than 30 degrees, it locks to 30 degrees in any condition that will occur in Moses or any condition that occurs at the motor of the servomotor in any direction, the servomotor will refuse it and return to the position, it will return to the part that is set to the example of 30 degrees. . If you set a servo motor to move a direction to 10 degrees in each second, it can do it like that 123456 and up to 10. If we want to define it to five degrees and exactly five degrees precisely, it can be locked on the Five degrees. Well, there are many servo motor applications than the toy plan helps the reader control the results of the traders by servo motors. The reason because it is when the servomotor must be pressed in position and when the air flows on the wing, it tends to move the wing in the rear direction opposing the servo when the servomotor retrieves the version and that the values are clear. I have

a document on the internet, I received a document on the internet, which can help you get a brief knowledge of the servomotor, see here our servomotor There are many shapes and sizes some are industrial engines able to turn permanently to 360 degrees well, the servomotor has a basic design see



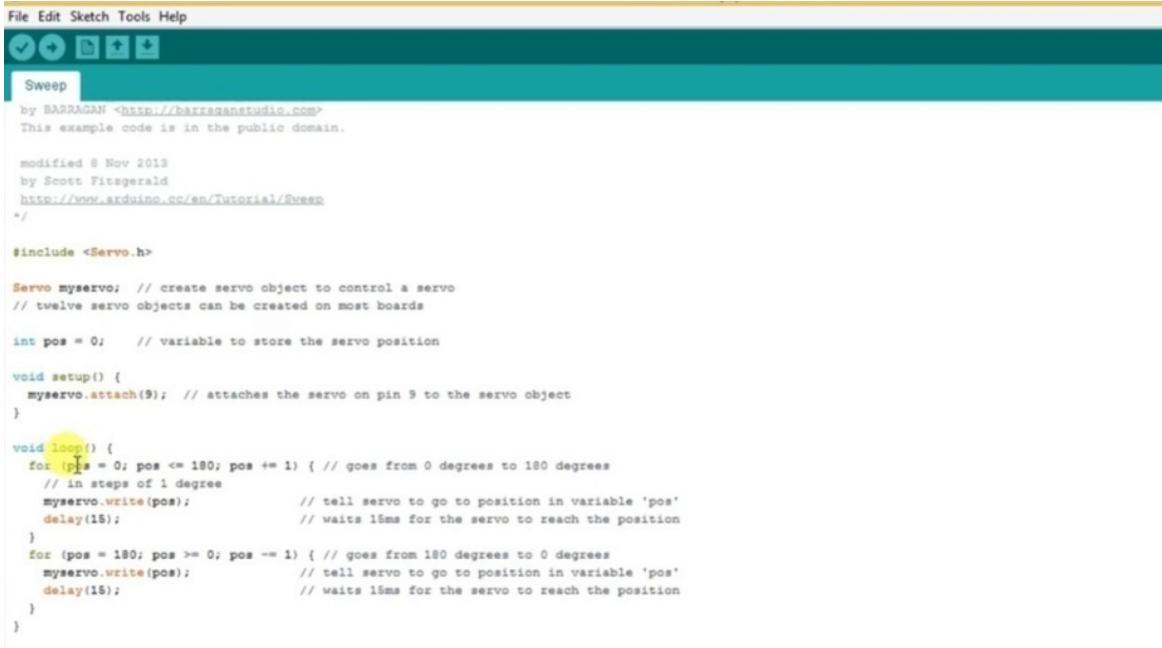
The design carefully, the thighs are the control signal or you can say that the thighs are the Arduino, the pulse wood converter we talked about is a DC. Now, coming here the cooking is the waitress part of the thighs part is a comparator of a small tip that is implanted in the servomotor and the thighs are the position sensor that you can say that a simple potentiometer cook is An internal amplifier part of the chip and the thighs are the motor. The thighs are the access of the motor when the Excel must rotate 30 degrees, it will compare its position here and it is indicated that I will position it as zero degrees so that it will give an error here and that the error will be amplified To define the rotation of the whole servomotor seeks to return to 30 degrees when the motor switches to 30 degrees, the data indicates that there is no error, he said he wanted 30 degrees and It is 30 degrees, there is no error. Thus, the value defined here as a burden or two more than 30 degrees has been recorded here by access and displacement of 30 degrees. Whenever a change is made on the server, it came back to its location, it becomes at the tip of the thighs,

but now we need to discuss how to connect a servo motor to an Arduino. See here, you can see that there is only one second there are three pins in the servomotor, they are as follows. The first is yellow that is for the signals that the red is for a five-volt output in the black is for a power voltage. Just see in the diagram I made a simplified version of it. Here you can see that the yellow wire is for the signal connected to the analogue pins of the Arduino.



The Five WorldSare are connected to the tray and the ground is connected to round pain, but can I need to mention something easy if possible, give the server or the individual power supply because if you connect the servomotor directly to Table and engine that has a size like the thighs will create large disturbances and a large amount of power possibly frying the card and can also cause damage to the computer to move continuously of a heavy torque servomotor and heavy motion, it is recommended to use an external power supply and connected to the ground and then send the signal from the Arduino for the movement of the stability of the tip of the servo motors and the protection of the servomotor and the mother. You can say that the Arduino table is fine, the thighs were the basic description of the servomotor in what follows we will use a servomotor and controlling it with Arduino with an example. About servo motors, I want to tell you that there is an integrated library in

Arduino that can control the servo motors. You can see here the file ranging from examples and go to the examples and go down, you can see that there are two programs built into the Servo Motor application. Here you can see that the default library of a servomotor is called the library that receives an object and the object is selected for all operations. Here, the object is said that the servomotor is attached to the pinning of the Arduino board, okay, the pinning will give the modulation line signal of the pulse width for the servo motor to rotate. And here you can see that the pin in Pinyon can say that putting an analogue spindle is called a pin. Just edit. You can see that the analogue spindle is called here, you can edit here you can see that, there you can see that the analogue spindle is called pine three and whenever the playback at the analogue spindle has changed playback to the servomotor will observe the analogue play or spindle and give a direction of movement and the angle that is here, you can and give a value that you can see that the value of the integer is defined by the function using the function card. Here you can see that in mathematics, the value of the digital the here you can see that the value of the analogue input is given as 02 1023 and the movement of the servomotor is zero degrees to 180 degrees here when the value of the potential meter, you can say that it is 255 or 555, it can take the servomotor at an angle of 90 degrees if the value of the potentiometers changes the angle of the servomotor, the angle of the servomotor also changes and the angle writing is performed. with the help of the function. My servo was the item and is done by the written dot function and the value is that we have valued the variable that is the value performed by the function of the thighs card. Thus, this can give you a value at which the movement of the servomotor must take place and the value, to which the servomotor must stop.



The screenshot shows the Arduino IDE interface with the title bar "File Edit Sketch Tools Help". Below the title bar is a toolbar with icons for file operations like Open, Save, and Print. The main window contains the code for the "Sweep" sketch. The code includes comments indicating it was modified on November 8, 2013, by Scott Fitzgerald, with a link to the original tutorial. The code defines a servo object and sets up pin 9 for the servo. It then enters a loop where the servo rotates from 0 to 180 degrees and back again, with a delay of 15 milliseconds between each degree.

```
File Edit Sketch Tools Help
Sweep
by BARRAGAN <http://barraganstudio.com>
This example code is in the public domain.

modified 8 Nov 2013
by Scott Fitzgerald
http://www.arduino.cc/en/Tutorial/Sweep
*/
#include <Servo.h>

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0; // variable to store the servo position

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
}
```

Okay, the thighs were on the ballistics of Sarong on another conference. Just talk about another servomotor program. Here you can see that the scanning is a much easier program because it used a servo directly and later to rotate 180 degrees to zero degrees and vice versa, it does not require a potentiometer for its operation, It is simply connected to the new spindle then it uses a loop for the variable position you can see that the variable is increased from zero to 180 degrees with an increase of a degree of 15 milliseconds if it reaches 180 degrees, then again when the position value increases at 180 degrees. Taken at zero with decrements of a degree. And all the values of the servomotor were written in the object to the servomotor by the correct function as a pose and the high-end have a difference of 15 milliseconds. Well, these are the basics of servomotor and servomotor programming on how to control it with Arduino. Well, the thighs were the structure of the servomotor we talked about here.

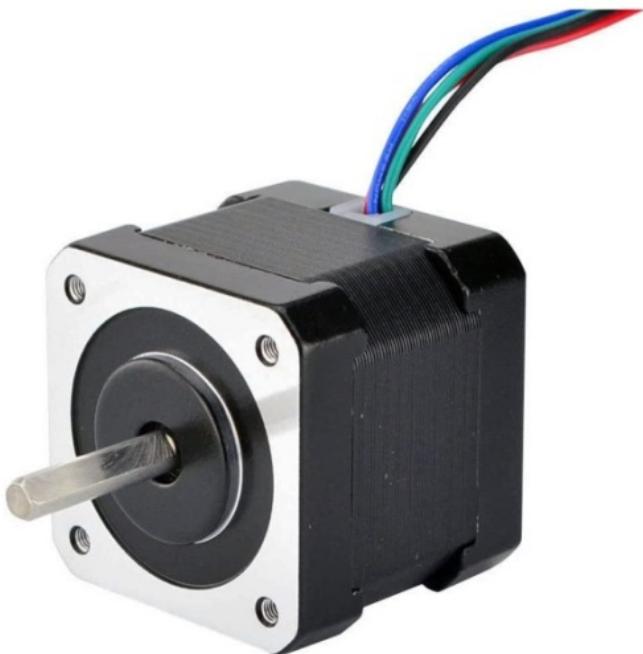


## Types of Servo Motor

Servo motors are classified into different types based on their application, such as AC servo motor, DC servo motor, brushless DC servo motor, positional rotation, continuous rotation and linear servo motor etc.

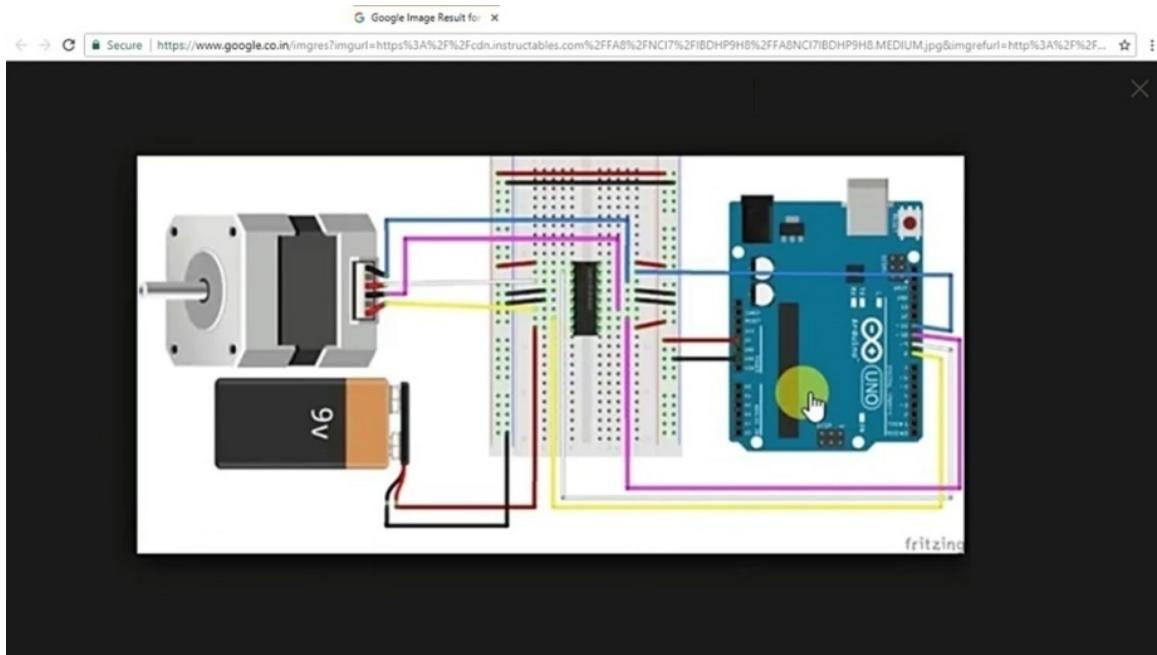
It uses a simpler engine and there are some electronic devices behind a potentiometer and these are the gear timer hearing a potentiometer is connected that gives the value to which the servo how to stop or to which the servo how to reach The servo. By the value employee, the value correction was carried out, the correction of the value is performed by the pre-integrated comparator in the servomotor and the value of the signal input is taken with the help of Arduino so that the Arduino can control the servomotor to reach a specific degree. These are the different types of servomotor and you can see here an AC servomotor constitutes an industrial servo motor, which has a petition positioning encoder rather than a potentiometer to give very precise values so that it can work very precisely here. A similar servomotor that we have taught is called continuous servomotor because it can turn on the Indian rotor continuously 360 degrees several times, you can see it here. You can see here that it can rotate continuously. These are the spatial servo motors that do not have the spindle that does not have the physical part that prevents them from turning permanently and that they are a potentiometer are of good quality so that they do not deteriorate Potentiometer will run continuously 360 degrees.

