Week 4 – Arduino Programming

Nurul Hazlina Noordin

UMP STEM LAB
See . Think . Explore . Marvel

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

Course Outcomes

	Outcomes
1	Cultivate attitude towards team working and professional ethics through mentoring sessions.
2	Demonstrate competency in application of technical aspect of Science, Technology, Engineering and Mathematics (STEM) concepts.
3	Design, construct and innovate STEM based projects.

Course Outline – UQB 2011

CO 3

mBlock
- week 2
- Computational Thinking

CO 3

Design Thinking
- week 3
- STEM outreach

CO 2

Arduino
- week 4
- Input outputs

CO 2

Coaching Technique
- week 5

CO 1

School Engagement
- week 6 - 13

Arduino & You

Please log into my padlet and answer these Qs:-

- What could an Arduino do?
- What would you like to know about Arduino?

https://padlet.com/nurulhazlina/ArduinoWeek4

W4 – Lesson Outline

ARDUINO

- Language
- Structure

1. Programming

3. Experiments

Digital Output

Exp 1
Blinking LED (out)

Digital Input

Exp 3
Push Button (in)
LED (out)

Analog Output

Exp 2
Dimmer LED (out)

Analog Input

Exp 4
LDR (in)
LED (out)

Exp 6
IR Sensor (in)
Motor (out)
Autonomous

Exp 5
Bluetooth to control Motor (out)

2. Electronic Components

Lesson Outcomes

1. Able to identify, construct electronic circuit using using a number of different sensors, actuators and communication media (wireless, internet);
2. Able to write programs in the programming environment "Processing",
3. Able con conduct experiments and trouble-shoot them;

Presentation Reminders :-

Learning Outcomes

Take note

Hands-on

PRE-LEARNING PREPARATION

Please ensure that you have the following:

1 PERSONAL COMPUTER

Running Windows, Linux or MacOS with a USB port

2 ARDUINO BOARD with USB Cable

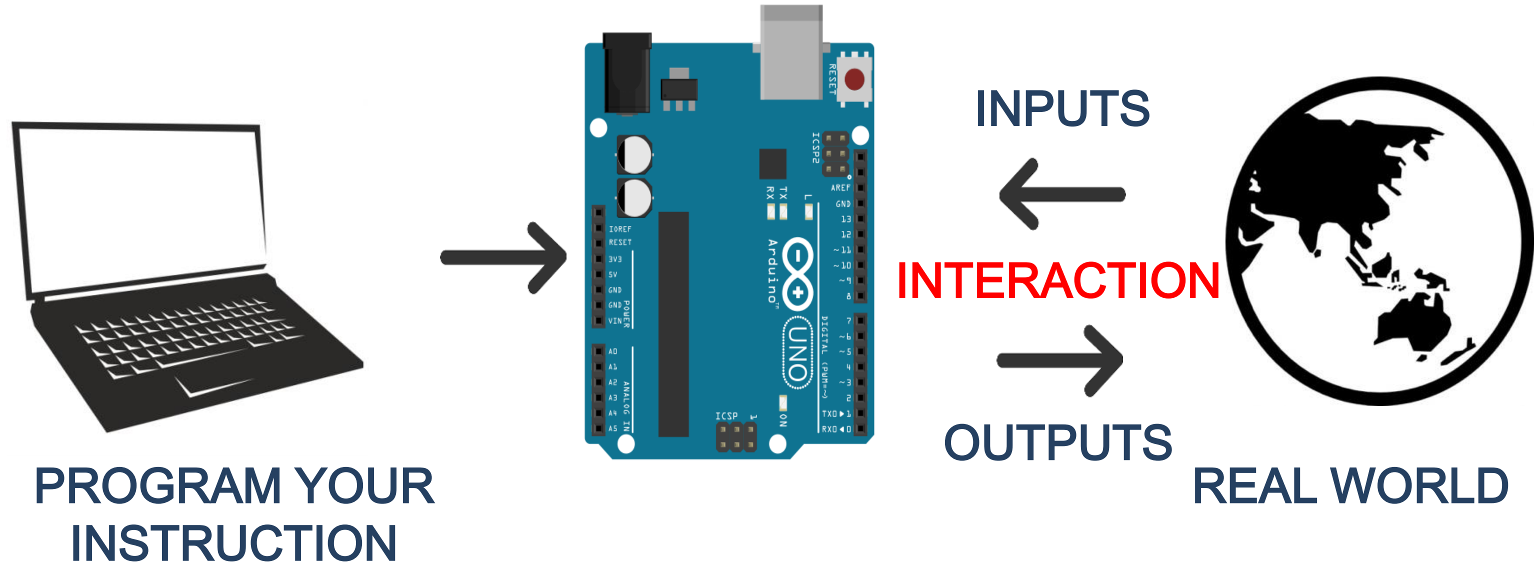
This guide uses UNO, but you can use any version of the ARDUINO board out there

3 ELECTRONIC COMPONENTS

Contains all necessary components and parts for all exercises

4 mBlock and ARDUINO SOFTWARE

- Referred to as an Integrated Developers Environment (IDE).
- Download the latest version according to your operating system (Windows, MacOS or Linux) at http://arduino.cc/en/main/software
- Once downloaded, click the executable file and follow the instructions
- A shortcut will be create on your desktop along with an Arduino folder in Mydocument

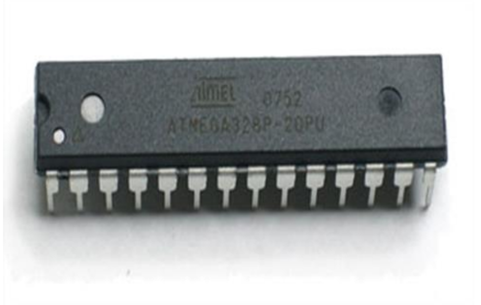


Microcontrollers are **dedicated** to one task and run one specific program

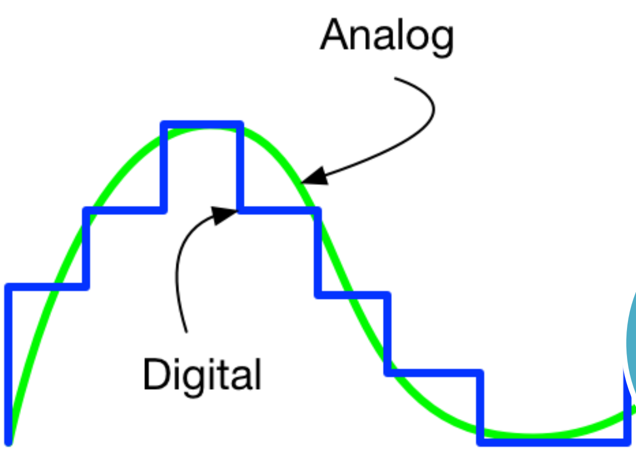
Examples of tasks could be:

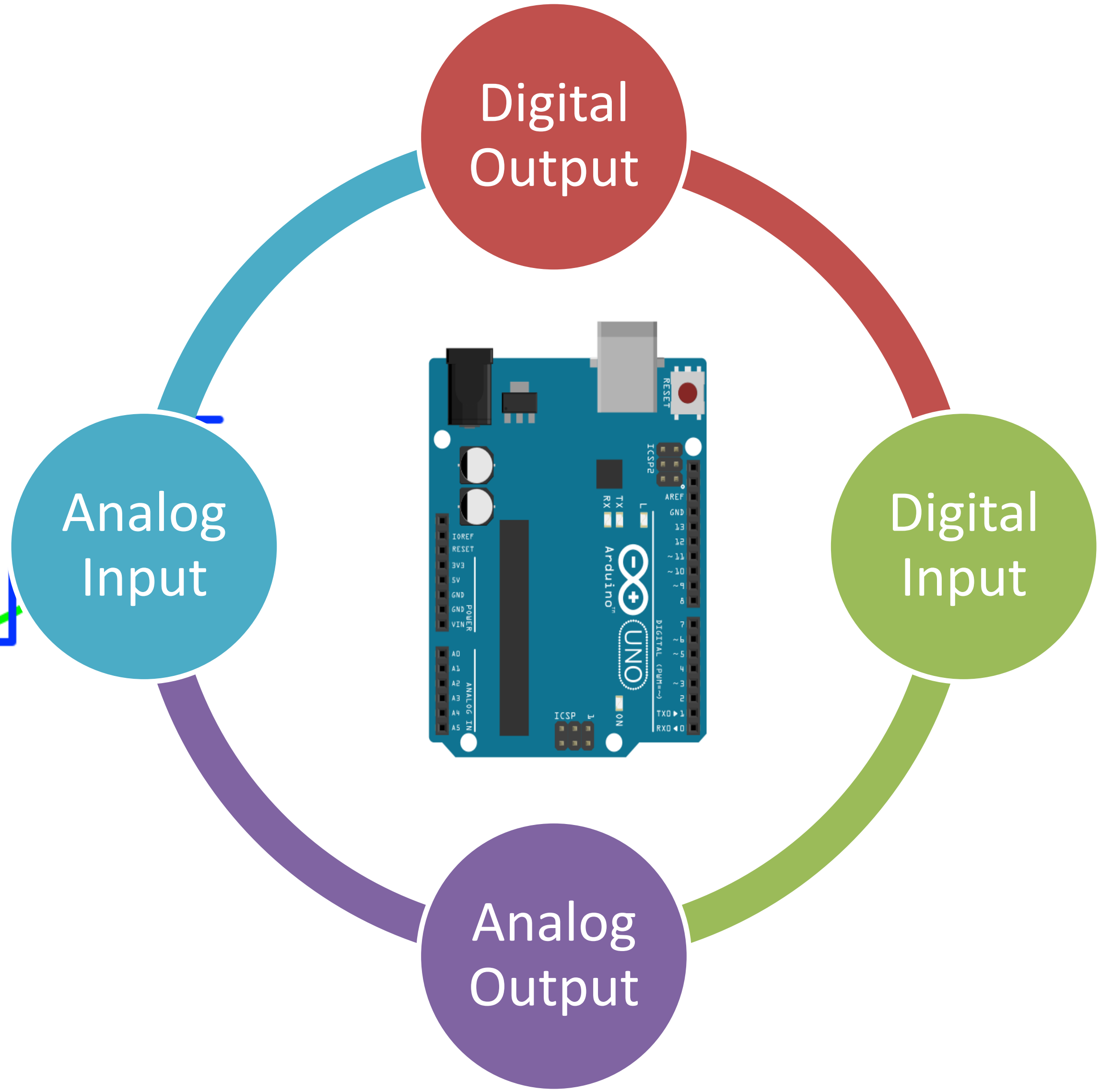
i. **Received from inputs** via ports (read from external hardware)
ii. **Process the data,** store in file registers, arithmetic operations (added, subtracted, logic gates), etc.
iii. **Control outputs** (control hardware)

