

**WELCOME TO ROBOTICS & ELECTRONICS
CLUB**



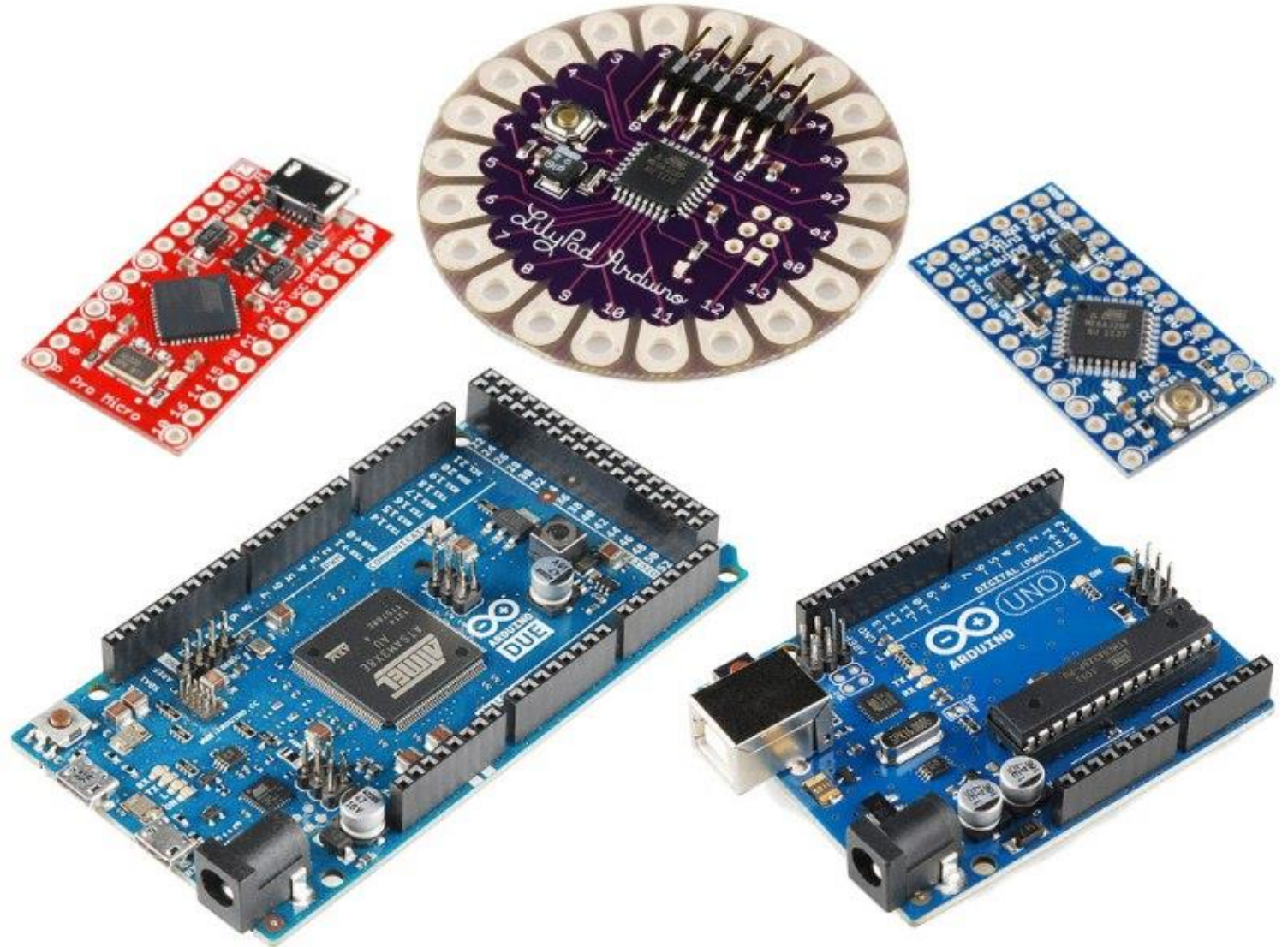
INTRODUCTION TO ARDUINO AND ITS PROGRAMMING

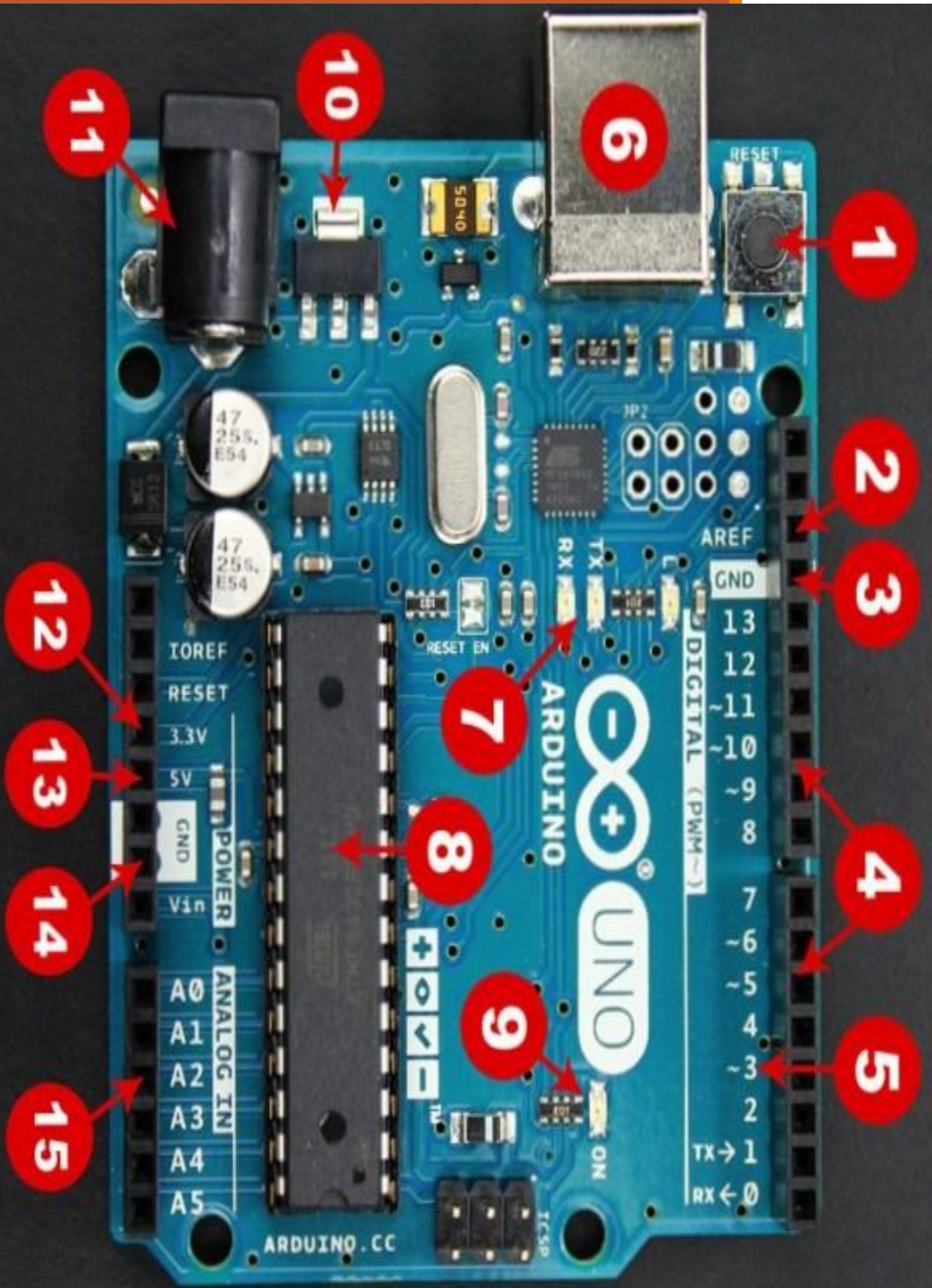
-First Session

Why Arduino?

- **Open-source project**
 - both software and hardware is easily accessible
- **Flexible**
 - Works with both digital and analog signals
- **Easy to use**
 - Connects to computer via USB
- **Inexpensive**
 - Comes with lot of variety and specifications
- **Lot of source code is available within software and online**

Check: “ <http://www.arduino.cc/> “





ARDUINO PARTS

- 1) RESET BUTTON.
- 2) AREF.
- 3) Ground Pin.
- 4) Digital Input Output.
- 5) PWM.
- 6) USB Connection.
- 7) TX / RX.
- 8) Atmega Microcontroller.
- 9) Power LED Indicator.
- 10) Voltage Regulator.
- 11) DC ower Barrel Jack.
- 12) 3.3 V Pin.
- 13) 5V Pin.
- 14) Ground Pins.
- 15) Analog Pins.

novation

RULE BOOK

1. Listen to your Mentors
2. Nothing is Big, unless you try to make it!
3. Follow Rule 1





- Collect Arduino Boards
- Connect Arduino UNO to the Computers
- Open Arduino IDE

LETS BEGIN!!!