# STEPPER MOTOR



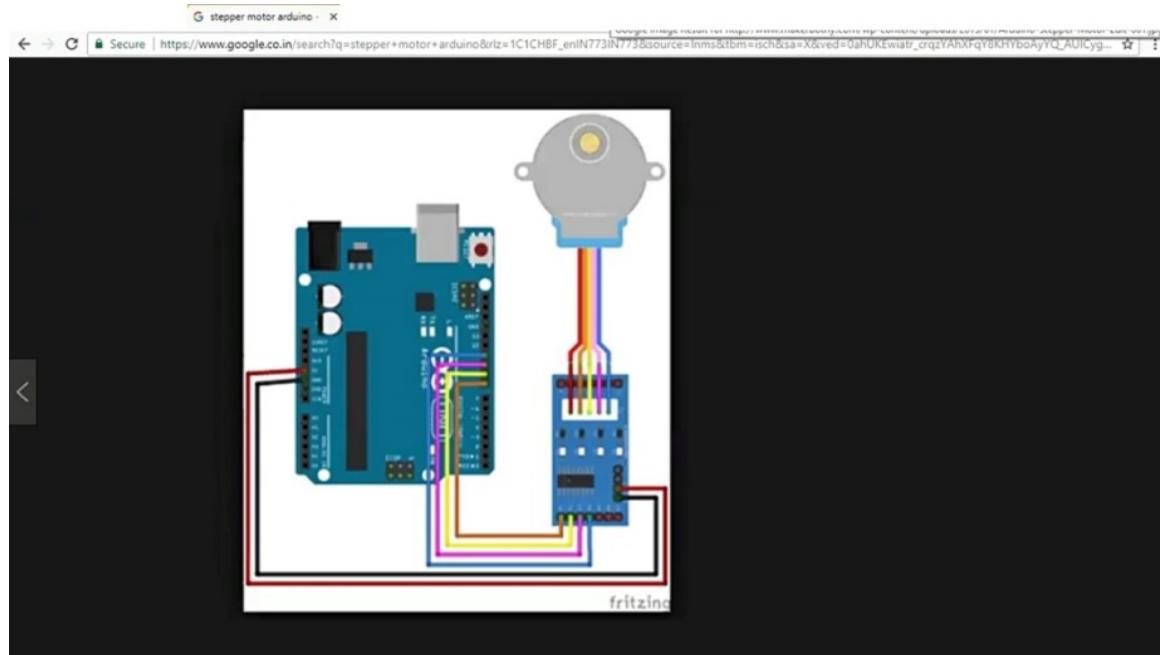
We will talk about step engines. Well, a stepper motor is a special computer engine used to operate at specific degrees as well as for 360 degrees. The thighs are the most common motor used in CNC and is used in 3D printers in two-dimensional events. These are usually heavy industrial engines that can perform all loads and can rotate in specified, you can set rotations or say some of 20 degrees, it can also move to the micro-step so that it can move to A 10th or one to 200 in many spatial engines of a degree, you can say that a stepper motor can usually pivot about 200 steps out of 360, which is about 1 8 degrees, but if you use The micro stripping method to control the servo motor, it can move about one to 200 the part of a degree that can give a very brilliant clarification of the servomotor and can give very precise results. Agree on the Stepper Stepper engine on the stepper motor, the stepper motor goes out step by step between extraterrestrials. They are the unipolar engine such as thighs and their bipolar engine such as thighs. There is only the difference in the wiring in you can say that they are a bipolar motor like the thighs OH four wires while a unipolar engine has five wires. These are the difference in polarities of the coils inserted in the

servomotor for the movement of the direction, the direction and the movement. And there is also a hybrid engine that has six wires, they can execute both unipolar as well as bipolar. Okay, let's just see a diagram so we can have a clearer understanding of what is a stepper motor. First of all, let's start to start here, you can see that the red part is in a magnet in the 123 and the four, there are numbers, they are the cause that activates the active thighs by getting the value of the computer. and to be amplified by the driver.



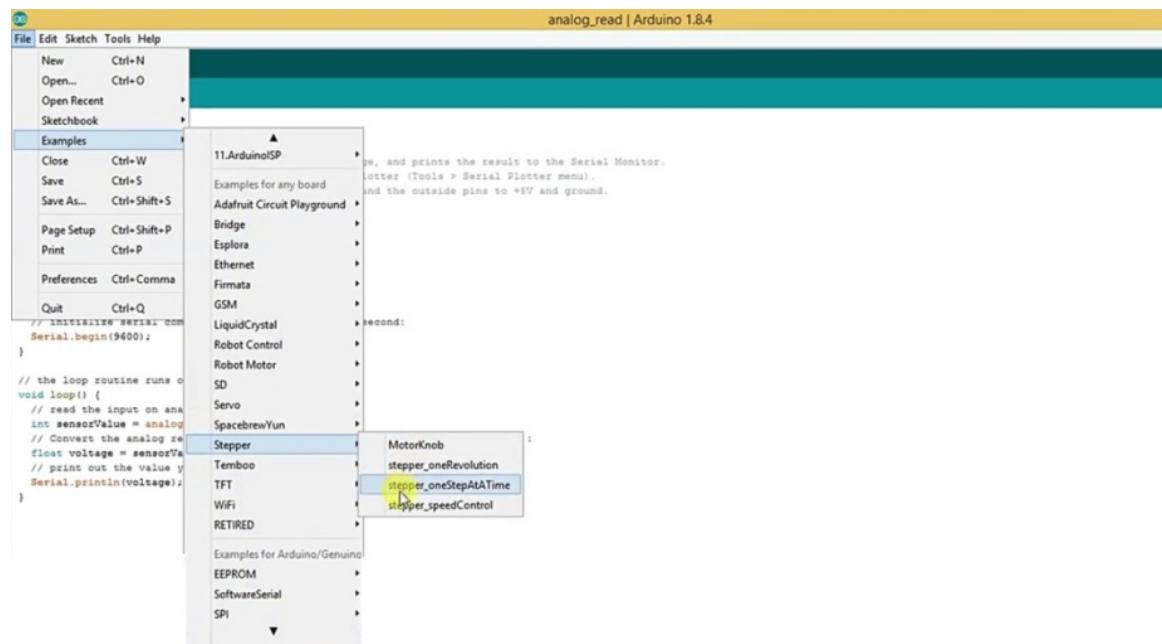
It is therefore possible to create a motion on the dry glow magnetic drive or that you can see that the step by step is made up of a magnet to give it to the computer motion, the thighs are the method of Control of a stepper motor. You can see that here in the diagram whenever the pulsed clocks of the evolution of the matrix are called in incrementation of the direction of 1234, it gives a low movement value to the rotor and it goes so. Continuing like the thighs in motion, it can give a continuous movement, but a very precise value, if you use magnets that drop the value, you can be precise even better servomotor and motor bouquet with no bouquet, there is also a special motor type not known as the BLDC that we have discussed previously, there is the same work principle in Motor CC The main difference is the outer coil is replaced by permanent magnets, while the internal core is consisting of pulsed magnetic coils and are

controlled by high-speed speed or you can say an electronic microcontroller speed controller and they are all 32-bit t Okyo. High performance and high speed of values so that they are used in equalicopteria following wiring. Let me give you an additional sample let me give you an example of non-unipolar steps like the bipolar thighs and thighs like the thighs. You can see that the thigh engine is bipolar because it has four cables passing through the driver here, they use a single chip here for the driver and driver or you can simply signal amplification, amplification power eating nine Volts. The battery in the signal we are going to run a Giant Engine by Arduino okay. While the function of the thighs is very similar to that used in a un polar motor, it has a driver and you can see that five wires coming out of the un polar motor and the power source. Here, the power source is used by the Arduino, but for better performance, the power source must be independent and protecting the engine of the attempt. Okay, you can see that there is a fundamental difference between the romanesco, bipolar and unipolar engine is that the voting drum on different types of codes comprising is different and have a very similar motion that they can to be used for both the micro-floor but their source code and their path of algorithm and programming

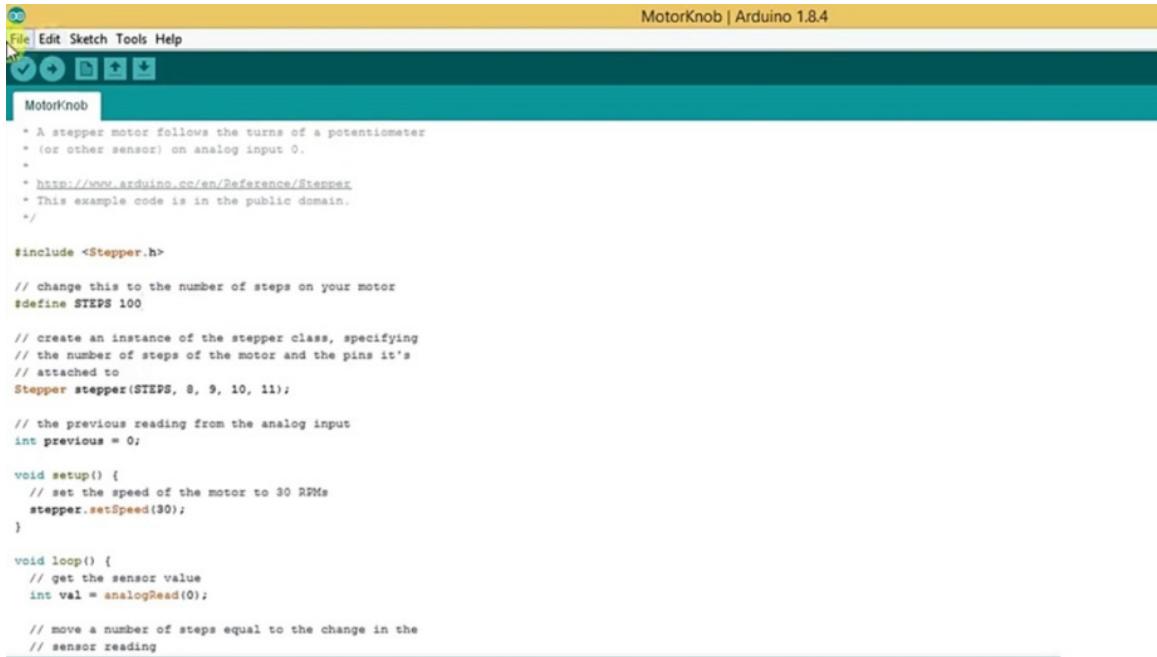


is very different from the other. They mainly talk about the bipolar engine, especially we are talking about bipolar engines because the bipolar motor is the most common type of motor we enter into an interface in

Creating an interface and to get an operation in the new next chapter, we will talk about wiring a step-by-step engine and use with a driver to facilitate the amplified signal in servo learning in progress. 'execution and execute the engine not nearby in future chapters. Okay, the library of the motorcycle at step is already preinstalled in Arduino.