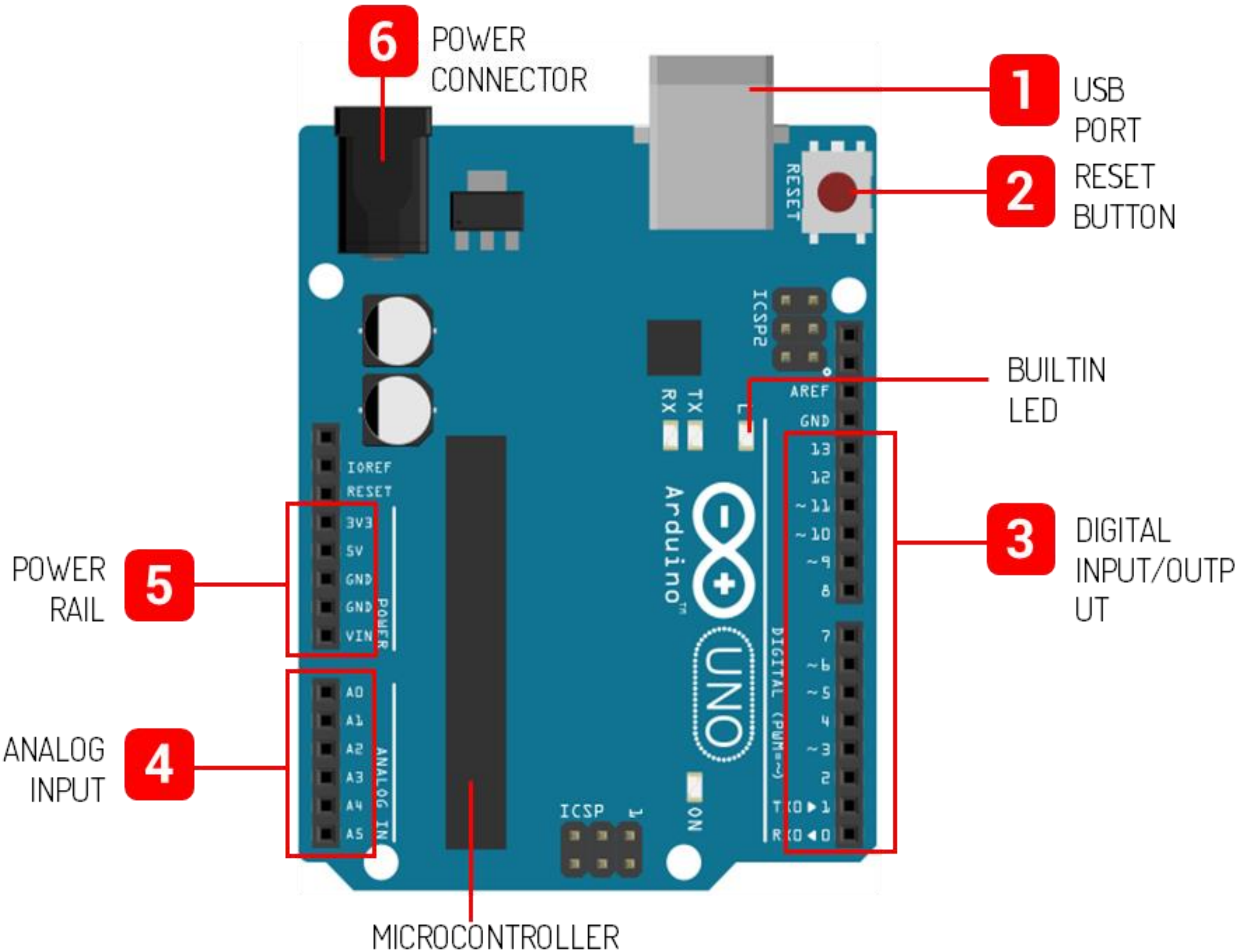
Processor, Storage and RAM all in one tiny package



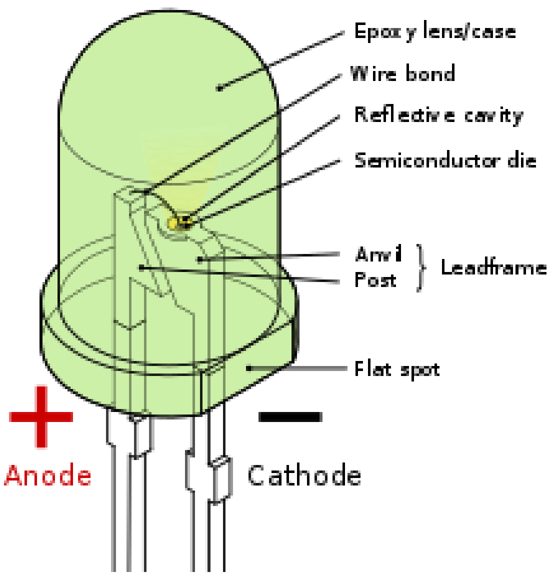
Signals


Analog
Digital


Digital Output
Digital Input
Analog Output
Analog Input


6 POWER CONNECTOR
1 USB PORT
2 RESET BUTTON
BUILTIN LED
3 DIGITAL INPUT/OUTPUT
POWER RAIL 5
ANALOG INPUT 4
MICROCONTROLLER