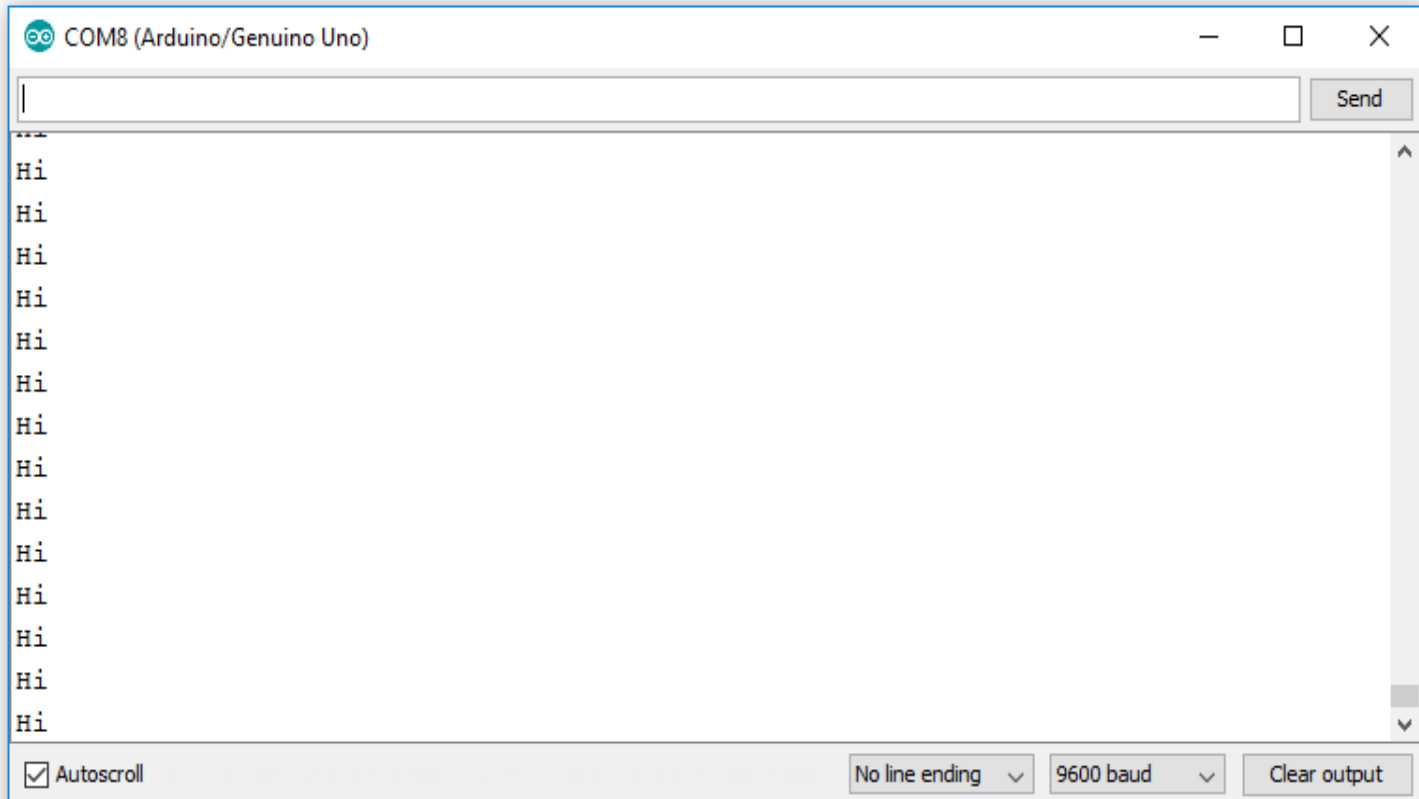
```
void setup()
{
    Serial.begin(9600);
}
void loop()
{
    Serial.print("Welcome to IIT PKD");
}
```





print_hi

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Hi");  
}
```



Done uploading.

Sketch uses 1468 bytes (4%) of program storage space. Maximum is 32256 bytes.

Global variables use 190 bytes (9%) of dynamic memory, leaving 1858 bytes for local variables. Maximum is 2048 bytes.

Questions?

1. What is Serial Communication?

- process of sending data one bit at a time, sequentially, over a **communication** channel or computer bus



2. What is the use of Serial Monitor?

- The **serial monitor** is the 'tether' between the computer and your **Arduino** - it lets you send and receive text messages, handy for debugging and also controlling the **Arduino** from a keyboard



3. Why 9600 is written there?

- 9600 is the baudrate used to communicate with Serial Monitor
- Sets the data rate in bits per second (baud) for **serial** data transmission. For communicating with the computer, use one of these rates: 300, 600, 1200, 2400, 4800, **9600**, 14400, 19200, 28800, 38400, 57600, or 115200



Data Types

- In programming languages, data type is an attribute of data which tells the computer (and the programmer) something **about the kind of data it is**.
- These has some constraints on the datum, like:
 - what values it can take
 - what operations can be performed.
- Examples : Integer, Character, Floating number





ARITHMETIC OPERATIONS



Arithmetic_operations

```
/*
 * Performing ARITHMETIC OPERATIONS
 */

void setup()           // put your setup code here, to run once:
{

  Serial.begin(9600);   //initializing communication with Serial Monitor
  int a,b,result;       //stating variables as integers
  a=5;
  b=4;
  Serial.print("Sum: ");
  result=a+b;
  Serial.println(result);
  result=result+1;
  //result++;
  Serial.print("Increament: ");
  Serial.println(result);

}

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

}
```

Done uploading.

Sketch uses 1848 bytes (5%) of program storage space. Maximum is 32256 bytes.

Global variables use 206 bytes (10%) of dynamic memory, leaving 1842 bytes for local variables. Maximum is 2048 bytes.



Arithmetic_operations

```
/*
 * Performing ARITHMETIC OPERATIONS
 */

void setup()           // put your setup code here, to run once:
{

  Serial.begin(9600);   //initializing communication with Serial Monitor
  int a,b,result;       //stating variables as integers
  a=5;
  b=4;
  Serial.print("Sum: ");
  result=a+b;
  Serial.println(result);
  result=result+1;
  //result++;
  Serial.print("Increament: ");
  Serial.println(result);

}

void loop() {          // put your main code here, to run repeatedly:
  1
```

Done uploading.

Sketch uses 1848 bytes (5%) of program storage space. Maximum is 32256 bytes.

Global variables use 206 bytes (10%) of dynamic memory, leaving 1842 bytes for local variables. Maximum is 2048 bytes.

COM8 (Arduino/Genuino Uno)

Sum: 9
Increament: 10

☒ Autoscroll

No line ending ▾

9600 baud ▾

Clear output



Arithmetic_operations01

```
/*  
 * Performing ARITHMETIC OPERATIONS  
 */  
  
void setup()                // put your setup code here, to run once:  
{  
  
  Serial.begin(9600);       //initializing communication with Serial Monitor  
  int a,b,result;          //stating variables as integers  
  a = 2;  
  b = 7;  
  
  float result_fl;  
  
  Serial.print("Addition (a + b): ");  
  result = a + b;  
  Serial.println(result);  
  
  Serial.print("Subtraction (10 - 2): ");  
  result = 10 - 2;  
  Serial.println(result);  
}
```

Done uploading.

Sketch uses 2082 bytes (6%) of program storage space. Maximum is 32256 bytes.

Global variables use 328 bytes (16%) of dynamic memory, leaving 1720 bytes for local variables. Maximum is 2048 bytes.



Arithmetic_operations01

```
Serial.print("Multiplication (4 * 3): ");  
result = 4 * 3;  
Serial.println(result);
```

```
Serial.print("Int Division (5 / 4): ");  
result = 5 / 4;  
Serial.println(result);
```

```
Serial.print("Float Division (5.0 / 4.0): ");  
result_fl = 5.0 / 4.0;  
Serial.println(result_fl);
```

```
Serial.print("Remainder (11 % 4): ");  
result = 11 % 4;  
Serial.println(result);
```

```
}
```

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

```
}
```

Done uploading.

Sketch uses 2082 bytes (6%) of program storage space. Maximum is 32256 bytes.

Global variables use 328 bytes (16%) of dynamic memory, leaving 1720 bytes for local variables. Maximum is 2048 bytes.



Arithmetic_operations01

```
Serial.print("Multiplication (4 * 3): ");
result = 4 * 3;
Serial.println(result);

Serial.print("Int Division (5 / 4): ");
result = 5 / 4;
Serial.println(result);

Serial.print("Float Division (5.0 / 4.0): ");
result_fl = 5.0 / 4.0;
Serial.println(result_fl);

Serial.print("Remainder (11 % 4): ");
result = 11 % 4;
Serial.println(result);
```

}

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

}

COM8 (Arduino/Genuino Uno)

Send

```
Addition (a + b): 9
Subtraction (10 - 2): 8
Multiplication (4 * 3): 12
Int Division (5 / 4): 1
Float Division (5.0 / 4.0): 1.25
Remainder (11 % 4): 3
```

☒ Autoscroll

No line ending ▾

9600 baud ▾

Clear output

Done uploading.

Sketch uses 2082 bytes (6%) of program storage space. Maximum is 32256 bytes.

Global variables use 328 bytes (16%) of dynamic memory, leaving 1720 bytes for local variables. Maximum is 2048 bytes.



CONDITIONAL STATEMENTS

If conditional statement