The Arduino has built-in libraries for the movement of the servomotor as well as for the big book to see them here when you produce an example, you can see when you go see the design of the library for good here, you can see this engine. Norm A revolution one step at a time in the speed controller has just seen them..



The screenshot shows the Arduino IDE interface with the title bar "MotorKnob | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main code area is titled "MotorKnob" and contains the following C++ code:

```
* A stepper motor follows the turns of a potentiometer
* (or other sensor) on analog input 0.
*
* http://www.arduino.cc/en/Reference/Stepper
* This example code is in the public domain.
*/
#include <Stepper.h>

// change this to the number of steps on your motor
#define STEPS 100

// create an instance of the stepper class, specifying
// the number of steps of the motor and the pins it's
// attached to
Stepper stepper(STEPS, 8, 9, 10, 11);

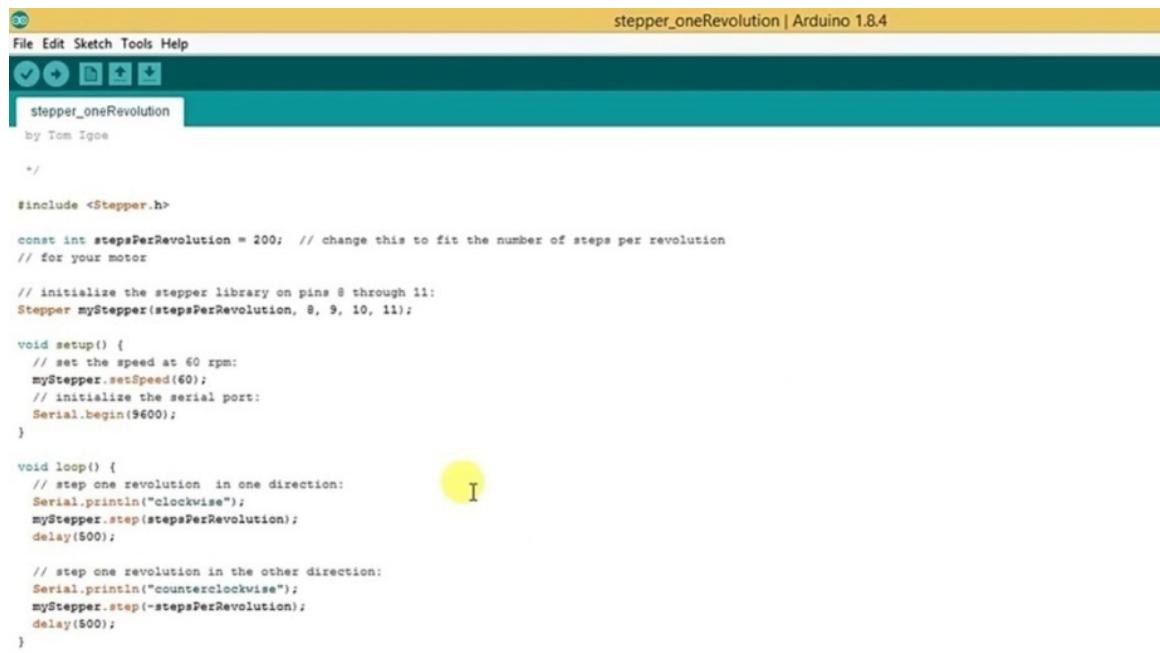
// the previous reading from the analog input
int previous = 0;

void setup() {
    // set the speed of the motor to 30 RPMs
    stepper.setSpeed(30);
}

void loop() {
    // get the sensor value
    int val = analogRead(0);

    // move a number of steps equal to the change in the
    // sensor reading
    stepper.step((val - previous) * 2);
}
```

First, the motor node is a library used to run the engine. Or you can say that it simply defines the pinin that the Arduino can be connected, let's say eight 910 11 and the variable is the steps. The steps are the variable that slows the number of steps to be taken by the servomotor to achieve a specified value. Here, these are 200 steps, it means that the engine runs from zero to 180 degrees, you can manually change the steps of P 60. or what you want. Let's see another program now. And still, go to Stanford, you can tell a revolution.



The screenshot shows the Arduino IDE interface with the title bar "stepper\_oneRevolution | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main workspace displays the following C++ code:

```
File Edit Sketch Tools Help
stepper_oneRevolution
by Tom Igoe

/*
#include <Stepper.h>

const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution
// for your motor

// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

void setup() {
  // set the speed at 60 rpm:
  myStepper.setSpeed(60);
  // initialize the serial port:
  Serial.begin(9600);
}

void loop() {
  // step one revolution in one direction:
  Serial.println("clockwise");
  myStepper.step(stepsPerRevolution);
  delay(500);

  // step one revolution in the other direction:
  Serial.println("counterclockwise");
  myStepper.step(-stepsPerRevolution);
  delay(500);
}
```

A revolution is a small program of the thighs we simply say that the engine of the motor pins that are connected to the night 910 11 we must all route them, every 200 steps to follow to obtain a movement of complete revolution and return to its initial position. When it has been started, the movement can be tamed or can be controlled by controlling the speed in the thighs, we can see that the motion is that every time difference of step has 500 milliseconds or half-seconds, you can also write them under 50 milliseconds. So the engine will work about 10 times faster than that, but you need to change the time direction of the card, you may have to change the edge time value of the variable so that it can give a clear rotation. If we modify the value for one, it means that the engine will have an evolution injured in a second and an outside gamble that it will rotate 10 times faster to get an initial version so that it is better to change the value of the two locations so. that you can have a gentle movement in both directions of another legacy that other parts are a motor step at the same time the thighs is a program mainly used for taking the step and is also used to obtain The precise value of the stepper motor.

The screenshot shows the Arduino IDE interface with the sketch titled "stepper\_oneStepAtATime". The code is as follows:

```
File Edit Sketch Tools Help
stepper_oneStepAtATime | Arduino 1.8.4
stepper_oneStepAtATime
if you don't know it. Then plug that number into the oneRevolution
example to see if you got it right.

Created 30 Nov. 2009
by Tom Igoe

/*
#include <Stepper.h>

const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution
// for your motor

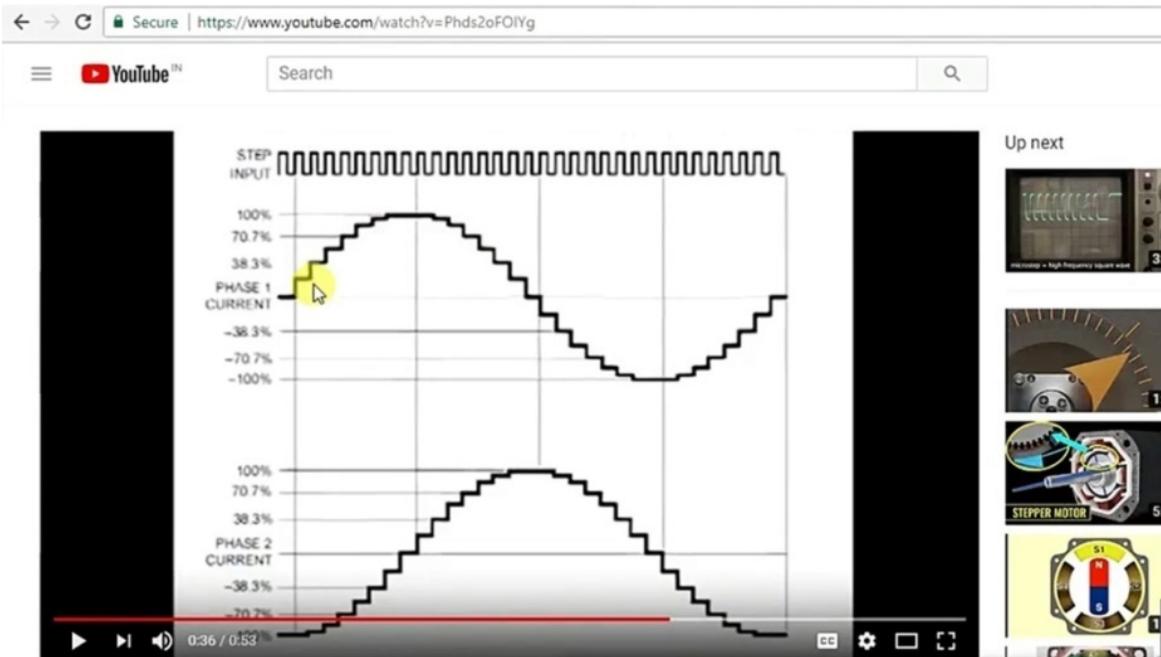
// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

int stepCount = 0; // number of steps the motor has taken

void setup() {
  // initialize the serial port:
  Serial.begin(9600);
}

void loop() {
  // step one step:
  myStepper.step(1);
  Serial.print("steps:");
  Serial.println(stepCount);
  stepCount++;
  delay(500);
}
```

The thighs program is used to get the step precisely unnecessary that you can get the wider interval that you can get this by the interval of a step, you can see that you can the angle that the stepper motor built from the previous one to the current location. . In reality, it is 18 degrees but sometimes it can also change due to the micro-stage you can change 1.8 degrees to 200 times more clearly or you can tell it that you can divide it into 200 pieces to get it A better-clarified value. Well, the thighs method is used for controlling motors used in 3D printers or digital manifestations because they require a high level of accuracy and higher accuracy for a high end for the level of High accuracy. The degree of stub must be very precise. So, they use micro-floors, we will talk about micro-stubbies afterwards, but let me give you a small overview of this step can be done in three parts. Let me show you the difference, okay, you can say

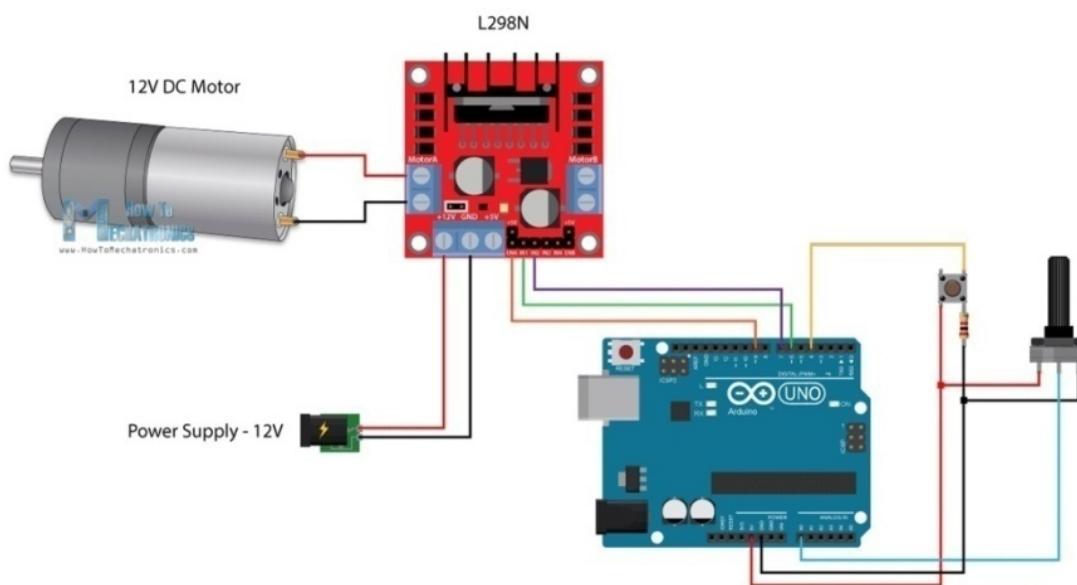


that it is identical to the modulation of the pulse width if the pulse is false, it can make a degree, but if the pulse is very slow or if you can say that the service cycle is very small, it Will give the value to the engine node of a degree a node of a point, it will not give the motor to move 1. 8 degrees, but it can also control it, engine half a 1.8 degrees to a quarter of a quarter 8 degrees that anyone 200 the part of a two and eight degrees for this process, the micro steam is used. I will show a diagram for one second here you can see that the example of the micro-stage thighs is a complete cycle of rights that has been divided into four basic steps so that you can say that everyone has a difference or 25% of the service cycle. But the thighs can also be broken into small parts here, you can see that the thighs are the method that allows you to see that each part is broken in our seven segments or so, that it can give a value From one of the four intootantven who will take you more than 28 by the value of the stage that should be done so that it can give a very precise step to take. The whole cycle is 1 8 degrees present, it can give 1 8 of 28%. At present, this can give 1 8 divided by 28 and it is the angle. The micro-stage is the most accurate step that is most accurate step can be taken using a specified driver. But remember a micro-floor thing, but remember a micro-floor thing, it's still to use a 32-bit card and a 32-bit driver to give a verifiable use in the industrial force, in

industrial equipment. , All step-by-step engines are controlled with a 32-bit arm architecture microcontroller and 32-bit driver. So that they can give a very good precise value and they can operate at very high speed. Okay, the thighs were the theory on the motor step by step here, you can see the Arduino the wiring cooking driver, the external battery source and the stepper motor here. We talked about the program in the chapter we are going to perform an engine, not for practical purposes.