Electronic Components

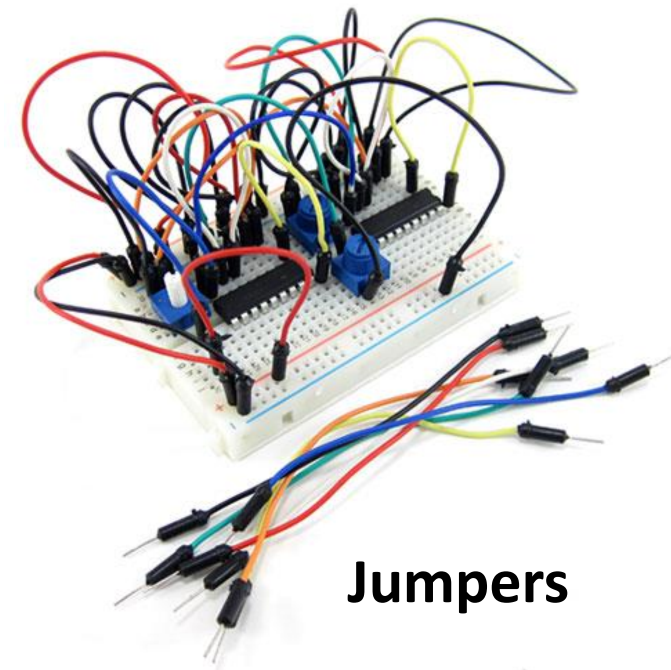


LED

Push-button

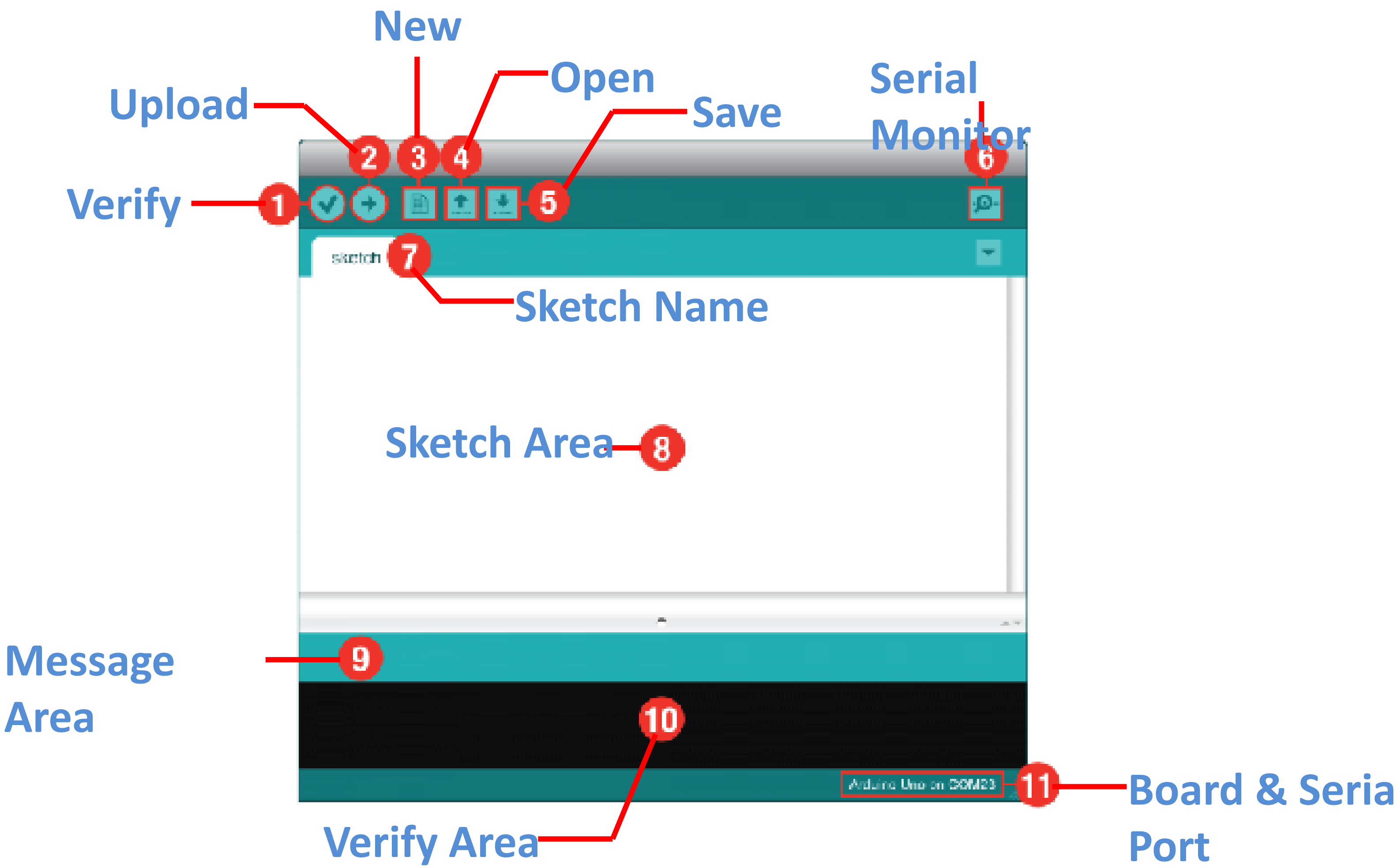
Resistors

Breadboard



Jumpers

Arduino IDE


New
Open
Upload
Save
Serial Monitor
Verify
1
2
3
4
5
6
7
sketch
Sketch Name
Sketch Area
8
Message Area
9
10
11
Board & Serial Port
Verify Area

Programming Structure

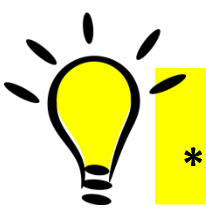
```
void setup() {
  // put your setup code here, to run once:

}

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

}
```

Setup( ) function

- initialize the variables, pin modes, start using libraries, etc.
- run once, after each power up or reset of the Arduino board.

Loop( ) function

- loops consecutively
- use it to actively control the Arduino board.



*** Case sensitive**
*** Always save file**

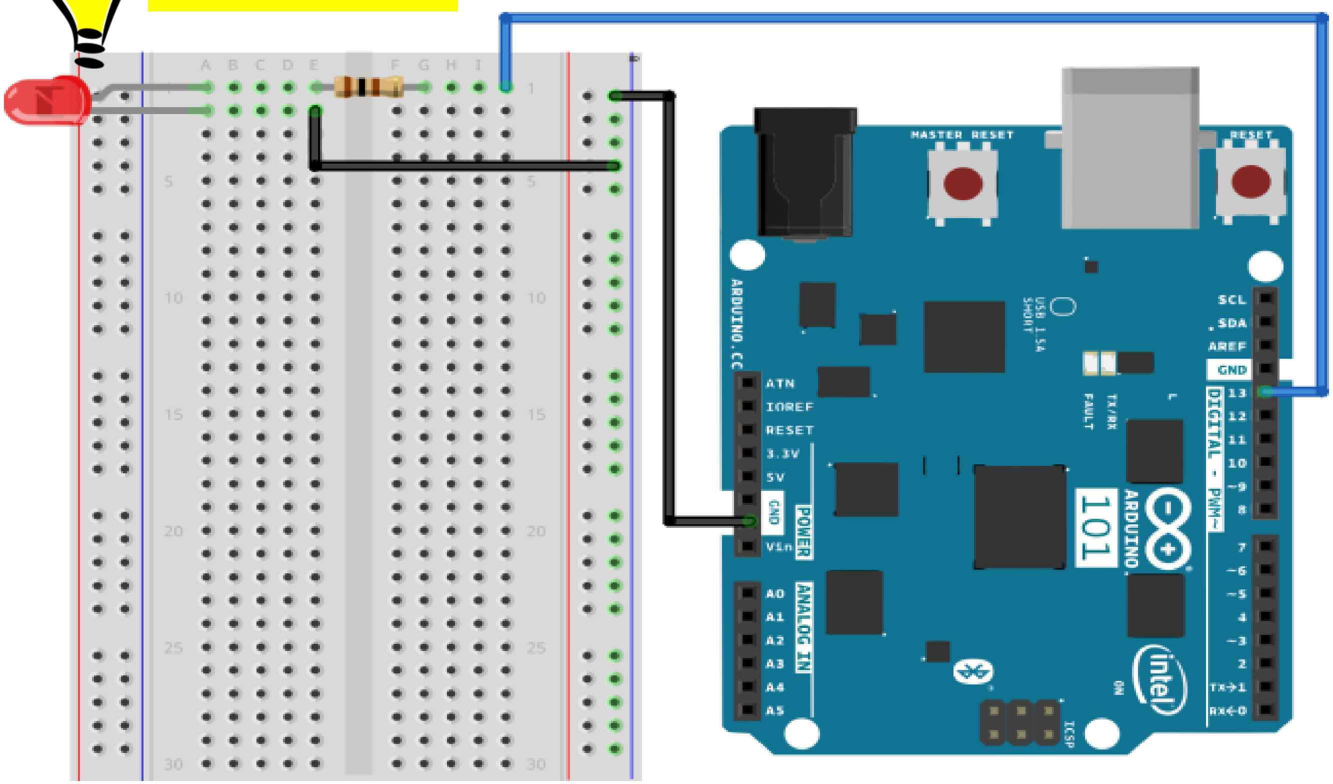
Objective:

to write a sequential Arduino instruction – LED ON, LED Blinks

1. DIGITAL OUTPUT

Circuit Diagram

***LED polarity**

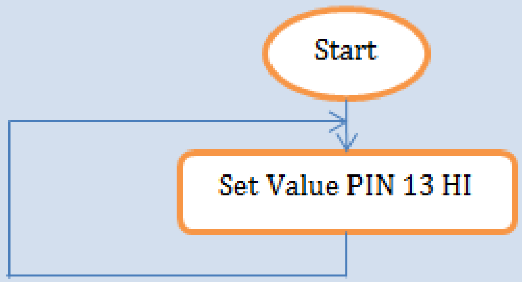


Resistor to Pin 13
Resistor → +LED
Negative LED →GND

Resistor **150** Ω (Brown- Green- Brown)
LED = Flag refer to negative

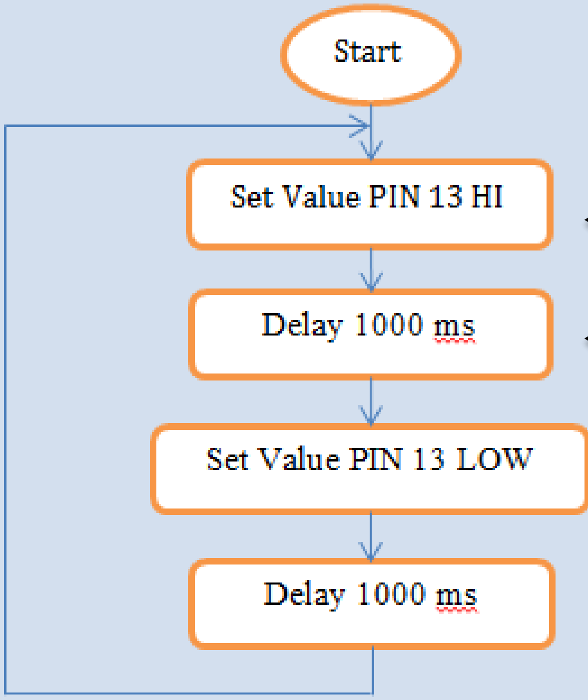
Construct the Circuit

Programming 1 – LED ON

Pseudo-code	Flow Chart
Pin in use 13 **1. ON Pin LED**	

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

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
}
```

Programming 2– LED Blinks

Pseudo-code	Flow Chart
1. ON Pin LED 2. Hold the value 1000 seconds 3. Set the value LOW 4. Hold the value 5. Repeat	

digitalWrite(pin, value);

delay(value);

Write your codes

LO 2 – Write Codes

Programming 2 – LED Blinking

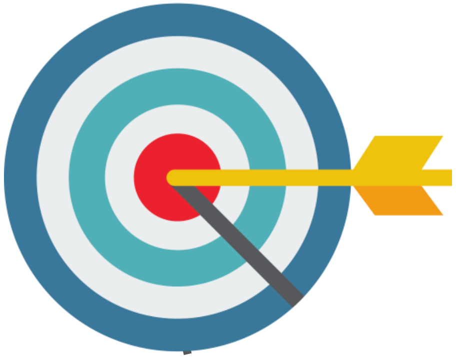
```
void setup()
{
  pinMode(13,OUTPUT);
}

void loop()
{
  digitalWrite(13,HIGH);
  delay(2000);
  digitalWrite(13,LOW);
  delay(2000);
}
```

We are turning on a light at pin 13, thus set it as OUTPUT

LED turn ON

Wait for 2 seconds

Codes placed in loop will be repeated



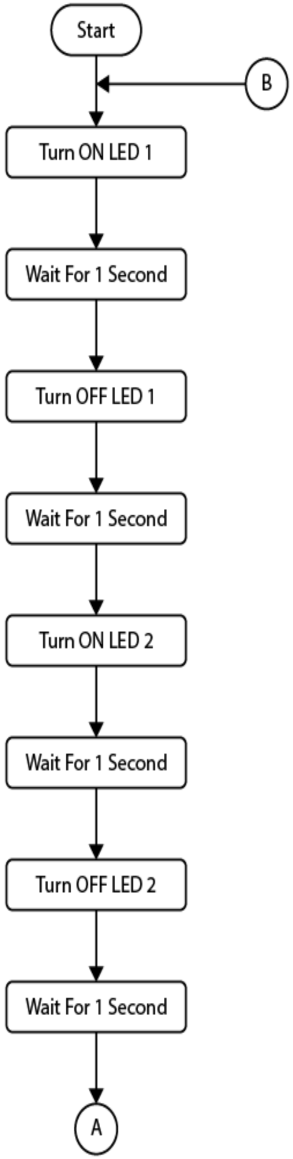
Try It Out

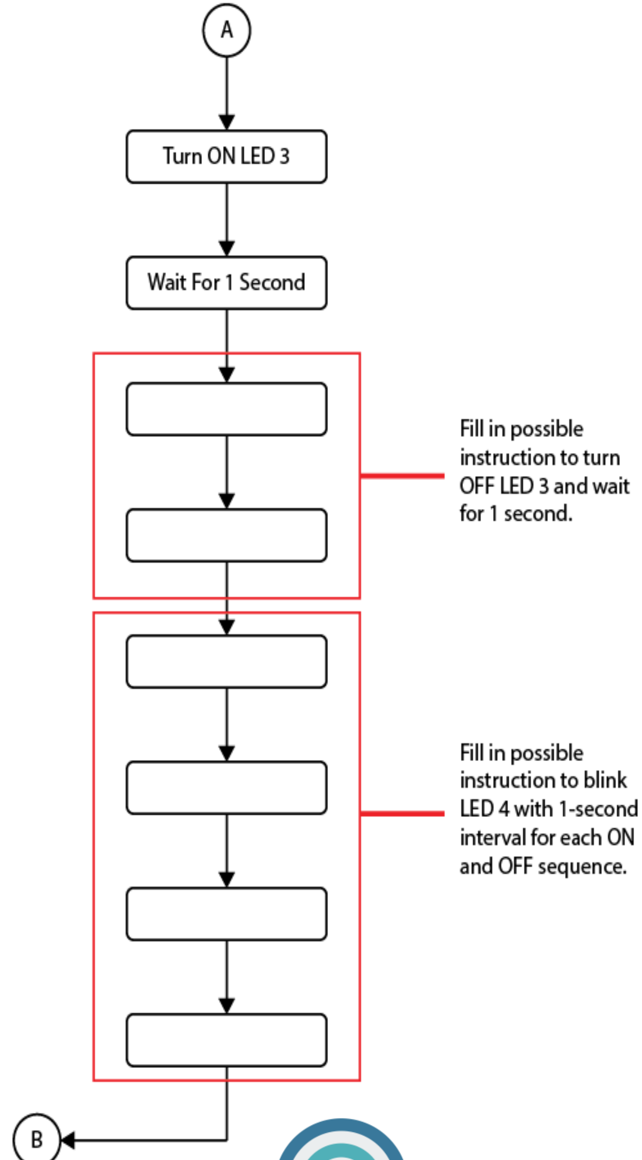
Change to different pin by moving jumper wire & edit the sketch

Make the ON time longer than the OFF time

Make the LED blink faster or slower by editing the delay value.

LO 3 – Conduct Experiments





Complete the chart

Programming 3 – Running Lights