```
if (condition)
{
    statement1;
    statement2;
}
```

Condition can be a

- **Boolean Expression i.e. True or False**
- **Comparison Operation**



Comparison operations:

- $x == y$ (x is equal to y)
- $x != y$ (x is not equal to y)
- $x < y$ (x is less than y)
- $x > y$ (x is greater than y)
- $x <= y$ (x is less than or equal to y)
- $x >= y$ (x is greater than or equal to y)



If ..else conditional statement

```
if (condition)
{
    statement1;
}
else
{
    statement2;
}
```





If_else

```
void setup() { // put your setup code here, to run once:
```

```
  Serial.begin(9600);
```

```
  int a = 15;
```

```
  if(a<20)
```

```
  {
```

```
    a = a +10;
```

```
    Serial.print(a);
```

```
  }
```

```
  else
```

```
  {
```

```
    Serial.print(a);
```

```
  }
```

```
}
```

```
void loop() {
```

```
  // put your main code here, to run repeatedly:
```

```
}
```

Uploading...



Sketch uses 1674 bytes (5%) of program storage space. Maximum is 32256 bytes.

Global variables use 184 bytes (8%) of dynamic memory, leaving 1864 bytes for local variables. Maximum is 2048 bytes.



LOOPS

First : FOR LOOP



loop

```
void setup() {  
  int i;  
  
  Serial.begin(9600);  
  
  for (i = 0; i < 10; i++) {  
    Serial.print("i = ");  
    Serial.println(i);  
  }  
}  
  
void loop() {  
}
```

Done uploading.

Sketch uses 1664 bytes (5%) 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 bytes.



loop

```
void setup() {  
  int i;  
  
  Serial.begin(9600);  
  
  for (i = 0; i < 10; i++) {  
    Serial.print("i = ");  
    Serial.println(i);  
  }  
}  
  
void loop() {  
}
```

COM8 (Arduino/Genuino Uno)

```
i = 0  
i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
i = 6  
i = 7  
i = 8  
i = 9
```

☒ Autoscroll

No line ending ▾

9600 baud ▾

Clear output ▾

Done uploading.

Sketch uses 1664 bytes (5%) of program storage space. Maximum is 32256 bytes.

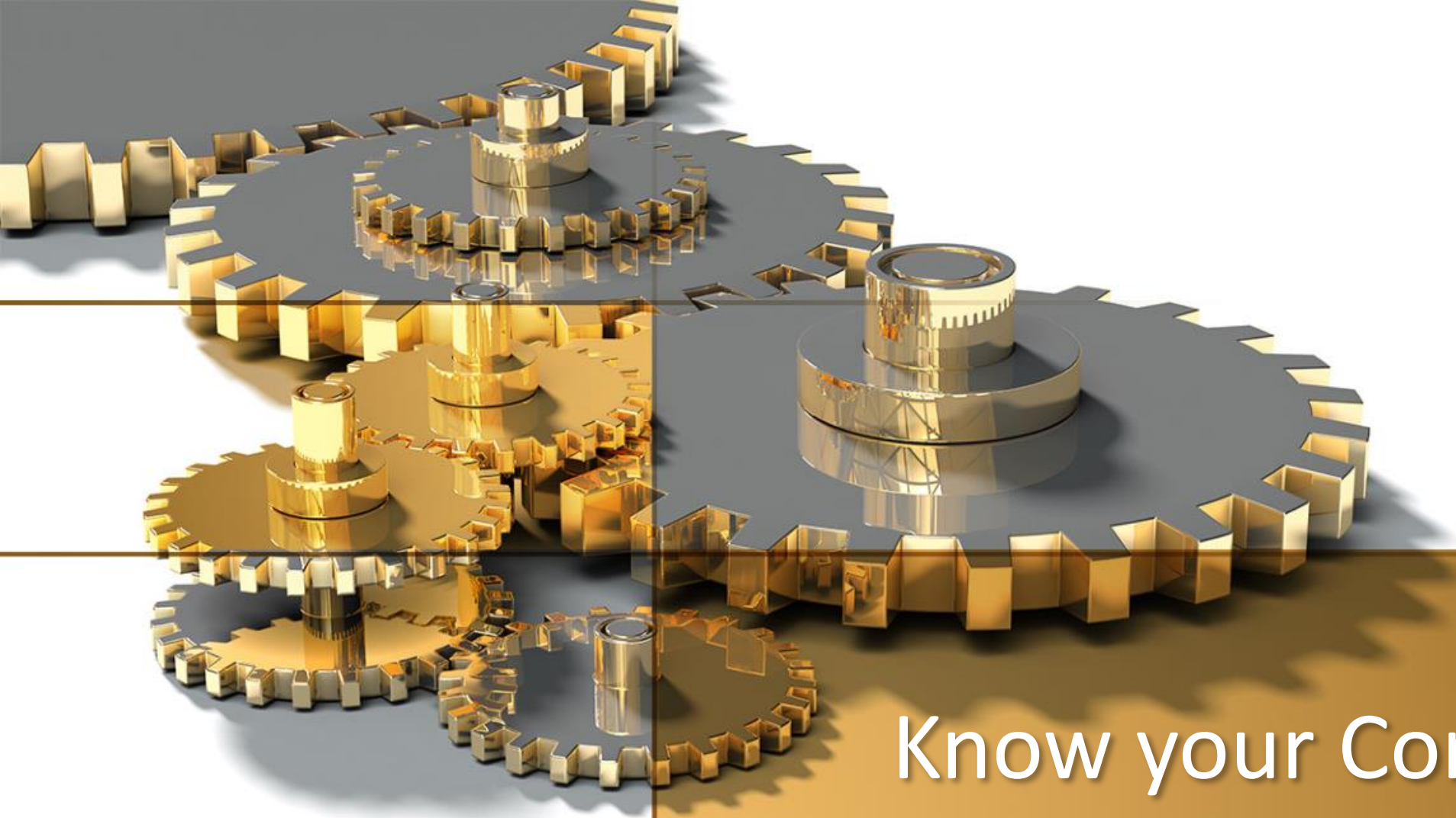
Global variables use 192 bytes (9%) of dynamic memory, leaving 1856 bytes for local variables. Maximum is 2048 bytes.

for loop

```
for(i = 0 , i < 10; i = i+1)
{
    Serial.print(i);
}
```

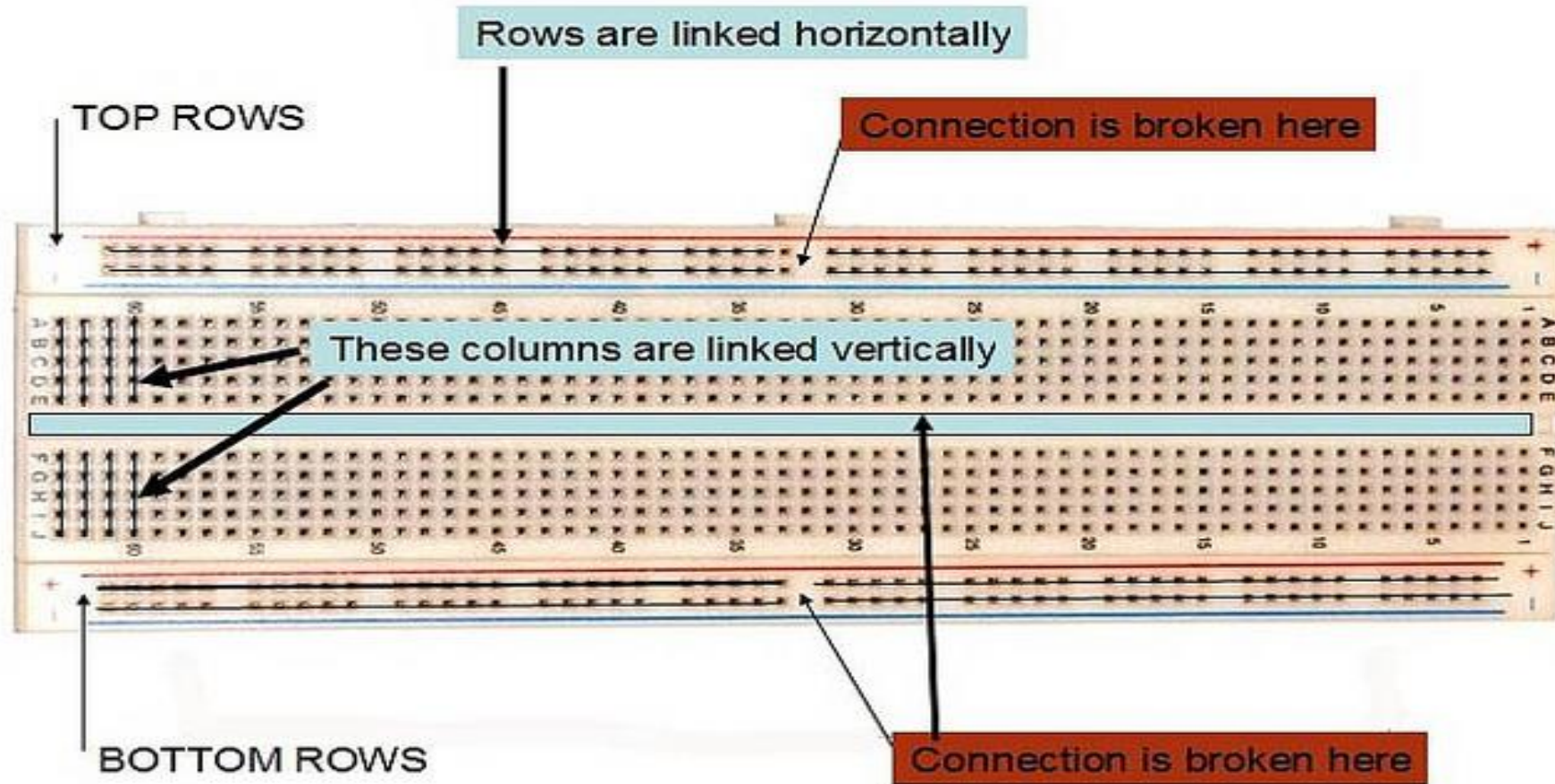
```
for(initializing , condition, change)
{
    statement1;
    statement2;
}
```



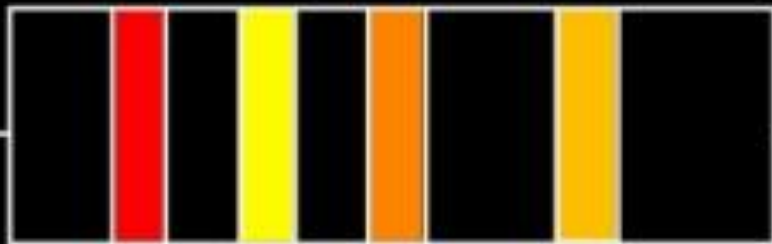


Know your Components

Connections in BREADBOARD



Colour Coding



24 \times 10³ \pm 5%

24 K \pm 1.2 K

Color	Number	Multiplier	Tolerance
Black	0	1	
Brown	1	10 ¹	
Red	2	10 ²	
Orange	3	10 ³	
Yellow	4	10 ⁴	
Green	5	10 ⁵	
Blue	6	10 ⁶	
Violet	7	10 ⁷	
Gray	8	10 ⁸	
White	9	10 ⁹	
Gold		10 ⁻¹	5%
Silver		10 ⁻²	10%
No Color			20%

Find out the value of resistor given to you!



Answer :



- The Arduino board has one built-in LED on digital pin 13.





blink