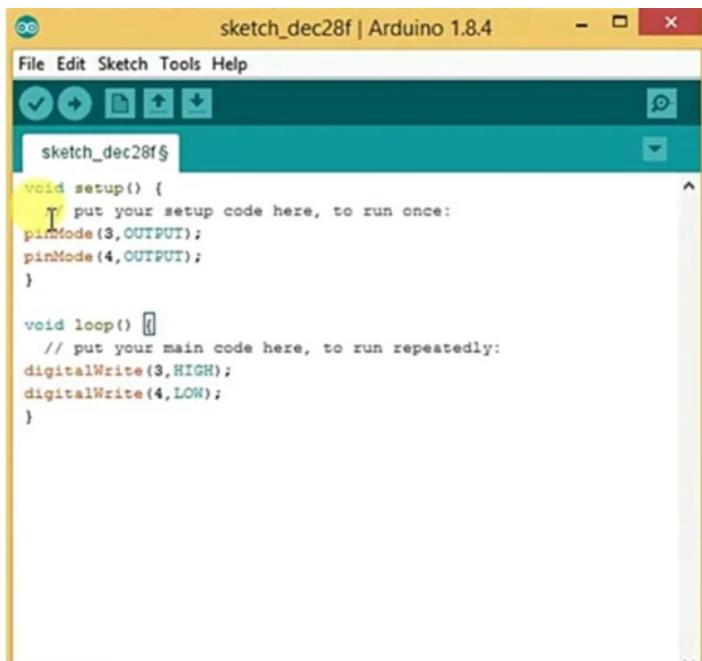
## RUNNING A DC MOTOR

We will talk about operating a DC motor with the help of an Arduino card and driver, we can not run a DC motor directly with the Arduino because the DC motor attracts a large amount of current in The current thighs can easily fry the Arduino card even in your computer or laptop.



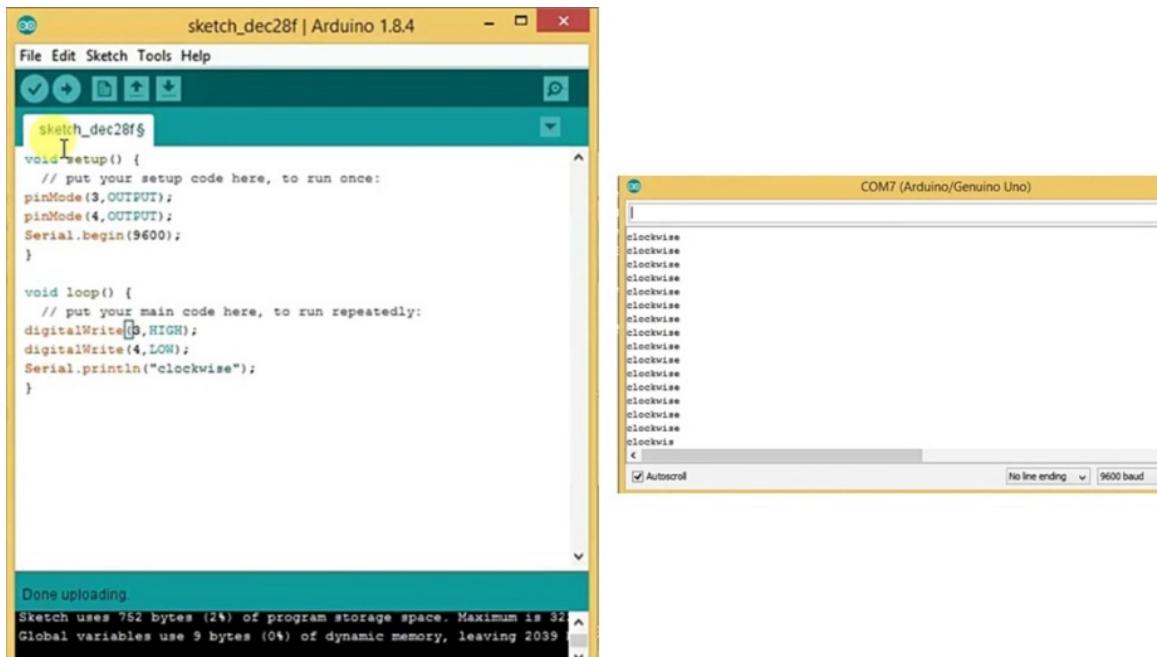
So be careful while connecting the DC motor to the Arduino because there should be no direct connection or there should be no leakage voltage in the world. So there should be no leakage voltage or current in the water which can directly go to the computer or the Arduino because it can burn it severely and cause you a lot of

damage. Okay, so protecting the Arduino from the motor in the car having good control of the motor we need to Connect it to a driver. A driver is specially designed as a machine used to perform the Motorvia controlled function using small signals of small digital signals provided by the Arduino and amplifying the signal with integrated voltage sources here, we are connected here. Connecting it here, we connect it to a 12-volt source so that the driver can amplify the signal in the control signal and for the control signal that we connect with Arduino and the output of the amplified signal and the control signal. is in progress. Engine For that, we can select the direction of movement as far as we will sing the speed variation of the motor have selected the analogue pin for motion control and insert in a weak so that we can have a clear vision of the motor functioning correctly, Let me tell you about the diagram. Here is the Arduino that comes out with three themes of three out of three, it goes out in the form of three things. The first thing is the analogue ping that is used to control the speed of the second and third pins from other deleted pins connected to the retail pins four and three in Arduino and are connected to the next loop and are connected to the following two pins on the following. driver. The first two brackets are connected to 12 volts, the motor is connected to the output terminal, we can change the direction of the motor as needed by modifying the value of the high to the low spindle. We can not keep both the high pin or if we look at it because it will create a potential difference of zero between them, which will drive the engine rotation in any other direction. To move an engine, we can set a high value and another bass to move in a direction. And we can change it as a larger low value and another high to move it in another direction. Okay, let's just practice a practice



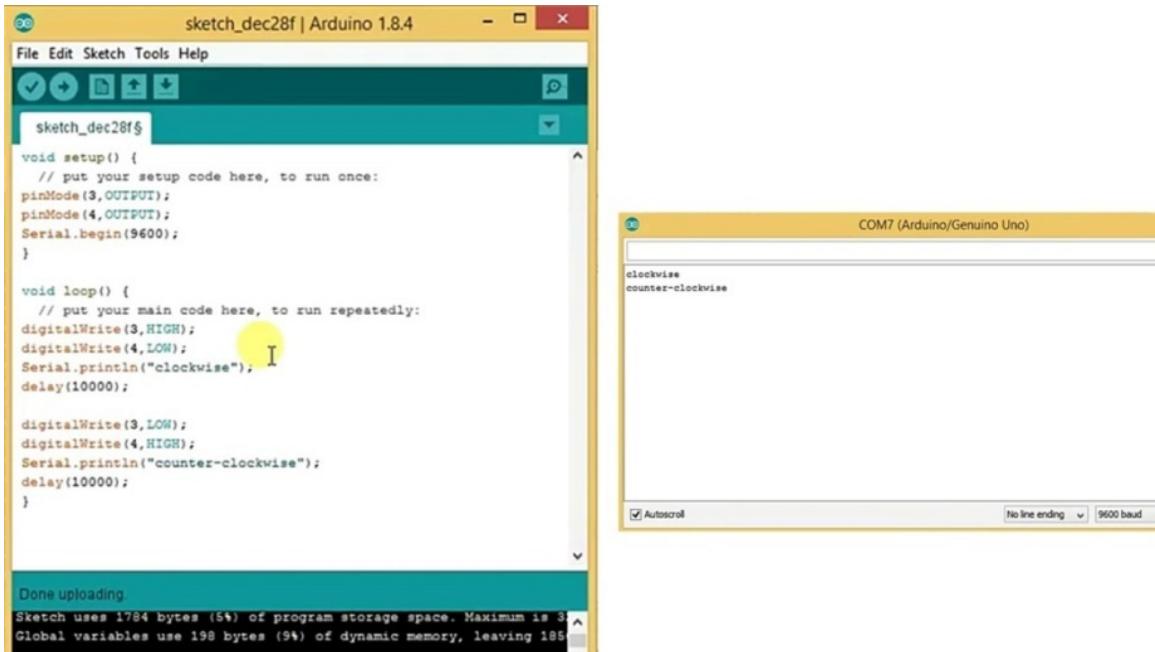
Open Arduino opens a new sketch and here. First of all, we must declare the pine code right that we say at the exit of the Pin three comma. We made the pine code too wiri pine modem number who outputs like the output because I connected it to this pain if you connected it to another basket please be careful by writing the pines d 'agreement. Now, Come to Word Bas Low Write a simple program Tree Tree Digital Eritinion Number Three virtue Hi Gu Doueur that we do the number of the number one Besoy Okay right in the Suite IT Lwr IT lodge detail on the plate number that Komal Or w, okay, just take the program in more than three and four, I declare that our port the port is connected to PIN10 for more details to the right of the peanut three high painful and the engine is the wiring to which You will check the wiring, because it will seriously damage the microcontroller if incorrect wiring is carried out at high voltages agree, just the compilation we compile well, it is enough to cancel here that it compiles here that the Compilation waited for a second, okay, there is no compilation error to download and download global values, okay, the world is downloaded. And you can see that the engine runs in a direction. If you want to change the direction of the movement, please go here. Say the spindle three as well, we experience both three and four our low status do not check that the engine can turn on both wheels that you can see that the engine has

stopped working now remains the law on the Thighs and number 10 again, download it around the world. You can now see that the engine is towed now, you can see that the engine runs in another direction. Well, the thighs are the thigh motor, the thighs are the control process of the motor direction, but if you want to see it on the screen, simply write a simple program. First of all, I have to stop the engine so that it will not create any disturbance in our programming had to stop the engine. Now, if you want to see the direction of the engine, it's good for us to get a clear view by using it to get the direction and the right series monitor is E Ri L Dot Veg in Dot Series Start at 9600 for Baud rate, we gave it 9600 for this function. In a simple Mega statement, you can say that the thighs are an animal form of the declaration. Ok, let's just. He takes a second. Wait, stop.