```
void setup()
{
    pinMode(13,OUTPUT);
    pinMode(12,OUTPUT);
    pinMode(11,OUTPUT);
    pinMode(10,OUTPUT);
}
```

Set pin as OUTPUT

```
void loop()
{
    digitalWrite(13,HIGH);
    delay(1000);
    digitalWrite(13,LOW);
    delay(1000);
```

Pin13 LED turns ON

Pin13 LED turns OFF

Wait for 1 second

```
    digitalWrite(12,HIGH);
    delay(1000);
    digitalWrite(12,LOW);
    delay(1000);
```

Complete the code to blink the LED connected to pin 11 with 1 second interval in between

```
    digitalWrite(10,HIGH);
    delay(1000);
    digitalWrite(10,LOW);
    delay(1000);
}
```

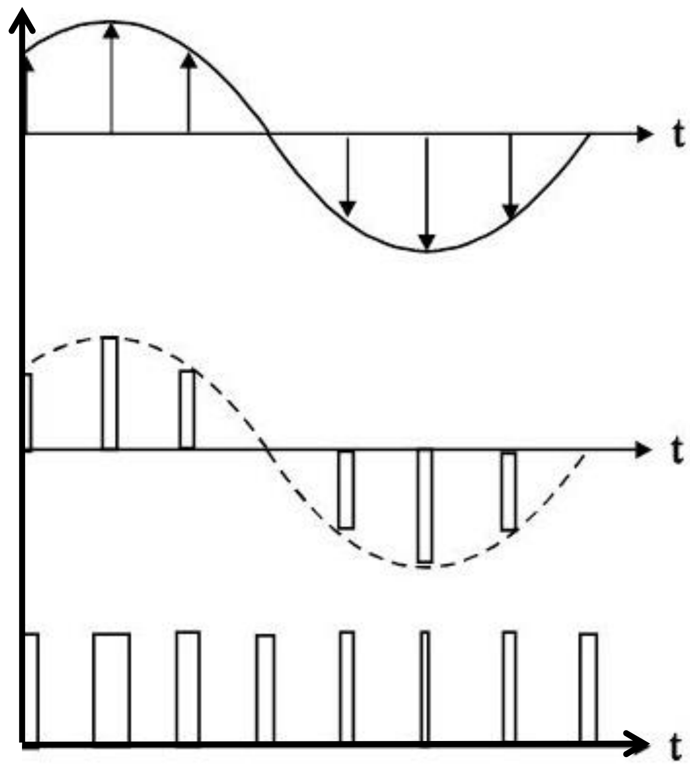
Objective:

- to implement PWM concept in electronic circuits
- to integrate analogWrite () for PWM signals
- to use for loops

2. ANALOG OUTPUT

Analog Output - PWM

Analog Signal Representation



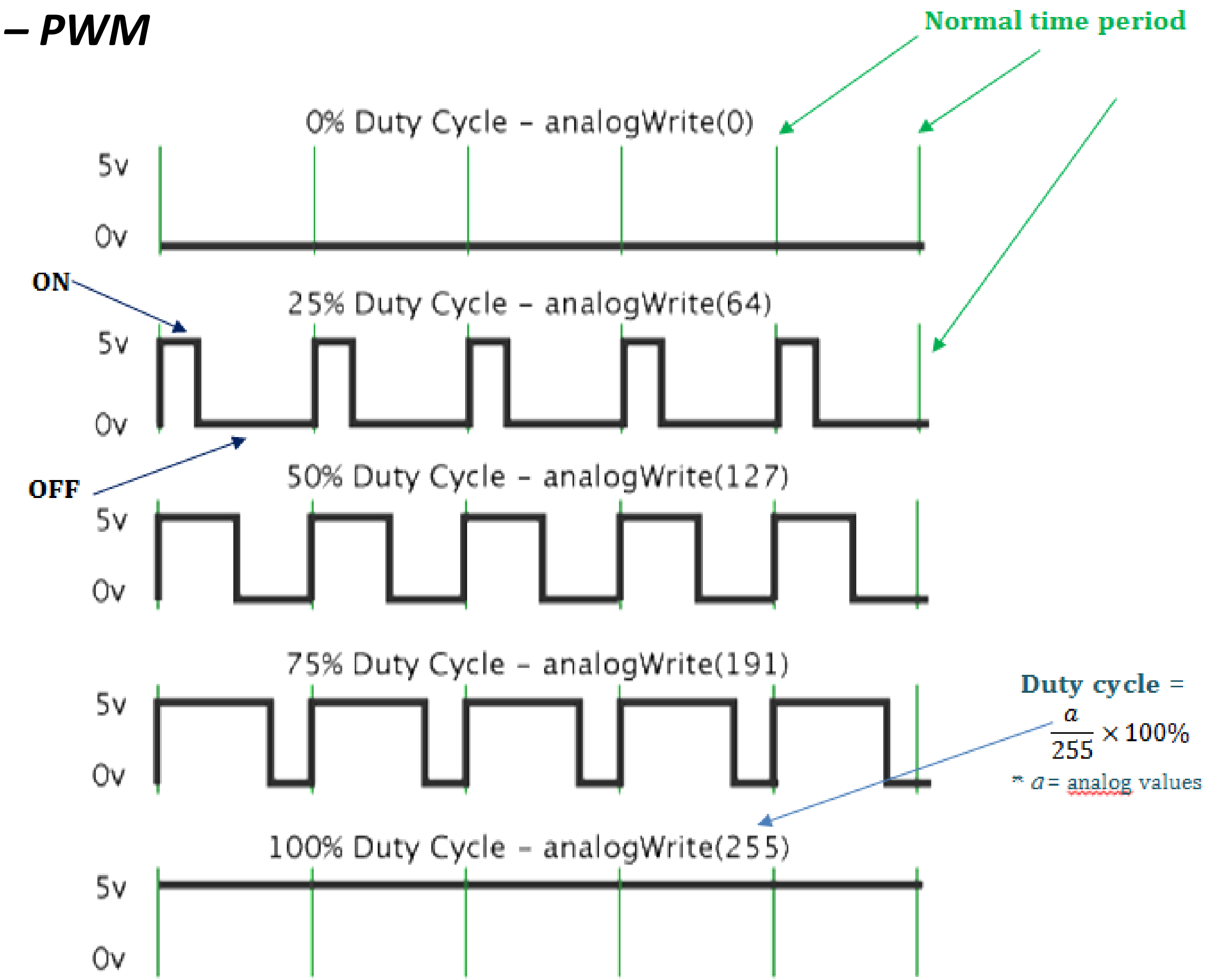
Analog Signal