```
int ledpin = 13,
void setup() {
  pinMode(ledpin, OUTPUT);
}

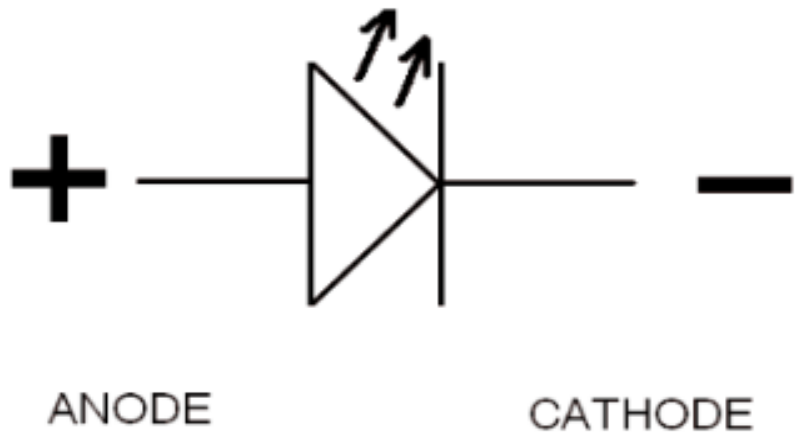
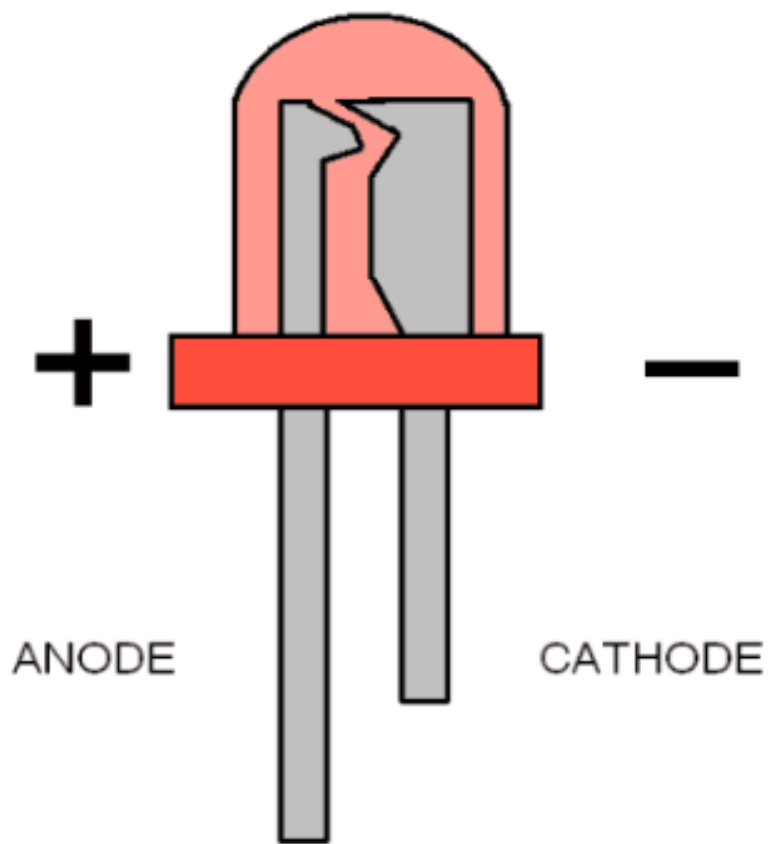
void loop() {
  digitalWrite(ledpin, 1);
  delay(3000);
  digitalWrite(ledpin, 0);
  delay(1000);
}
```

Done Saving.

Sketch uses 948 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.

LED



- Longer Leg == +ve
- Shorter Leg == -ve
- Now Connect an external LED to Pin 13



This is how it looks!



Precaution : Always connect a Resistor in series with an LED



Now it's time to work with

SENSORS



Input or Output to a Sensor can be

- Digital (or PWM)
- Analog

Look at the Arduino pins given for it
We will discuss about PWM soon...



Initializing a PIN

How do we know that a pin is Input or Output pin

It can be Both. HOW?



By initializing a pin in Void setup

Ex:

pinMode (pinName, OUTPUT);

or

pinMode (pinName, INPUT);





blink

```
int ledpin = 13,
void setup() {
  pinMode(ledpin, OUTPUT);
}

void loop() {
  digitalWrite(ledpin, 1);
  delay(3000);
  digitalWrite(ledpin, 0);
  delay(1000);
}
```

Done Saving.

Sketch uses 948 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.

Working with Digital PIN

Giving Input value to a Digital pin which is in terms of voltage

It will be either HIGH or LOW Voltage



Changing input of a pin in Void loop as 5 Voltage

Ex:

```
digitalWrite ( pinName, HIGH);
```

or

```
digitalWrite ( pinName, 1 );
```



Changing input of a pin in Void loop as 0 Voltage

Ex:

digitalWrite (pinName, LOW);

or

digitalWrite (pinName, 0);





blink

```
int ledpin = 13,  
void setup() {  
  pinMode(ledpin, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(ledpin, 1);  
  delay(3000);  
  digitalWrite(ledpin, 0);  
  delay(1000);  
}
```

Done Saving.

Sketch uses 948 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.

**We will discuss about Analog pins in a while
but now you can guess**

How will we give input value to a Analog pin...



i.e.

analogWrite (pinName, LOW);

or

analogWrite (pinName, HIGH);



Potentiometer.....

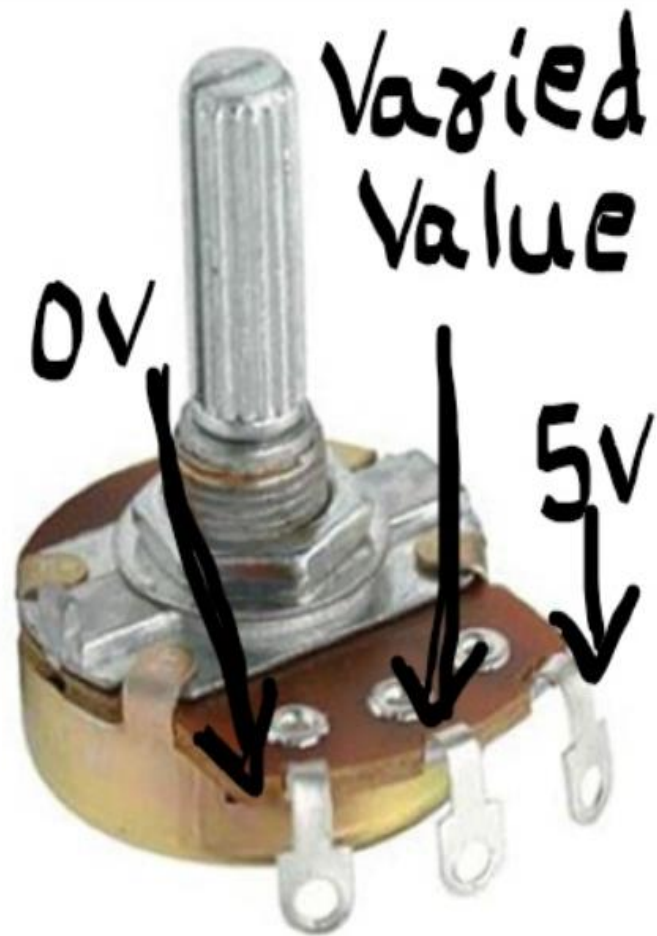
- 3-terminal resistor
- Forms adjustable voltage divider
- Can acts as a variable resistor
- In Arduino, we read it as an **analog value**

Types are:

1. Rotary

2. Slider





Pin Connections

PS: Remember Varied for Middle Terminal

On varying the resistance in potentiometer, it will act as a voltage divider.

The divided values will be in Analog as above stated.

Now, we will try to read those values!!



Analog Values can be read as:

```
a = analogRead(pinName);
```

Similarly, for reading digital values!





pot

```
int a;                      //initializing an integer

void setup() {
  Serial.begin(9600); //Serial communication starts
  pinMode(A0, INPUT); //Analog PIN A0 is giving input from Potentiometer
}

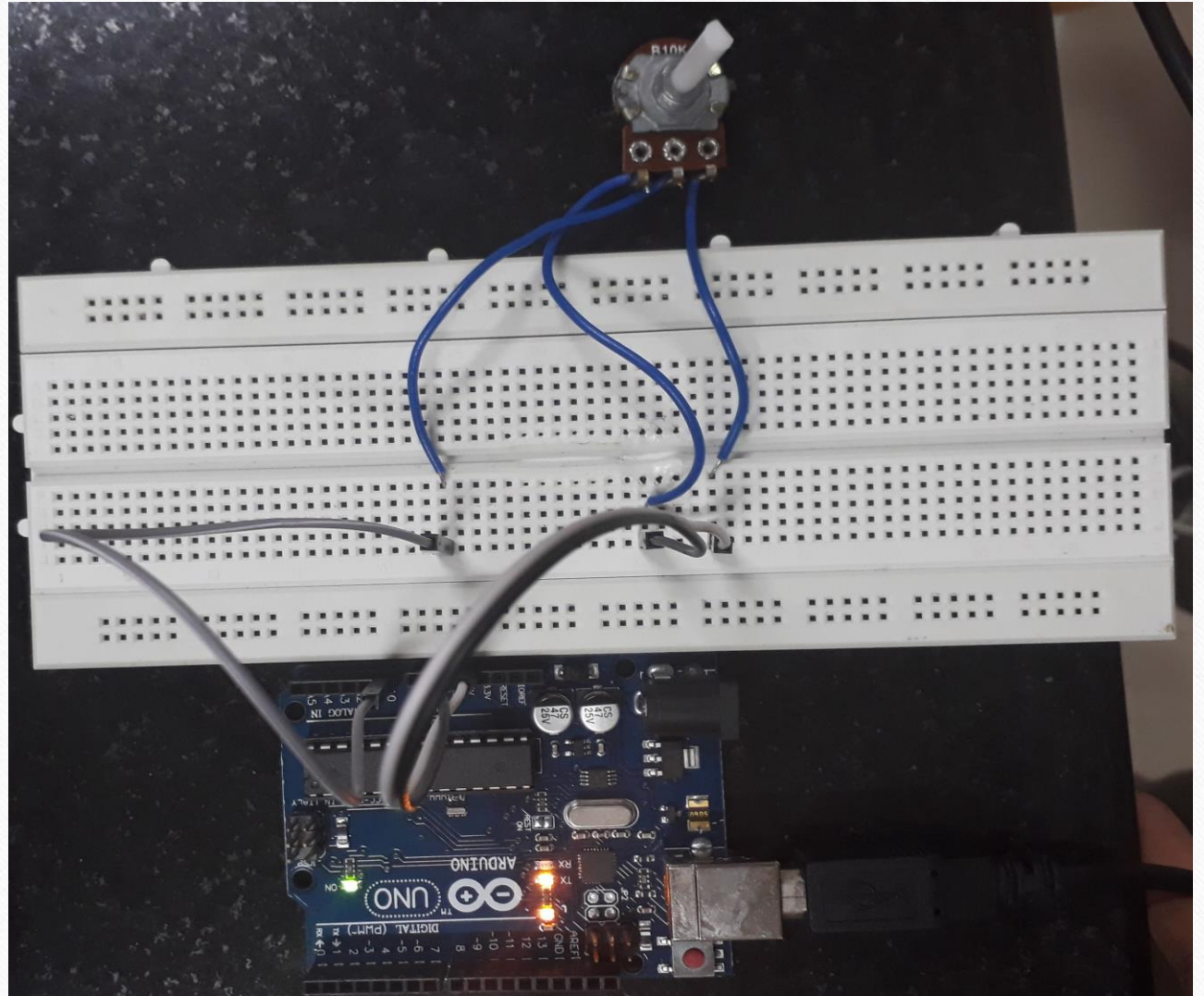
void loop() {
  a = analogRead(A0); //reading values from potentiometer
  Serial.println(a);  // Printing those values
  // delay(100);
}
```


Potentiometer

THINK

What if you connect an LED with
the potentiometer?

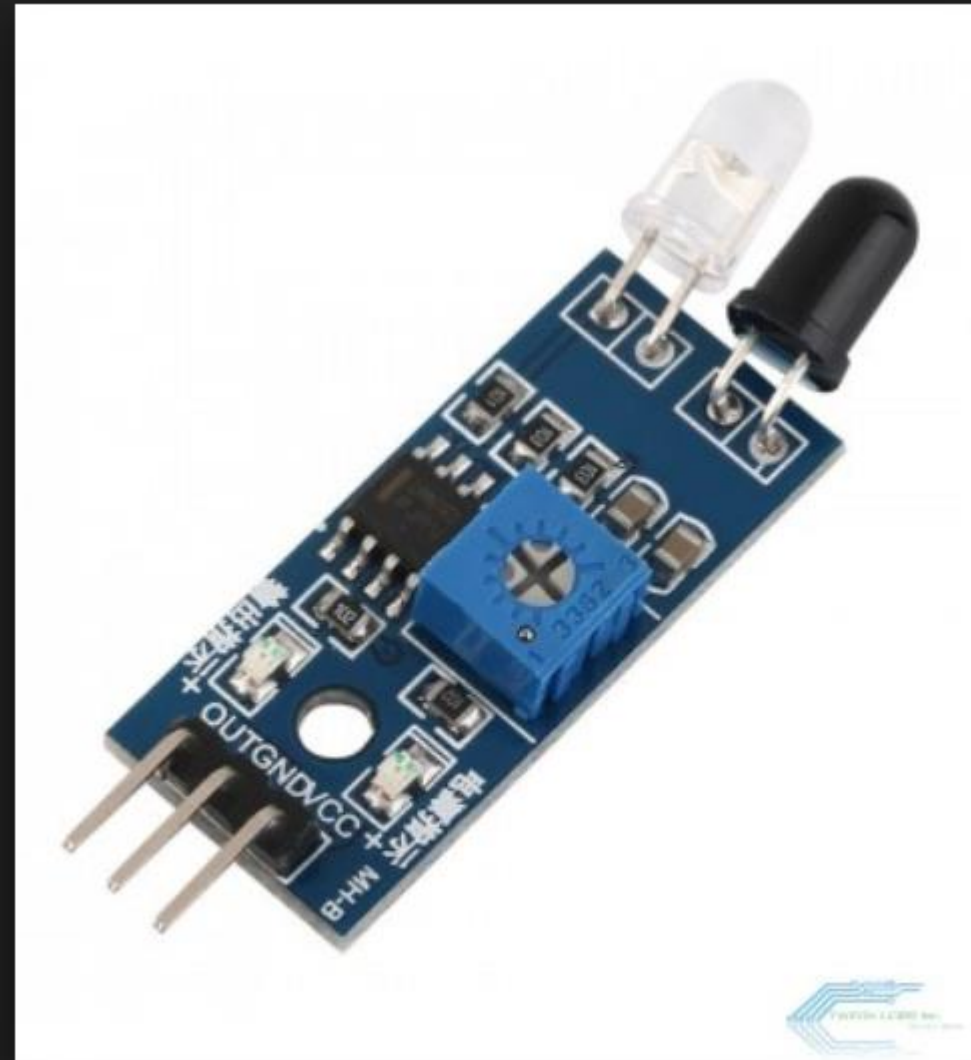
Try to implement it



IR Sensor(Infrared)

IR consists of:

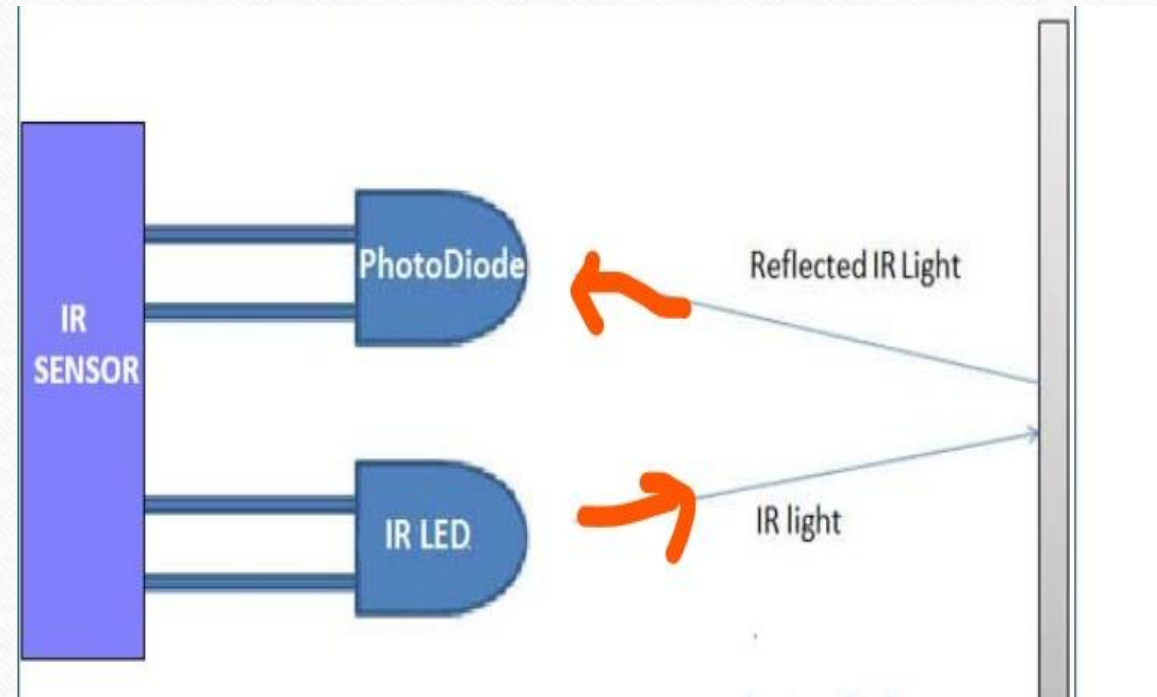
- Emitter- IR LED
(White LED)
- Detector- IR Photodiode
(Black LED)



IR Diagram

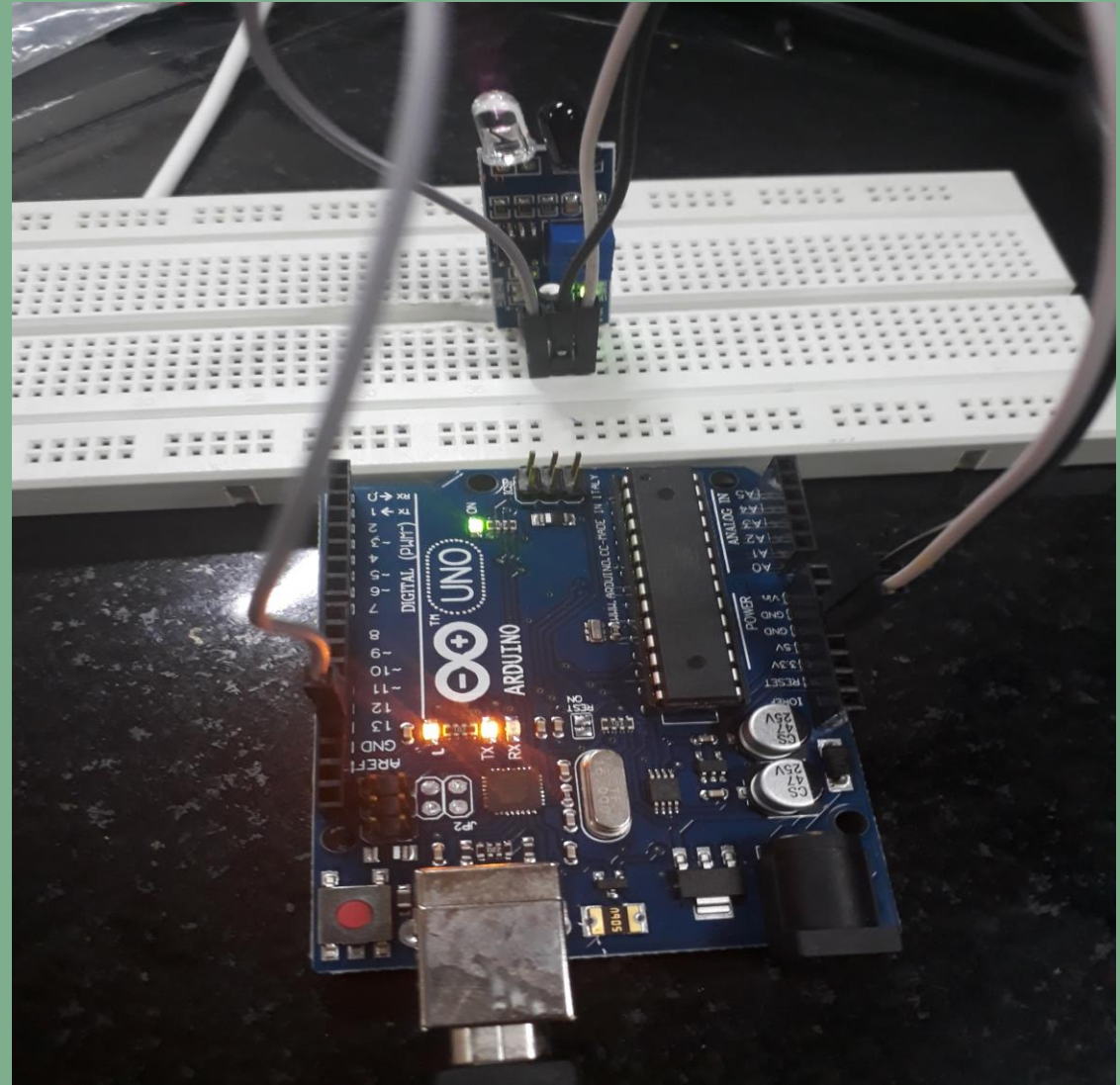
The IR photodiode is sensitive to the IR light emitted by an IR LED.

The photo-diode's resistance and output voltage change in proportion to the IR light received.



IR Sensor will give digital values
which can be read as:

```
a = digitalRead(pinName);
```



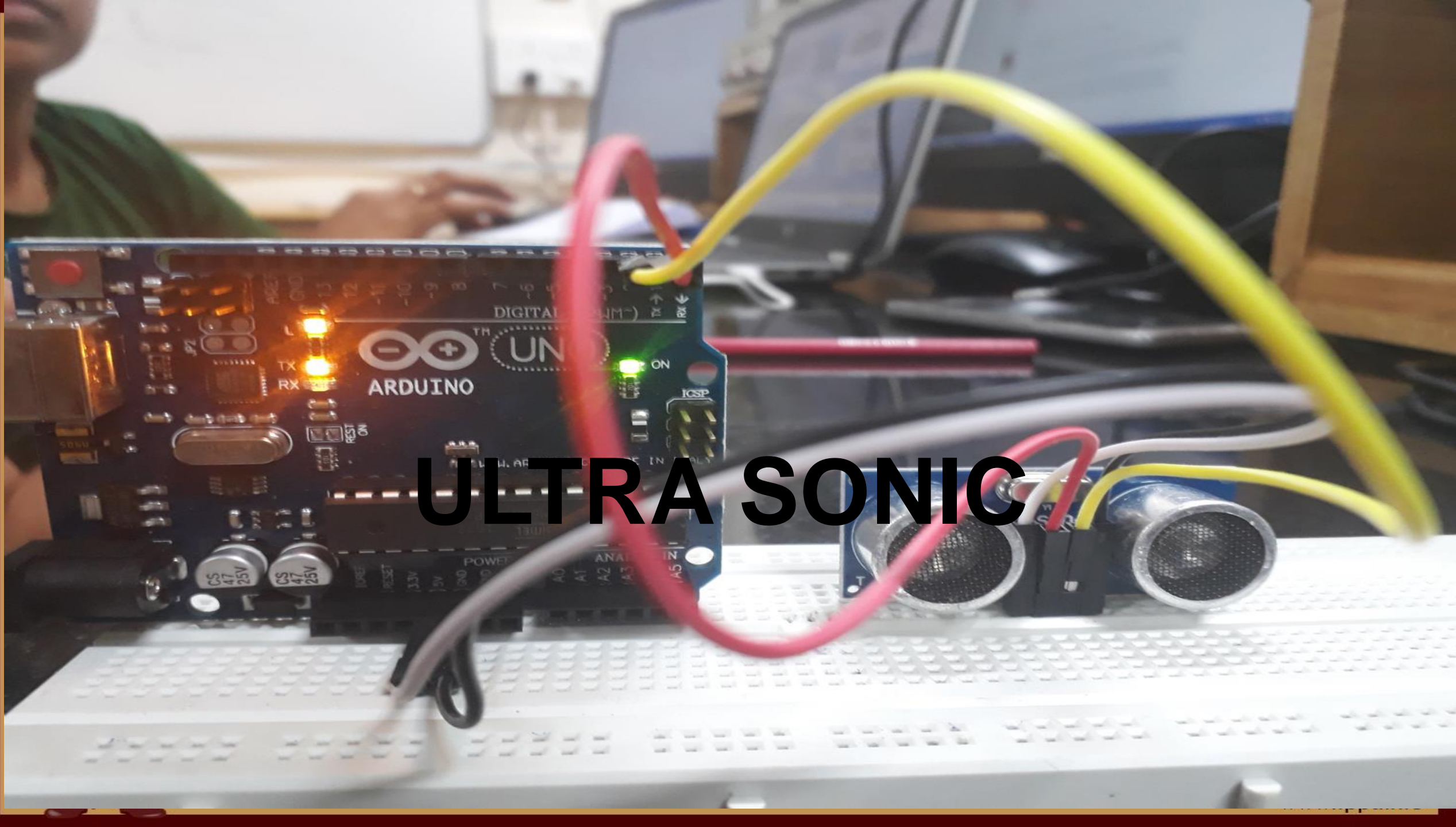


ir