Okay, return to programming here. SPI TIME SERIES SPI L DOT P R P I N D LM for the following line. Okay. The right is blocked less okay, we are over who says one thing too. Top, I'm sitting with the number three. Now, download it to the world. Cancel this, I cancel it. You can save it as desired. Now see the direction of the engine? I think it's clockwise. It goes in the direction of the needles of a watch. But if we want to rotate it for 10 seconds clockwise and 10 seconds in the opposite direction of the needles of a watch? Well, there is a method for that. First, let me stop the engine. I will show you the

method that I stop the engine takes time and stops that there is a very simple method to rotate the engine in both directions only come here if we want to rotate the engine in a direction for 10 seconds. The wide delay program is trying again. OK, stop it. Now, come here right di JT Lwr to the digital writer in number three, SL or like L or W on Talton Digital, right? Read our Idva Big idea my retail error? For coma, it gives high,



Come here you can see that the thighs are high for that moment for 10 seconds and another direction for another 10 seconds is our daughter PTL N Print LN you can tell the counter-clock Wiseor counterclockwise finish the line and Write the day of 10 seconds 12344 agreements ok. Oh, now give I give them water for your understanding of the loading of the program. And let's say with a DC motor. Okay, so in the program, you can download all programs at the start of the module in your presence that you can get the program okay, it takes a while. Okay, simply opens the serial monitor. Let me open it once again. Now you can see clockwise for 10 seconds. It works continuously, wait for a second to come. And you can see that it turns in another direction. Well, by the thigh method, you can control the engine in both directions and you can get a keen sense of clockwise and counterclockwise by changing the polarity.

# RUNNING A SERVO MOTOR

Well, a servomotor is a special type of electric motor used to give a precise degree of value. It is not used to give a continuous rotation well to connect a servo motor to an Arduino, it is very simple that you can do it by connecting the yellow wire to signal the red to five volts. You can connect it with something yellow wire to report or that you can see the pin that we have set to control the servo-five volts wire will be connected to the red wire will connect an output of two to five volts in the ground wire. to be connected. at the land terminal of the Arduino. Yes, it is good to run small engines, but if we use weak servos, it is necessary to display them with an individual power supply. Okay, let's start now with the completion of a core. Okay, come here. Yeah is a code that is a library is always a good library for the control of the Arduino control controller, but we want to write our code? So we can first go here at the top, especially we call the library I we call the library by writing Include I NC Read will change the colour that means that the servo is resistant to servo dot. Hs is the extension of file C okay. Now we are the rights e r V also a line will have my m so, or you can just say that my it's good. We give high my agreement. Now, in the word configuration, we need to define the positions that should be in bulk or vo go to the top, we must declare an integer for the preservation of the value of the angle that must be taken in the stage we declare That I declare I nt in gl in gl is not possible, so simply say d, for example, for example, it means that the degree of agreement is equal to zero. Now, come here says M while it's the good Girl Empire has TT SC, H is that we define the server number p number three. We, therefore, call why not attach the number three PIN connected to the service. Now it's very easy to control the engine for 32 at a fixed or 30-degree location, simply write my adult DR WR I TMY right straight 30 I will use are variable, now, I'm just trying to do you like SIM possible. Okay, the last program is running. Juice Thorold for a second but the symbols will be removed from here. We are declaring. Now downloaded. Here we must define my setter before, okay, come, cut it here. Okay, simply download the monist demonstration. Let's go back. This is the compilation of a second. Wait just a second. It will

be ok. And it is that S the Servo was taken the position of the legislature at 30 degrees for TD



The screenshot shows the Arduino IDE interface with the title bar "sketch\_dec28g | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, refresh, and other functions. The main code editor window contains the following C++ code:

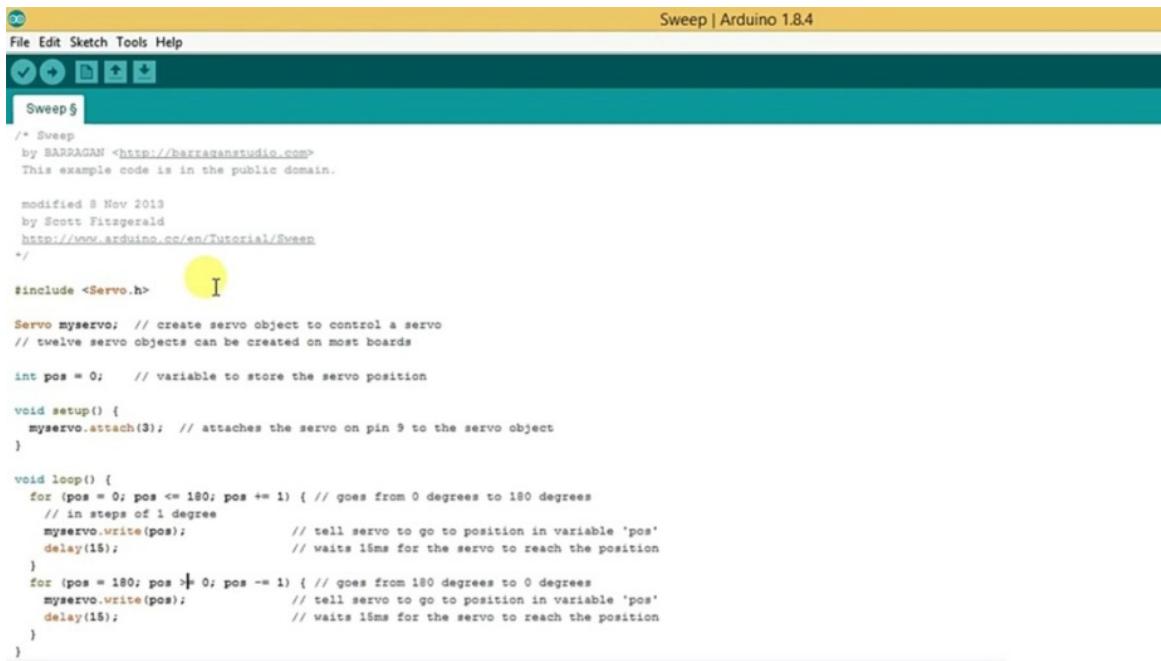
```
#include <Servo.h>
int deg = 180;
Servo my;
void setup() {
    // put your setup code here, to run once:
    my.attach(3);
}

void loop() {
    // put your main code here, to run repeatedly:
    my.write(deg);
}
```

At the bottom of the code editor, a status bar displays "Done uploading." and "Sketch uses 1490 bytes (4%) of program storage space. Maximum is 32256. Global variables use 50 bytes (2%) of dynamic memory, leaving 1998".

Okay, it changes the value now register to 19 NO Tit's ads of a Kindle edition. She simply says that yes of understanding, simply say the amount of which is the last value that can be defined using the servo. Okay, but what would happen if we tried to move, it is not possible to move it because it is locked at the position to accept what if we want to see the servo movement very slowly, you can simply Go to open files from an open example to Servomotor See where Select the Penis Three code, four minutes you can select an envelope in which you are connected to Arduino. First, let me explain the code, we called the responsibility, then we defined servo objects like a servo. Then we defined a position for that and it is initialized with zero. The next master was not attached, he states that the spindle took a signal that is doll three for my case and in the word of the loop. We wrote a program folder for the position that corresponds to zero less than 180 degrees should increase by a step of 50 milliseconds. The step that is implemented must be written because the version of the servomotor engine at the same extent when the person reaches 180 degrees and is greater than zero, it will then remain with every 10 milliseconds and the version will be written in

the servo. The engine as the angle. Let us download it to the world, it is a second. Now you can see the movement of the servomotor.



The screenshot shows the Arduino IDE interface with the title bar "Sweep | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations. The main area displays the C++ code for the "Sweep" sketch. The code initializes a servo on pin 9, attaches it to the servo object, and then enters a loop where it rotates the servo from 0 to 180 degrees and back again, pausing for 15ms between each degree. A yellow circle highlights the line "Servo myservo; // create servo object to control a servo".

```
/* Sweep
by BARRAGAN <http://barraganstudio.com>
This example code is in the public domain.

modified 8 Nov 2013
by Scott Fitzgerald
http://www.arduino.cc/en/Tutorial/Sweep
*/
#include <Servo.h>

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

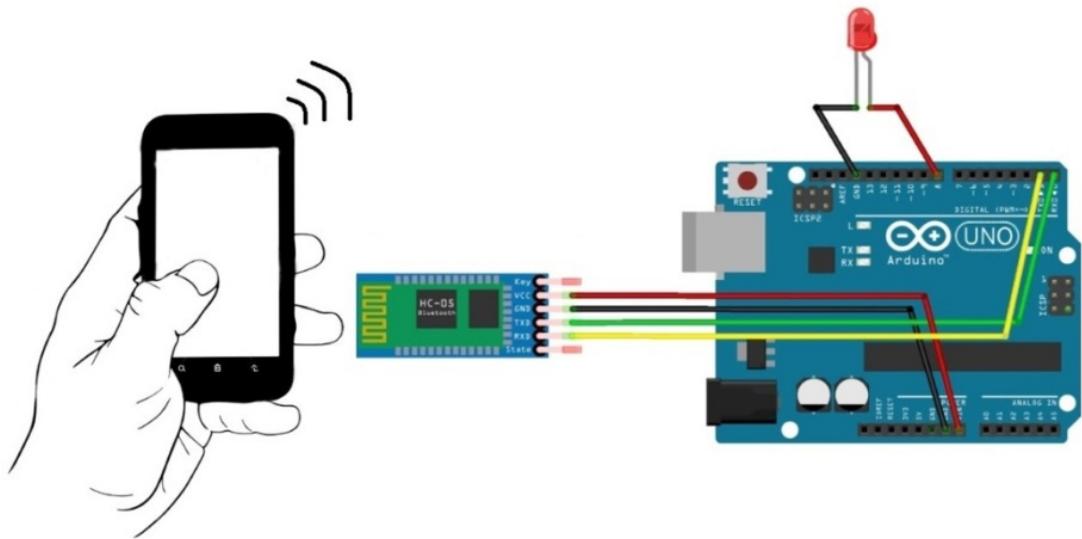
int pos = 0; // variable to store the servo position

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
}
```

Its thighs occurring are the motor of the thighs is the control method of a servomotor that you can write it manually that you can call a program, give it a logical function or if you wish, you can simply test it, you can simply take the servo. Engine by the cooking motor on the control of the servo motors. Keep in mind to run heavy servos it is better to give them a personal power supply connecting the ground power supply also on the microcontroller plane, then use them if you connect a servo directly to the microcontroller, it can damage the Microcontroller. as components.

## ARDUINO BLUETOOTH



There

are only a few connections to do. So, let's start. As you can see in this Arduino table, we have these plastic socks with numbers next to each socket. OKAY. Now, these plastic jacks have several Dunbar leaders and we know the bins that the Arduino microcontroller used to interface with the outer wall and these pins, we connect to Bluetooth a lot and the resistance and resistance for tests. Other bands. Benzédrine was a point x and t spend these two people as you can see all X X 0 and 1. They are angry and one of these will go to X or X and Bluetooth will make every 5 volts and Incredible Arduino Benn's will provide VCC or the power of this Bluetooth motherboard that is here. The hints which are the soil will offer the ground to the Bluetooth module as you can see here on this black cloud will get the ground from the one we know and we will have this sort when you receive 5 volts on what you will of Arduino. So, these four songs are the connection to the Bluetooth that you want. So we are managing and running. Then we connect Negative Alid on the ground here because you can see why the positive fold 15 with a resistance value between two to a child. So, it will burn you if we connect directly 5 volts will go through this advance. Why it only works at 1.5 to 2 and 5 volts. We must therefore need to store to minimize the trail volts. Then we are good to leave. You have finished with this circuit, but please do not connect our thinking x x and x to x as you can see. You must connect our X we know to t X.