Pulse Amplitude Modulation

Pulse Width Modulation

Analog Output - PWM

Pulse Width Modulation – PWM



Analog Output PWM

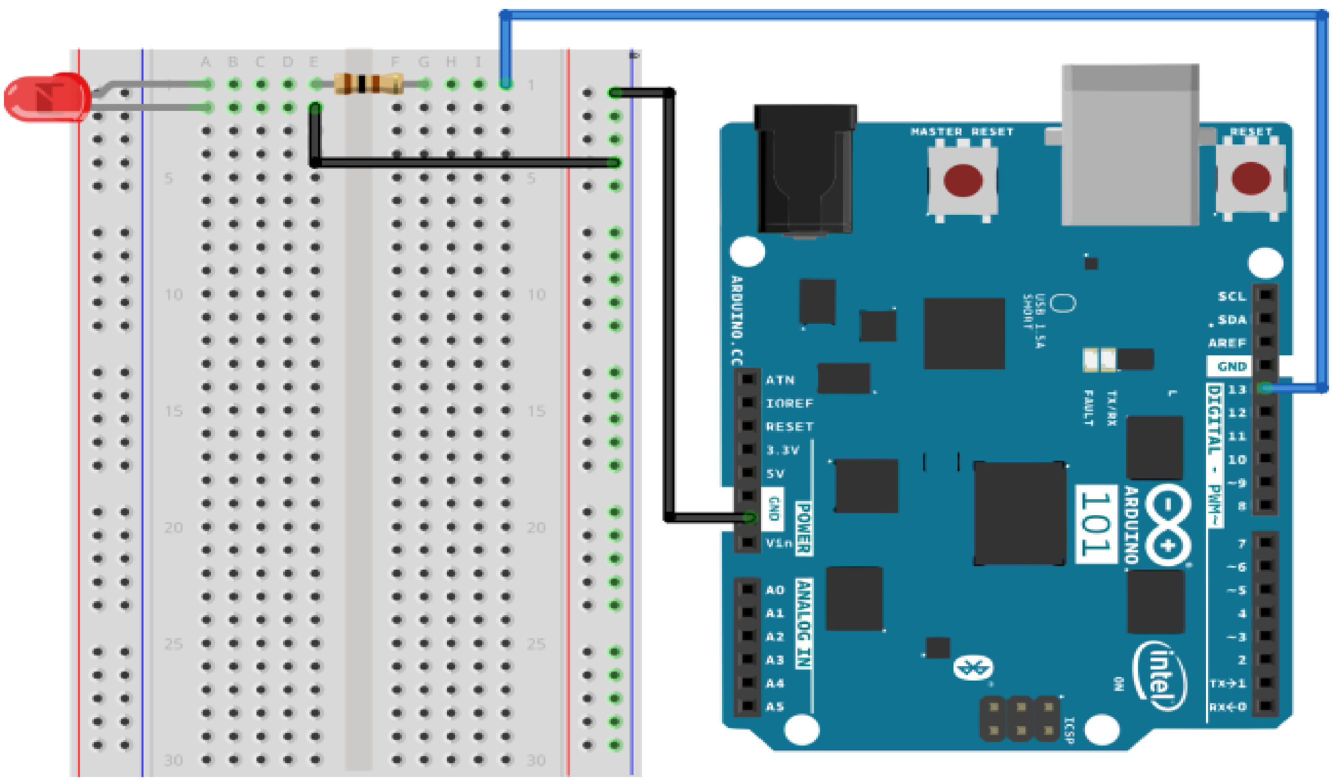
Arduino Uno Pin Assignment – Analog Output

analogWrite (pin, value);

Pin = 3,5,6,9,10,11

Different Arduino would have different ADC values

Circuit Diagram Programming 4



Note that the pin changes – WHY?

Resistor to **Pin 11**
Resistor → +LED
Negative LED →GND

Resistor **150** Ω (Brown-Green- Brown)
LED = Flag refer to negative

Construct the Circuit

LO 3 – Conduct Experiments

Programming 4 – Fading Light

for (initialization; condition; increment) { //statement(s); }

Pseudo-code

START

1. Set LED Pin 11 as Output
2. Set value of pin 11 as 5
3. Hold the value
4. Set value of pin 11 as 10

.

. Repeat until max value 255

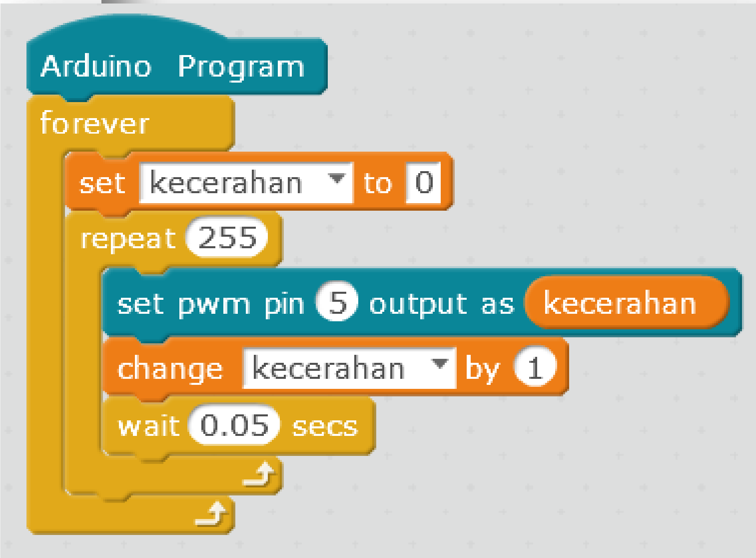
END

1.Draft your flowchart
2.Sketch your Arduino Codes