```
int a;

void setup() {
  Serial.begin(9600);
  pinMode(13, INPUT);
}

void loop() {
  a = digitalRead(13);
  Serial.println(a);
}
```

ULTRA SONIC

ULTRASONIC SENSOR

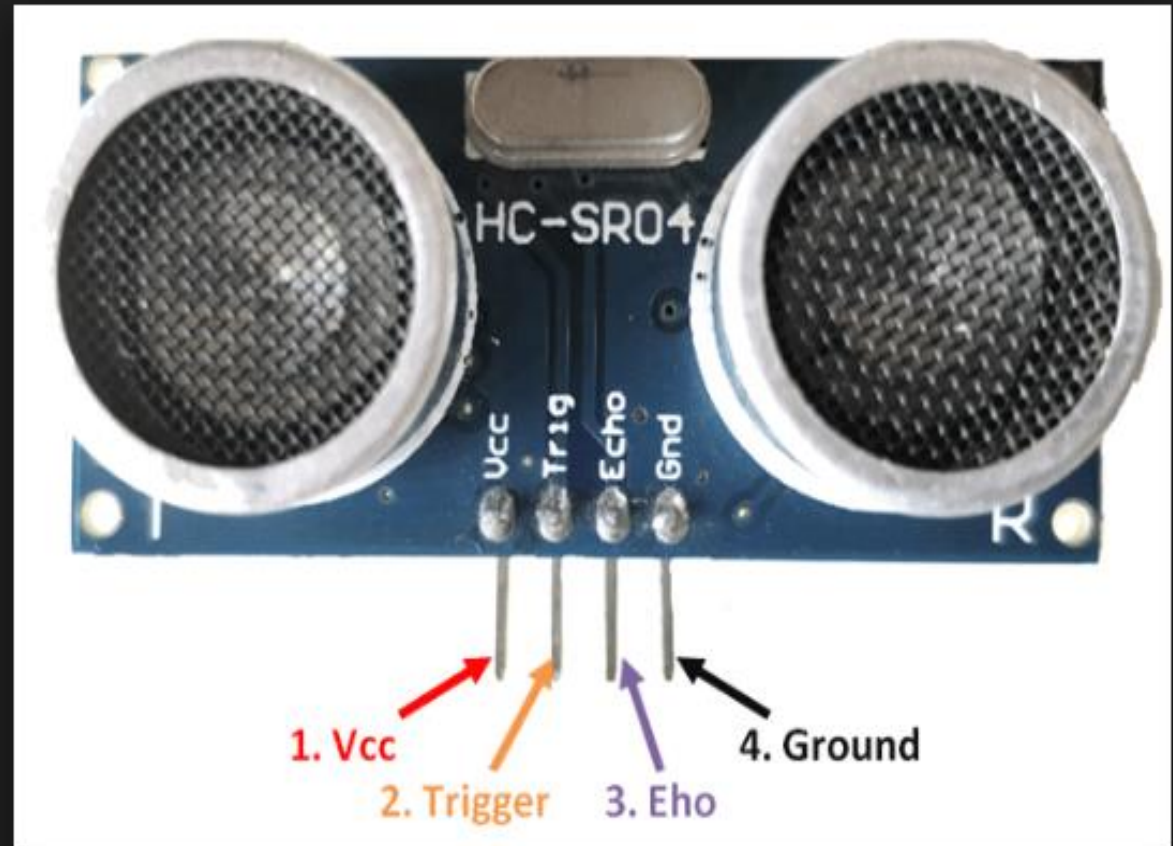
Trigger Pin

- Sends high frequency signal

Echo Pin

- Receive back the signal

Time is being recorded.



TotalTime measured by Ultrasonic Sensor is twice of the time taken by a signal(or pulse) to hit the object and revert back.

Time(in calculations) = TOTAL TIME / 2;



From Physics,

$$\text{Distance} = \text{Speed} \times \text{Time}$$

Ultrasonic measures time in Microseconds

$$1 \text{ Microsecond} = 1 \times 10^{-6} \text{ second}$$

Speed of Sound in air = 340m/s

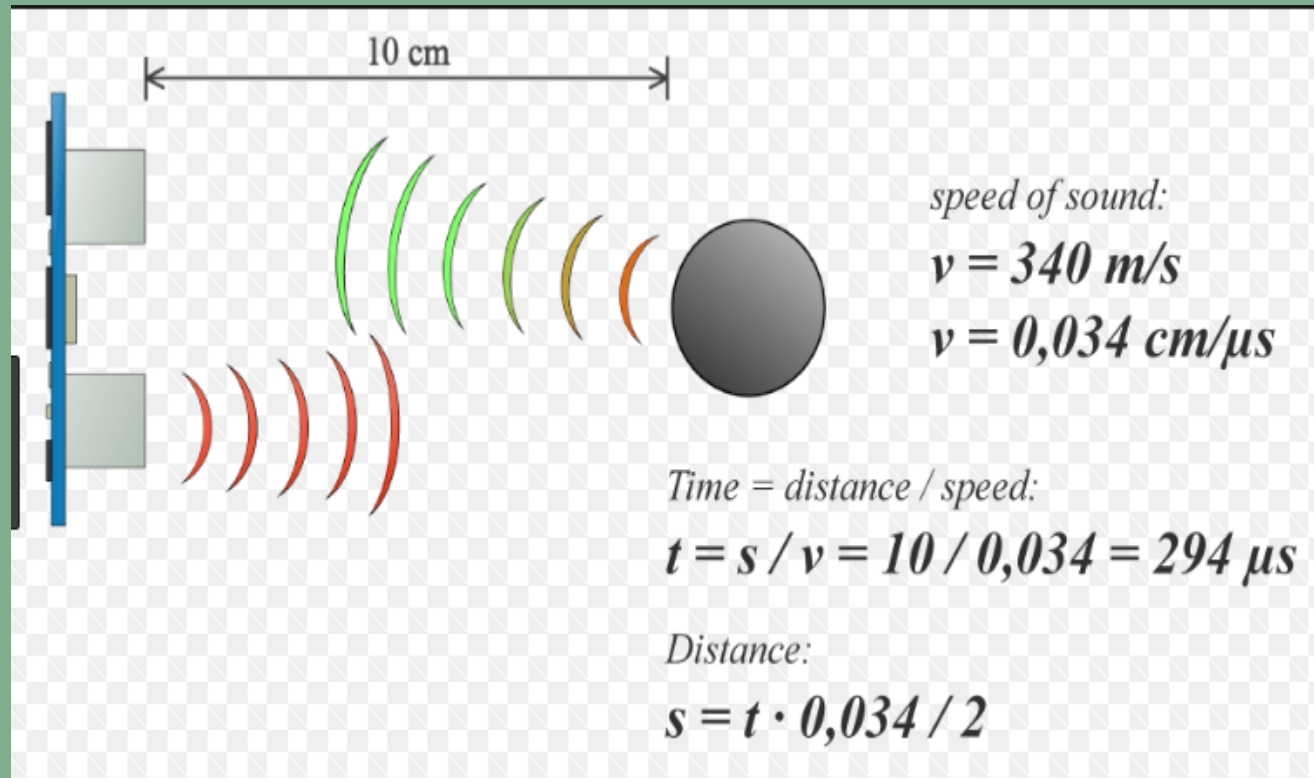
$$= 340 \times 100(\text{cm}) / (10^6(\text{microsecond}))$$

$$= 0.034 \text{ cm/microsecond}$$

$$\text{distance}(\text{cm}) = 0.034 (\text{cm/microsecond}) \times \text{time} / 2 (\text{Microsecond})$$



We will talk about PWM i.e. Pulse Width Modulation





ultrasonic

```
int triggerpin = 1;
int echopin = 2;
int distance, t;

void setup() {
  Serial.begin(9600);
  pinMode(triggerpin, OUTPUT); //TO ULTRA SONIC SENSOR
  pinMode(echopin, INPUT);    //FORM ULTRA SONIC SENSOR
}

void loop() {
  digitalWrite(triggerpin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin, HIGH);
  delayMicroseconds(10);
```





ultrasonic\$

```
digitalWrite(triggerpin, LOW);  
delayMicroseconds(2);  
digitalWrite(triggerpin, HIGH);  
delayMicroseconds(10);  
digitalWrite(triggerpin, LOW);
```