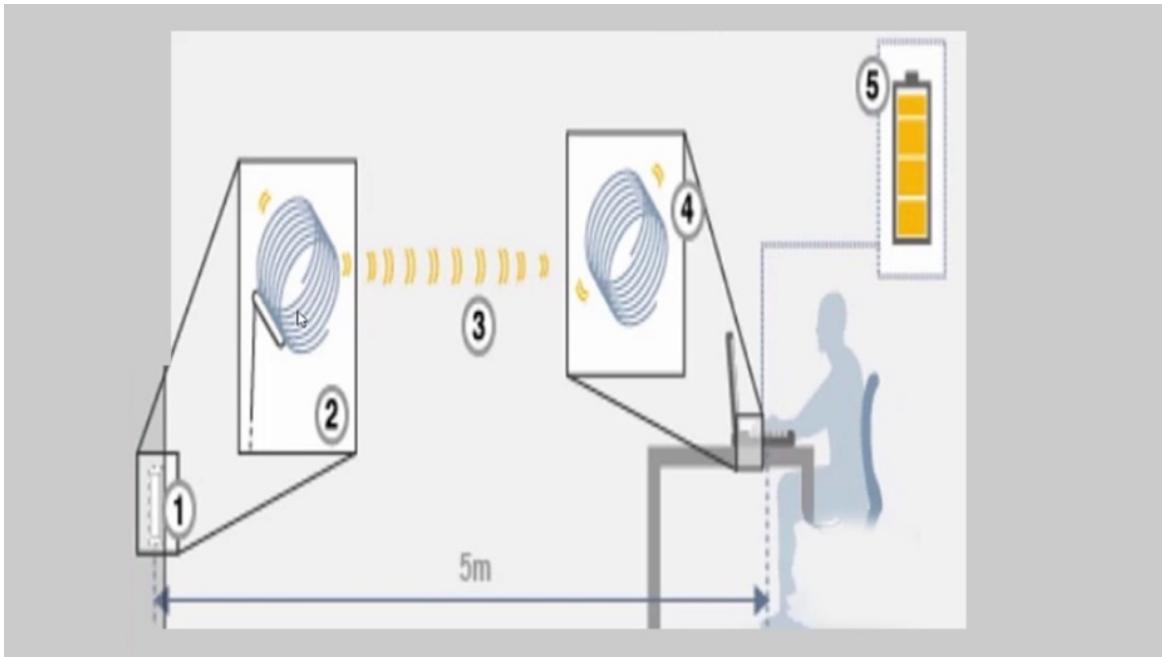
Since RX means that it is evolved out of the arduous or a c or c of the signal of this transmitter of a little with these tonnes for the transmitter and vice versa the transmitter and the Arduino send data to the truth receptor. So, each will be connected to the other not the same as all the other watches are here at T and T to be turned off so as not to connect neither x nor x anti x 3 x nor with two or three no, you will not receive Data here The X means transmitting and RX means receive.

## **INTRODUCTION AND HISTORY OF WIRELESS POWER**

Thus, while this board deletes this connection to feel wireless. This means that energy transmission through the air is the process of transforming electrical energy or wireless distance Bhagwat. Also known as the thirsty transmission resource and wireless power that is a transmitter provides a device to the device and the device captures the recipient. Eliminate the use of cables for power transmission. This is one of the most sought-after benefits of wireless power. We do not need to use the cable again. We want to get rid of firearms. You can see many and many smartphones competing with building or activating duck fill that will make users or the life of end-users a lot. So, in this course, we will know how this bullet is made in the phone or using Arduino or transmitting your controller card with an induction resonator of the impression. We will explain that the buoy later. But for the moment, that's what you need to know about wireless power. Now, let's talk about some stories about it in 1899, Nikola Tesla is a fly of this law in Litan 200 lights of 40 kilometres but only 15% efficiency. As in active mode, most of the water wasted and less efficient. This is one of the first reborn wireless transmission trials. In 2007, a group of engineers at MIT came or offered the idea of using a resonance induction to transmit wirelessly. Id Buy a 60-watt bulb of two meters or seven feet with 40% efficiency. This is a tubercle or light bulb that they can see wireless as you can see. Here we have a coil. And here we have a

coil. This is the first practical example of interesting. This is a globe that ends this task. Now, before ending this conference, let's talk about the graph of the footrest effectiveness. We all say that the Almighty you have reached 40% of temporary efficiency and Nikola Tesla has reached only 15% efficiency. It is, therefore, a graph of the effectiveness of what is travelled or electricity as you see here and we have effectiveness. And here we have a distance. You will not see only when the distance increases the decline inefficiency. So, if in meters, we have 75 centimetres, we will circumvent an efficiency of 95%, but when we continue when we reach 225 centimetres less or 225 centimetres, we know that we can only get 40 or less than 40 % of the efficiency you can see here. This black box means that it is the result of experiences. While red dots mean it's an experience but not made for various boxes that mean it's his theory. As you can see this line, the blue line is the theoretical line as you see the results. As you can see the main point here is that when you increase the distance, the afternoon will diminish significantly until it reaches zero. The highest distance you can achieve

## HOW WIRELESS POWER WORKS



First, you need to see this diagram or system that will help us understand how power or wireless work at this point. You can see a magnetic coil or an antenna placed in a box and can be set to the wall or ceiling. The second component here is the carton antenna with resonance at certain frequencies so that it is a coil with a certain frequency. The resonator stated that it produces credit. The fifth component here is an electromagnetic wave transmitted by the air to a number to four. As you can see, we have said that it would work at a certain frequency so that it would transfer waves instead of electromagnetic waves to the recipient of a boring number for this fourth component, there is a second magnetic coil or An antenna installed on a laptop or TV. It is when it is the same frequency as the first call here and absorbs the energy sent by the first coil. As you can see, it's number five. It shows that the energy loads the device so we have here a coil. And here we have a coil. This one from the electromagnetic waves. This is this magnetic wave and everything moves into energy or store in the battery or all that is combined here. Now let's see the bases. We have inductive couplings. We have a magnetic field and we have the resonance. These three components of the month must be used to be applied to any system. Now let's see our description of the system. Now, as you can see here on the left side, we have an AC DC CC converter. Now we convert DC AC

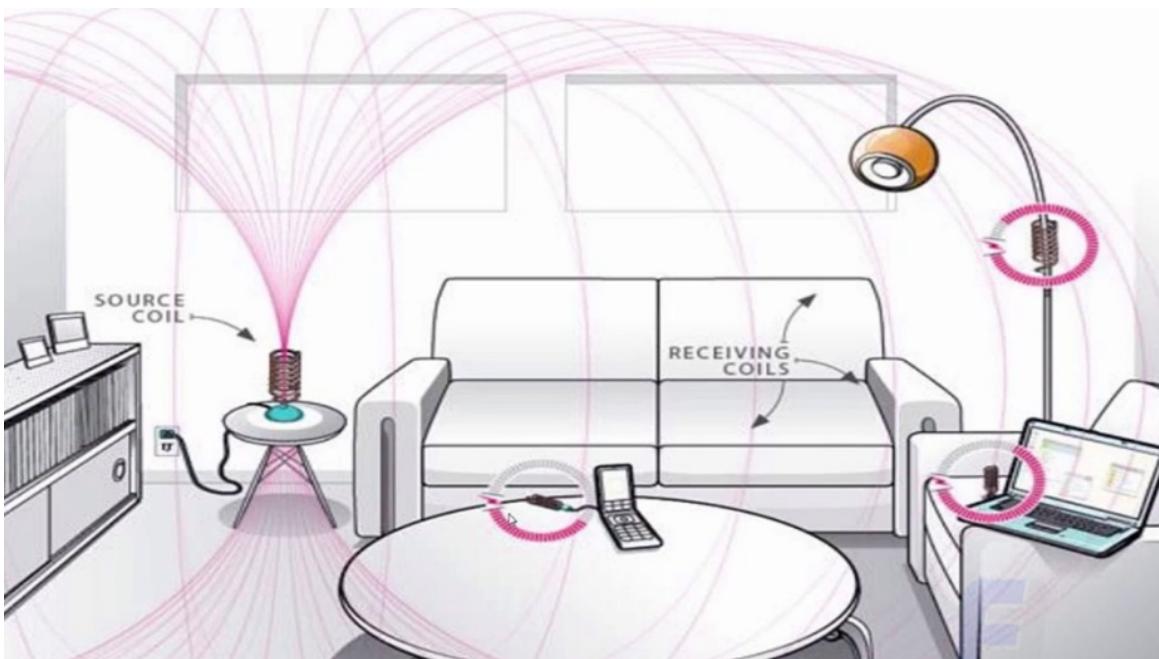
power consumption to DC or direct current. After that, we convert amplifier frequency amplifiers or resonance. Then we have a corresponding impedance that did not work. These two elements must be there to ensure that the transmitter and the receiver operate at the same frequency and produce the same quality magnetic field. After that, we have a source with meters as you can see. This will produce a magnetic power and send it to the device that comes from here, it will return to where I will send it to DC or Rafter DC Revitters. It is at this point that the frequency of the DC rectifier will improve the magnetic waves or the resonant frequency waves and transform them into direct or direct current that can be used by God. This is a quick description of the system indicating the internal components you need to use to produce energy. As you can see or break wireless electricity if one of these components is unclear for you, I can explain it more, ask questions up now. And quickly look at this operating principle. As you can see, it is a physical symbol or a wall of physics at the source. Electricity is Transfield as you can see here is a magnetic wave on the receiving device. The magnetic waves of Detresporti become electricity. So, we have a thread. We transferred this power over the electric waves Tom McClintock and sent it through the air. Here we have a coil that receives this magnetic wave and transforms it into a barrel crossing all the devices and the rotates. That it is. I hope you now have a clear idea of wireless operation.

## **ADVANTAGES AND DISADVANTAGES**

These companies are safe enough efficient and have a good range. This will help us get rid of many cables so that we will consume less

Kabar, it's safer. No risk of shooting because it is wireless, you can not touch a wire that is poorly connected, it can go to places where you can not have a physical cable. There is no one else who has the advantage of the man we have to withstand the need for the battery is eliminated. Electronic waste is eliminated. Maintenance costs will be recorded because there is no cable. It's just an underwater transmitter over the range of what can be increased. So these are the main advantages of what is thirsty. Now let's see that its advantages, why bar transmission can only be used in a few minutes. For the moment, this efficiency is only about 40%, we can achieve more than 40% of energy-efficient or electricity transmission. There is a great need for standardization and adaptation. So, a small overheating occurs because of different voltages of the transmitter and the receiver. Reprise the old equipment or purchase new equipment could become very expensive because you could have a device that you want to activate at a wireless power, but it does not support that you will need to buy actions in your equipment to transfer it to the wireless. -The ring road. And it will be very expensive or you will have to buy a new device with a wireless power option. It will also be expensive because you will have to buy a new device and there is a very good possibility of flight. While sport is just like any other wireless network you have about it can come to steal your power. If you transfer it over a wide range, there are ways to protect it, but it's expensive for now.

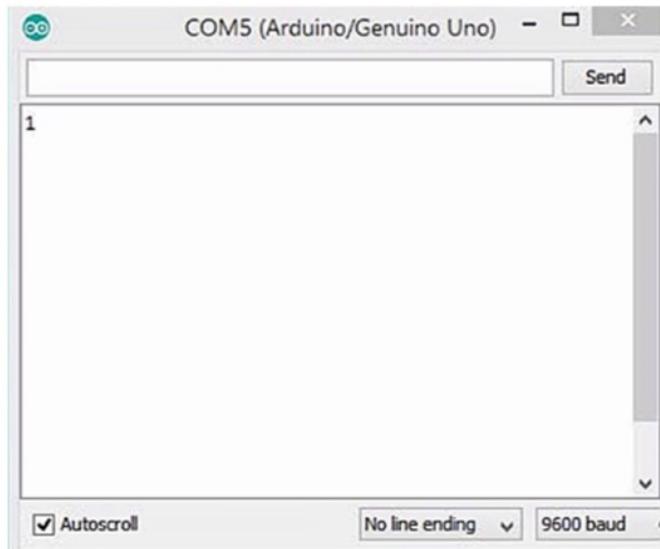
# SCOPE OF WITRICITY AND WITRICITY APPLICATIONS



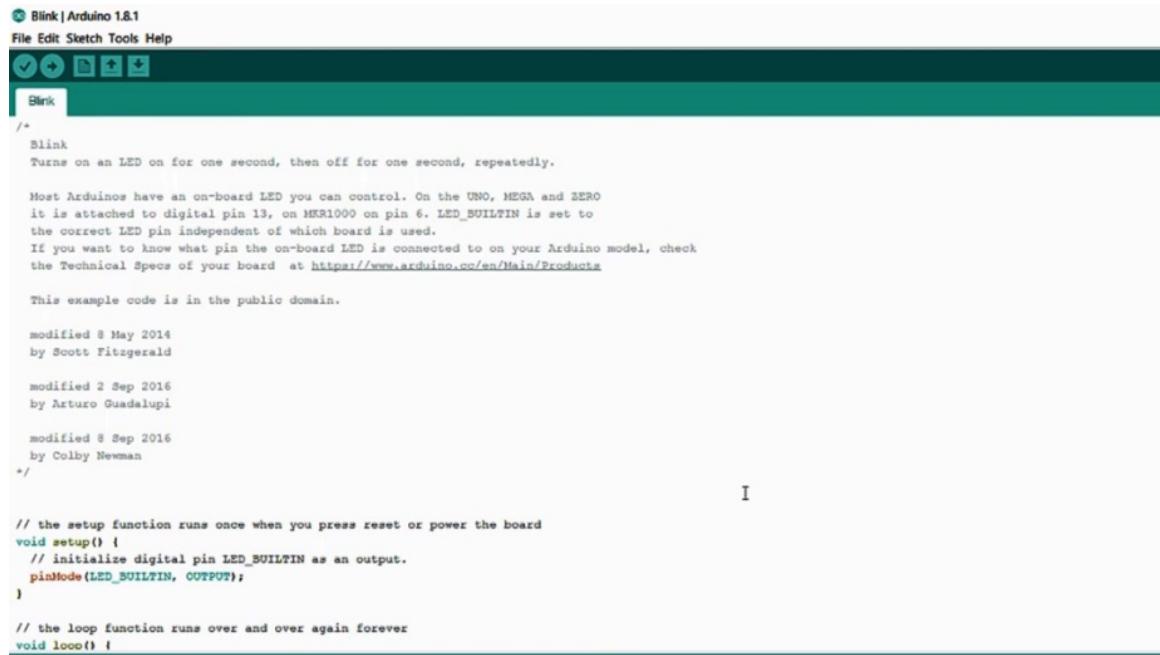
Let's start by talking about the futuristic wireless electricity range for two centimetres of electric transmission to three to four meters depending on the equipment you have. Scientists work on the extension of this range and should become 10 meters in no time. As you can see that a source code is connected to the mains power can transfer the energy of the electronic devices of this part. It's all the barn with a silver reel. This is the laptop with Steve Alcorn. This is aligned with Steve Alcorn because you can see all these well-received receiver coils from this transmission piece and will work wirelessly. It is therefore an example of using wireless electricity transmission within ours, as you can see. It's about four meters wide. Now, as you can see, these are three but batteries can be loaded. Simply place this unit to tell one of the surfaces. This is a surface used when it's a lot of surfaces. You can simply place your device and will be charged wireless without having to connect. Let's say a cable. You must be able to load it. But it's a very small distance that you can see. You can load it by turning your hand. It is therefore a bit

of our booth to sit down, but they work to solve it so that this system can be achieved with high efficiency. More than 40% This is another example of electric car loading using writing, as you can see, an electric car has a battery and this battery is loaded using as you can see here. The coils are connected to the AC boiler. They are transfinite with magnetic waves. Current wireless technology is implemented in the Bluetooth Wi-Fi connection and APS-C communication or near the satellite communication of the microphones mouse wireless mouse and headphones. These are examples that use wireless technology in general. And here we are talking about wireless power transmission. Now, what do the products of wireless power transmission look like? This makes them more practical, with no manual or charging or changing batteries. It eliminates bar codes unlikely and expensive. This makes the devices more reliable. It is never missing from the stack to use a product failure. It is by repairing the lowest link to the environment. Of course, because it is reduced to use lots and a lot of wires, it has reduced electrical waste. So you do not have to throw cables and do not have to keep them, let's just add copper wires on your network or more and more devices that you can boil wirelessly to reduce the quantity of waste and help you reduce help the environment.

# HOW DOES IT WORK



In reality, 0 5 or 0 6. Depends on what you will have for the Bluetooth model runs on serial communication. Here is the under the Android application. It has designed serial data on the Bluetooth module when you press some buttons. The Bluetooth will do all the other in the data and sends the Arduino jet that X has been completed with what you are going fundamentally is Arduino's ex-Penn the code adapted to Arduino chickens who have received data from This Bluetooth all and compares the received data is the one that turns on or if it is 0, it will disable the link on the received data to test it.



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.8.1". Below the title bar are standard menu options: File, Edit, Sketch, Tools, Help. A toolbar with icons for file operations like Open, Save, and Print is visible. The main window displays the "Blink" sketch. The code is as follows:

```
/* Blink | Arduino 1.8.1
File Edit Sketch Tools Help
Blink
/*
Blink
Turns on 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

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald

modified 2 Sep 2016
by Arturo Guadalupi

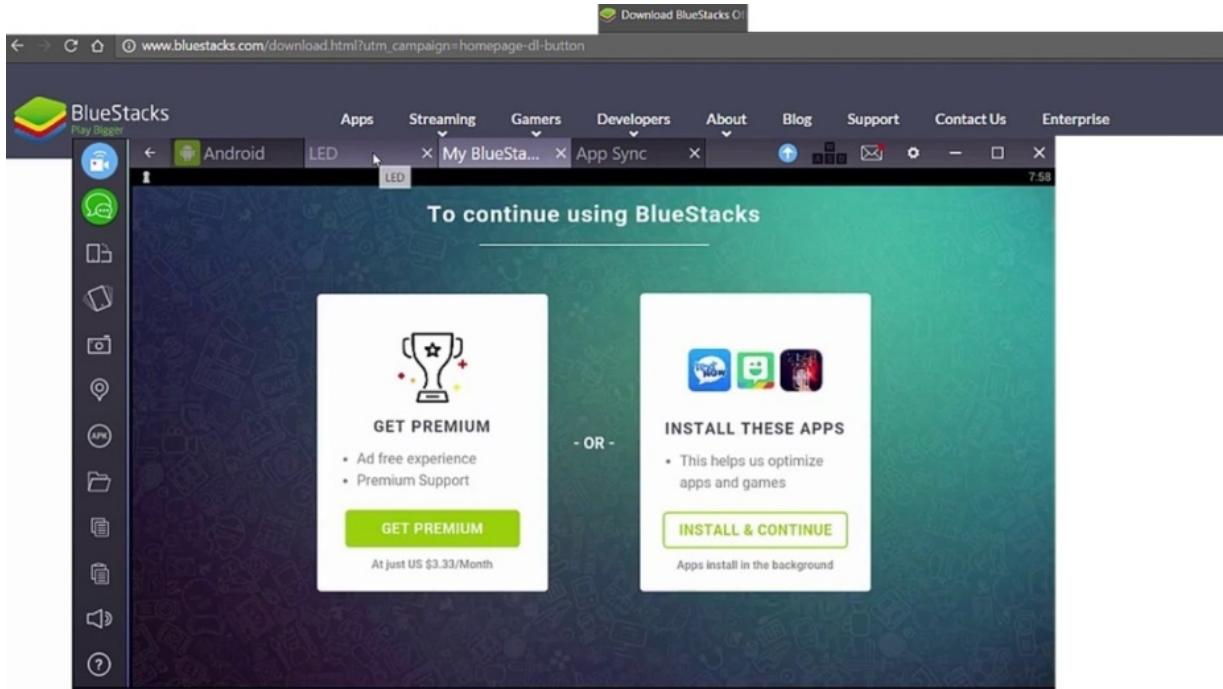
modified 8 Sep 2016
by Colby Newman
*/
I

// 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() {
```

You will need to open the serial monitor and watch the received data. Allow me to show you the monitor series before starting to experience it. Go to the Start menu and look for an ID. Now, as you can see here, it's our idea. This is an invoice from society. You have to go tools here and go to the serial monitor here. As you can see, that's it. This is the city of Toronto that will act as our mobile phone and the simulation. You can also go by clicking Ctrl-Shift. Also, I'm on the keyboard and it will be Bob. It's a person once we connected the green all we need to go here and choose the ball we connect the green onboard. That's why it works. As you can see, it's really easy.

# ANDROID APPLICATION



You can download the Android application from here and the source code of the entire project so you can change if you have an earlier Java experience. Now, how to use them, do not know the application by here from this link that you and WD Tooth Comber Slash and Slash Arduino Bluetooth which is our official website or the other link. If this thing has not worked now after being on this website, you will find the download link in this article that will be as you can see here. This is the download link you will get from the article and you will be directed here to this very folder. Now, as you can see, we had an ad megabyte file that is and this zip format you must unlock it and start downloading this farm. Now, once the download is complete, close it and go to where you downloaded this right click then extract. So, Arduino Bluetooth as you can see. This is our file. Once you have entered this folder, you can see that we have your files. This is the Arduino code. This is the beginning and it is the zip file for the source code of this large file. This file is basically what you will download and install on your phone. Now, for simulation purposes or will use an Android simulator for the Windows environment. You can get it by going here bright blue stocks, as you can see that Blue Stacks is an American technology company that uses the Blue Stuck drive used to simulate the Android system on your Android Windows operating system for PC and Mac. If you went there, you can simply click here download bluster and just wait. So you can go there. As you can see here, it downloads. But since I downloaded this and install it, I'll find it here. Blue batteries. That's all. Now, as you can see, it will take

time to start that. That's it. This is our simulation. OB Stocks It will take time to download everything. I did not think about the specifications of your computer as the graphics card and the Sibiu. Now once you have had once your office, you can simply check these options as you can see. This one is the one we need. We must install it. OKAY. This requires the location. We must therefore copy this location on the best to tear this. This is our biggest finding simply double-click on this case and now, it installs an OK key, you can see that it has been installed. Just click this to start our application that is covered. He asks us to do something if we want to continue using Bluestar. OKAY. And it's thought to continue. Now let's go back to the main demand. OKAY. It looks like we need to get these applications installed before you can use this software. OKAY. Now, it asks us and I wanted to activate Bluetooth. Just click Allow Bluetooth shoot on. This is our first plan, but as it is a simulation environment, we could deal with problems because there is no Bluetooth. What you want for this environment. I just wanted to show you the use of the interface. So, let's go back after running the software and once turning and go with the truth, you have to build your device with a C. 0 5 0 0 5 0 2 0 6 2 What you will turn on I see 0 5 0 0 0 6 Luthern then scan for available devices using a mobile phone invoice to see 0 5 0 6 by entering a different password or says one two three four or four zeros. It's the first step for running the software and build your Android device with the Bluetooth that you have. As you can see, this is the user interface you need to click on your devices, then bless their devices, a screen of your party to lift your Bluetooth what you go from the list that is seasonal 5 or 6 0 6 After the connection successfully entering the PIN code 4 0 or 1 2 3 4 Press the key, then on this cover that is connected to UP, we know on this basis to turn off this light that is connected to one or the other you know when you're done. Just click on the disconnect button and you must go then turn off your router what you are. There is an interface for the application. After downloading the file and install it on your phone. It's for this lesson on the Android application. If we returned here as you can see, it is faced with a problem telling the Bluetooth truth your sins. There is not here. In simulation mode, but you simply transfer a file to your mobile phone and install it on your phone. Since it's Bluetooth, you can simply discover that it is 051 you all and everything will be able to go there. That's it for this lesson. Thank you for watching. If you have any questions, please ask and you are on the IBAR. That's all for the moment. It is educational engineering.