```
digitalRead(echopin);
```

```
t = pulseIn(echopin, HIGH);
```

```
distance = t*(0.034)/2;
```

```
Serial.println(distance);
```

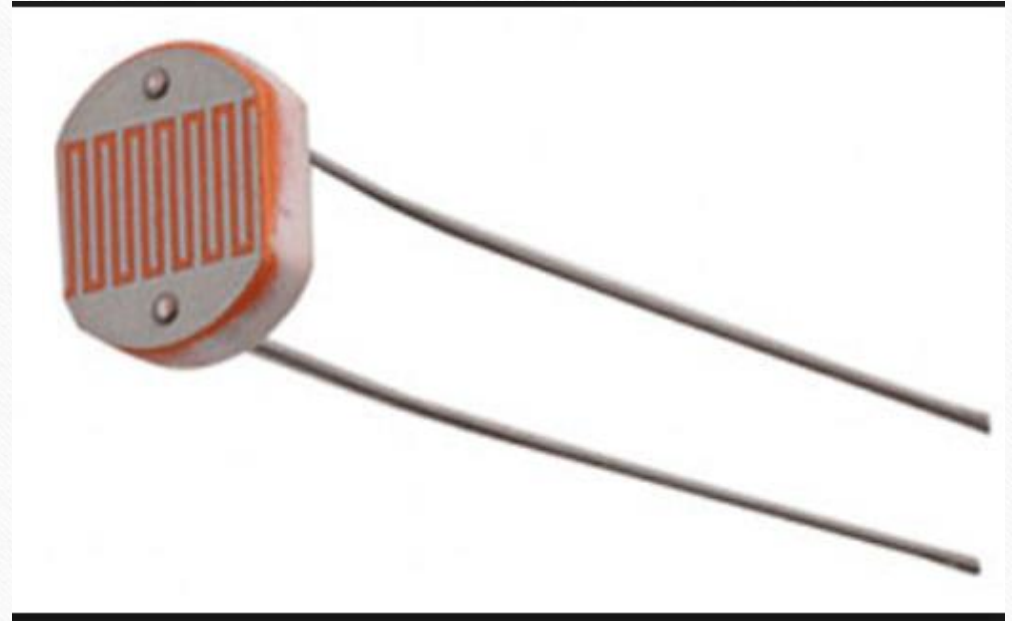
```
}
```

```
|
```

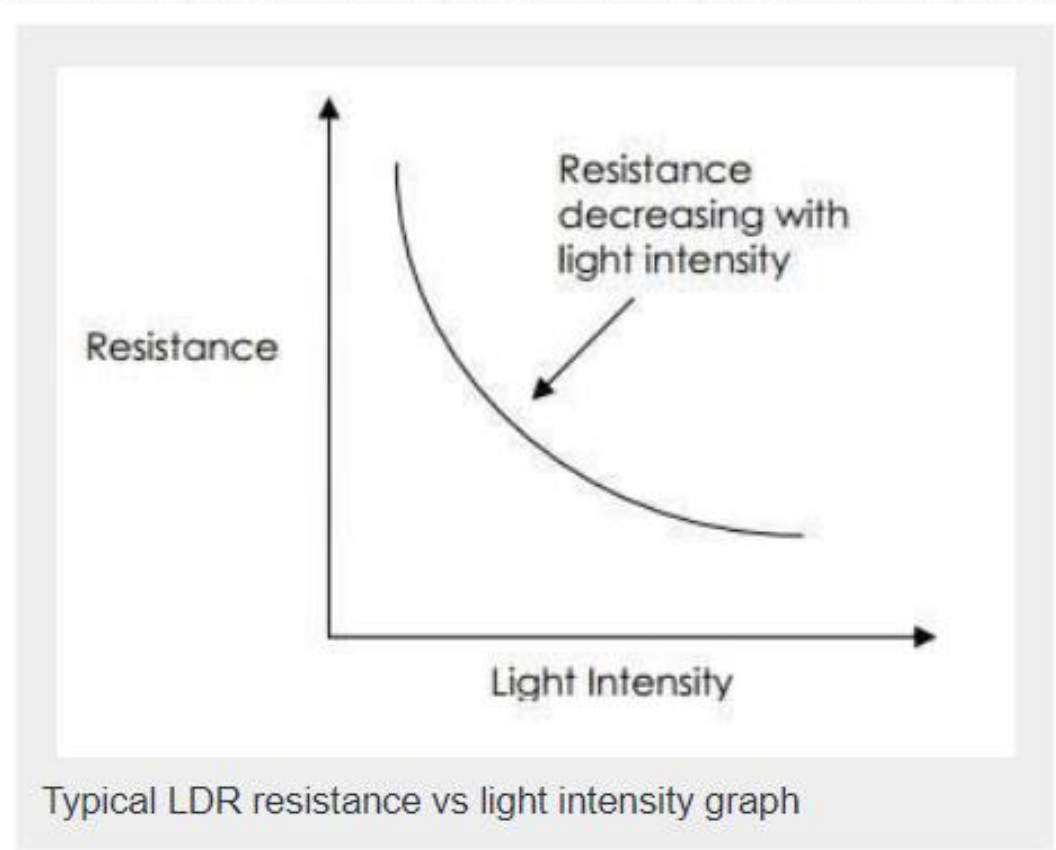
LDR

(Light Dependent Resistor)

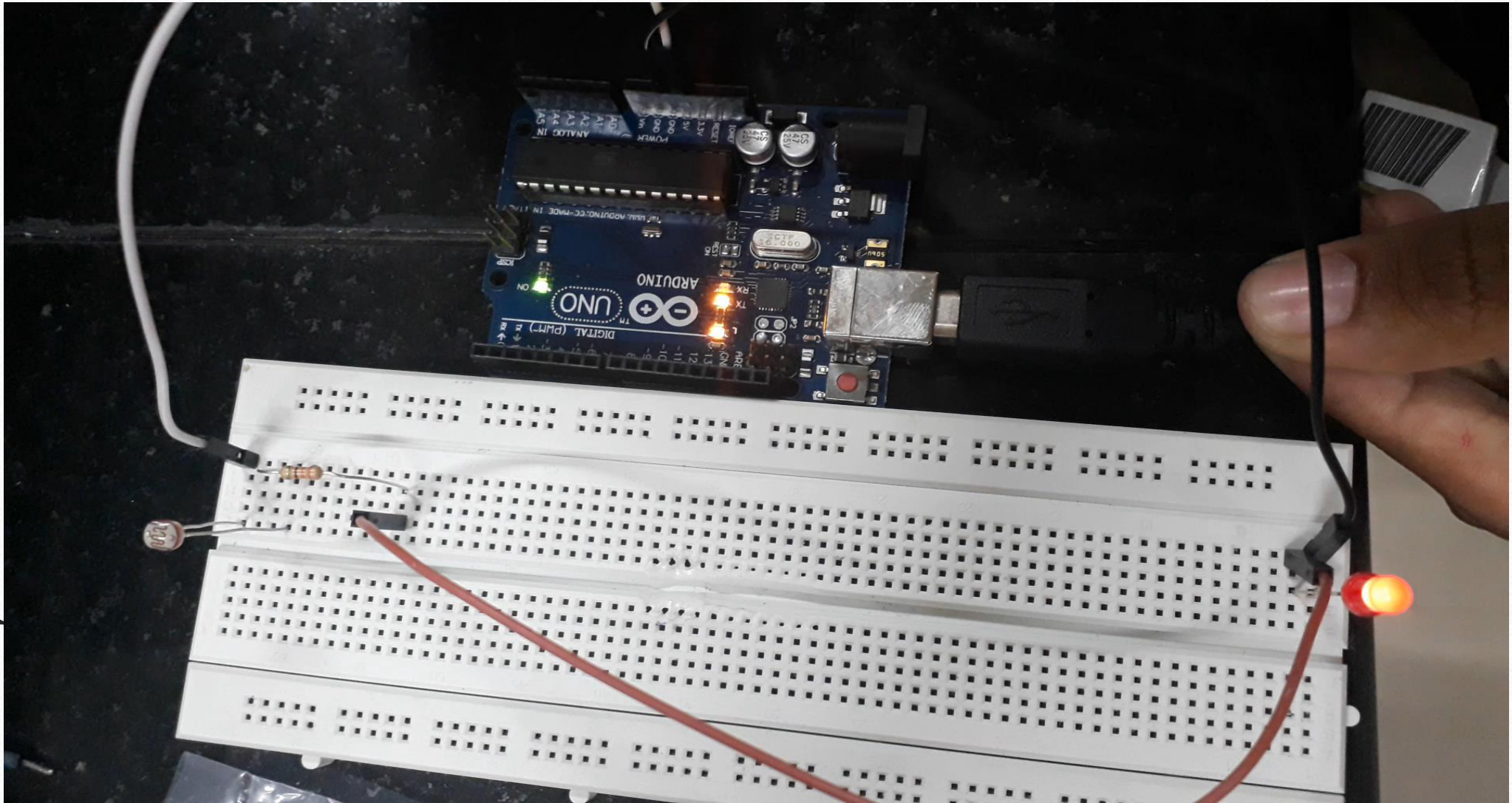
An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.



As light intensity changes,
the resistance also varies as
shown in the graph



LDR





Ildr

```
int a; //initializing an integer

void setup() {
  Serial.begin(9600); //Serial communication starts
  pinMode(A0, INPUT); //Analog PIN A0 is giving input from ldr
}

void loop() {
  a = analogRead(A0); //reading values from ldr
  Serial.println(a); // Printing those values
}
```




THANK YOU!!