## SCHEMATIC

This is for us to weigh on everything this is the USP programming cable that you connect to your computer and it is also used to look at what we know in some cases if you do not have a lot of Requirements, this is the decision you connect. BHULLAR 5 VOLT

SMALL Supply for this if you need more requirements from the Board of Directors. And as you can see, these are the sockets that interface the Arduino on our microcontroller with the other for this dx and our zero expense and the one connected to the XX as you can see here connected to the x in the wall DXI take place to start in the window out of the window. And these two sons that the Earth and the VCC are connected as land and VCC for the bolts for OPs on them. Arduino is Arduinino Boring The Bluetooth module using these two red and black wires. This is a quick conversion for a schema. And as you can see that number 13 is connected to this store and the positive terminal thereof, why this negative minute is connected to the bottom on board. Then we will talk about why we connected it has been about 13 years since we would use it in the Code section, but that's it for the schema. This is the final version before going to the code.

## CODE



```
CODE | Arduino 1.8.1
File Edit Sketch Tools Help
CODE

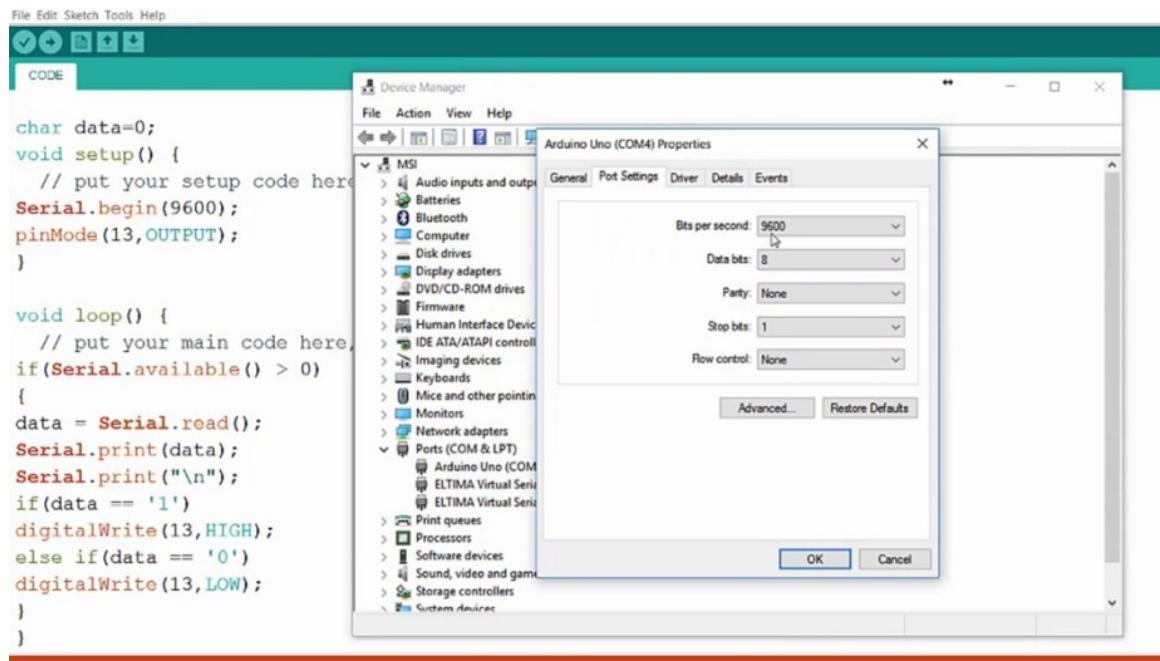
char data=0;
void setup() {
    // put your setup code here, to run once:
Serial.begin(9600);
pinMode(13,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
if(Serial.available() > 0)
{
data = Serial.read();
Serial.print(data);
Serial.print("\n");
if(data == '1')
digitalWrite(13,HIGH);
else if(data == '0')
digitalWrite(13,LOW);
}
}
```

So, try the code we have to know that you know I just know how to go to the man and hit, we agree on any identifier. That's it. And we know. First of all, we must first create a new project as you can see here. This is on your project window that I and CREE are the size of this window. And let's see what we need from this manages ok. He has a little launch. OKAY. We need to increase the size of the police to 20 so you can well it's good. This is the configuration you can put your configuration code here to run once. So, between these two hooks, we are going but our configuration code for turns. This is the newspaper, so it will be it and you have to put your main card here put on the beat command indefinitely. First, if Pfui must identify any variable, we need to define this area here, we need a named variable of data and give it a zero value. This is the initial value. You do not need to put it so that you have control of the first value of this. First of all, if you are literate without equal to zero and that you will have to get the available value, then more. So, no, we have to take control of this variable, we give it a value that we know is zero. So this change will happen from this base that is zero. Not in this area, you must click with your mouse button here and start the senior communication module. You only have to write a cereal point begins. Then, as you can see here, the baud rate that is measured in bit per second how many bits per second will be transmitted or received. It must be identical for both modules and we use 9600 baud flows for the serial data regime mission here. And the configuration, then we must assign the number of tray 13 to be out. But since we communicate it, it will take an output signal to activate or disable. Was right that men on board, which is 30 years old, then produce that. We initialize the flow rate of the 9600 baud communication, then we attribute the number 13 tray to be as an output. Now, in the empty loop, we must write a statement if it checks if there is a real communication occurs or not. And so that we send only data when we receive

data. Thus, if okay, it is the main form where we need to condition more zero and inside, inside these two buckets, we would like our code. OKAY. Ok, now I would of course write that he is in the middle between these two taxis. First, we need to read the incoming data and store the incoming value and data on your banking data are equal to cereals that read. Now we need to bring the incoming data and the value of the data into the Syria data on top so that we can make sure we will see the cereal of light. Dr Brant by the variable data he is after you need to move a line below. So, if we thought when I will not come from a line, as will see again. Brant OK semi-colon inside that we must flaw on an abbreviation for the new line and means in the new line. OK after thinking of the data we want to verify that these data are equal to an ok. That's what it's equal to one. If this is equal to the one we need to write as we mentioned in the application where we need to activate the head, which means that I think that the value of one of the high logic of about 15 is now so The value received is zero. Then we need to write a weak or zero logic to report about 15 to 10 of this advance. Ok, that's it. This is our code. Now we have to back up the game 7 slides or bend or bend you with your chords. OK called. Just click this button to verify that our code is not at all of the syntaxes. As you can see, combine sketch ok everything is fine from. As you can see, this is not all it is done here, the combination of our code contains only 19 lines. OKAY. And now I thought if you want to do that, we just have to click here. After assured that we choose the right to come from here. After that, we do not know we need to go to this device manager. And check the clipboard you can see here, which are on I, I'm connected, we have to move the fight here. After that, you can simply be connected by clicking the Block button here. But since he does not have an Arduino, he will not make war. And the next lesson, I will connect to an Arduino on the money and we will test this code in the city on the monitoring card. But before doing this examine what we did here. At first, we have set a volleyball to store received data. Then we defined the table for transmission. And this line. Then we said digital flap 13 as an exit. After that, we send data only when we receive data. You must check if there is one received person. If there is then the incoming data and store it in this variable mix the value in the Value data and see the monitor on your line to reflect on any new data verification if a data value is equal to 1. Now, if the value is now. This is then less than if the value is zero and then layoff. This is our code. It's really simple. You can change that with what you want, but all this followed the same principle.

## TESTING CODE



Arduino Tip for you or you will be bored and your computer or your laptop OK to connect it. Let's go to the Device Manager. To see the combo that will be assigned to us, we know. So, as you can see here, there is no affected. Now, once we can Nick We Board as you can see Arduino on the coming. If you double-click here as you can see. We see these other bullets to things. 9600 Baud rate. Now we have to go here and click on Chrome for all we know, then simply continue to Click Check the file called again, and then click the download button to download it. OK downloading download. Now the code is downloaded from our Arduino on a ball to test it. Come on Siri tools on the monitor.

The screenshot shows the Arduino IDE interface. The top menu bar includes 'CODE | Arduino 1.8.1', 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main area is titled 'CODE' and contains the following Arduino sketch:

```
char data=0;
void setup() {
    // put your setup code here, to run once:
Serial.begin(9600);
pinMode(13,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
if(Serial.available() > 0)
{
data = Serial.read();
Serial.print("The Number you entered is : ");
Serial.print(data);
Serial.print("\n");
if(data == '1')
digitalWrite(13,HIGH);
else if(data == '0')
digitalWrite(13,LOW);
}
```

To the right of the code editor is the 'Serial Monitor' window, titled 'COM4 (Arduino/Genuino Uno)'. It displays the following text, indicating the program is receiving input from the serial port:

The Number you entered is : 1  
The Number you entered is : 2  
The Number you entered is : 3  
The Number you entered is : 4  
The Number you entered is : 5  
The Number you entered is : 6

At the bottom of the serial monitor window, there are two checkboxes: 'Autoscroll' (unchecked) and 'No line ending'.

Now, if we sent one as you can see or two or three or four. And the thing we send here if we use our Android app, let's say and I think we are here must come back. Now, to see the changes here and when we enter, but it must show us this number. You talked about asking us and a tool with that number we liked. So, if you have written it, he will start with one. Now to see the changes. Let's add a sentence before this one says that I am bobbing that your impatience is ok. Now check our code and download it. Now, if you have written a sentence answer with a sentence, the number you find yourself like the same as any other number. As you can see is that I think this version of this sentence, is somehow thinning the variable it has done in this line. So, he has been integrated into us for us. We showed the T. As you can see, it is essentially the way cynical drugs work and this is tested for our code.

# EARN WITH ARDUINO

How to make money with Arduino. Robotics is a daydreaming field that creates armour in the future of the iron or even makes a more reasonable project that you can use in everyday life through the spirit of an inventor of robotics is a world full of promises and gives us the power to remove problems with solutions This new might sound complicated and we think that you have to learn for a long time because we need significant finance to make it happen if you want to realize the robot project to commercialize you need electronic tools for hardware production equipment and therefore investors That only a company that is capable of this is why we have the right to wonder whether this dream can be accessed or if it's too big but when we want to succeed we can't give up. I just want to make my robot project, I will do the project every week. I have a butterfly in my stomach today with any computer just on the Arduino board on components such as sensors on your actuator can realize a promising robot project and your internet blessings can make your discovery known and start selling it on the crowdfunding platform among the best Kickstarter and Indiegogo Known Kickstarter allows you to make your project look in exchange for 5% some managed to increase hundreds of thousands of dollars and other millions of dollars, it's not so simple. The success rate is 36% and three-quarters of those who fail to get less than 20%. At least we can say is whether it's like it or not at all. After your Kickstarter campaign, you will have funds to start production and this will allow you to make yourself known because you are a small team and maybe still alone, you don't need to get a lot of customers to make your business worthy of your business you can see the campaign Successful and simple concepts. The key is that he still responds to real needs. Their testimony that has launched many major projects that fail which why you need to know the main points that will make you project success. This success is not for mysterious reasons. They are just logical. (With content requesting your emotional visualization and senses). Highlighting the benefits of your product, this is a very important point. Present on social networks a few months if possible before your launch. This is social proof that prepares the launch of

your Kickstarter campaign by taking emails from people who are interested in which you get through social networks and/or your website. And this is to warn them that your campaign will be launched. Indeed Kickstarter highlights projects that have funds that come in the early days. Contact Blogger, Influencer and Media in your industry to help you market your product. And of course, have the motivation to patience. To select the project you want to live in course by knowing that you will not be appreciated until you finish your campaign, maybe you don't know in this case there is a method to find ideas or at least to identify whether a good idea.

**THE END**