```
void loop()
{
  for (int i=0; i <= 255; i++){
    analogWrite(PWMpin, i);
    delay(10);
  }
}
```

Programming 4 – Fading Light

```
int ledPin = 6;    // LED connected to digital pin 9

void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
```

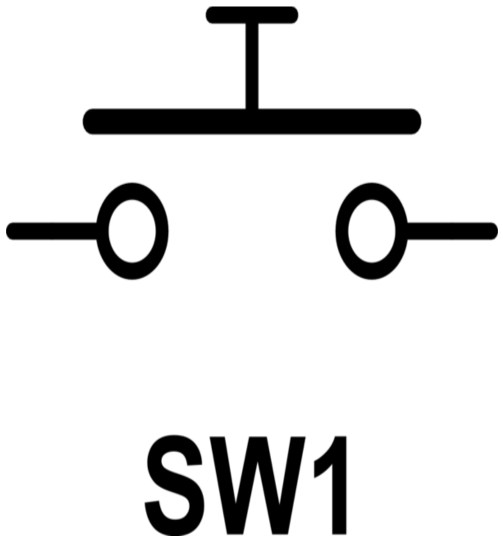


Objective:

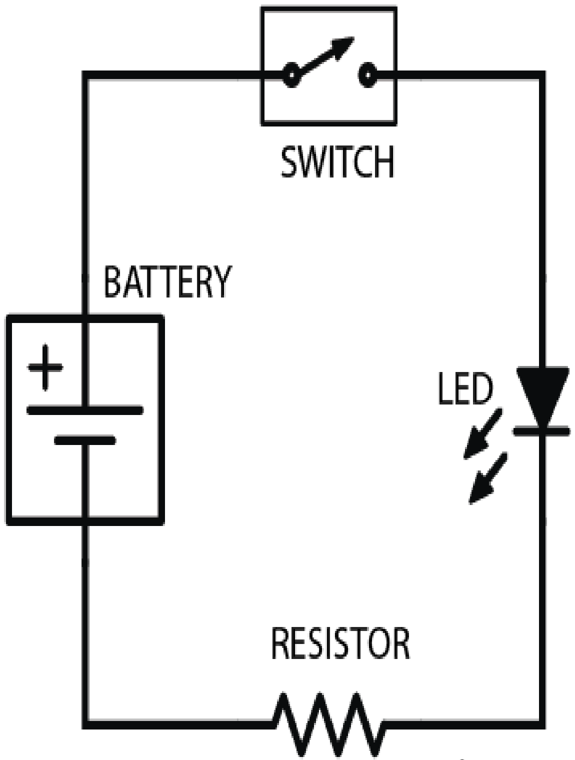
- adopt conditional statement in Arduino programming

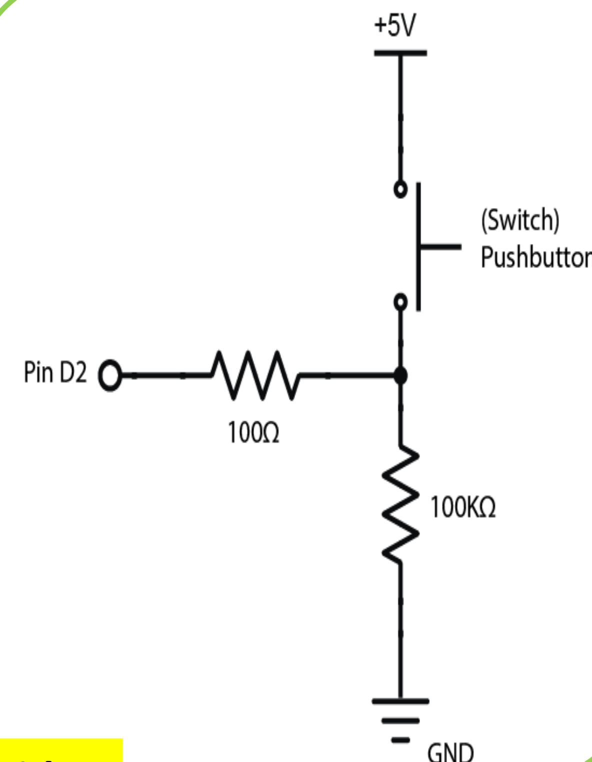
3. DIGITAL INPUT DIGITAL OUTPUT

Push Button – digitalRead



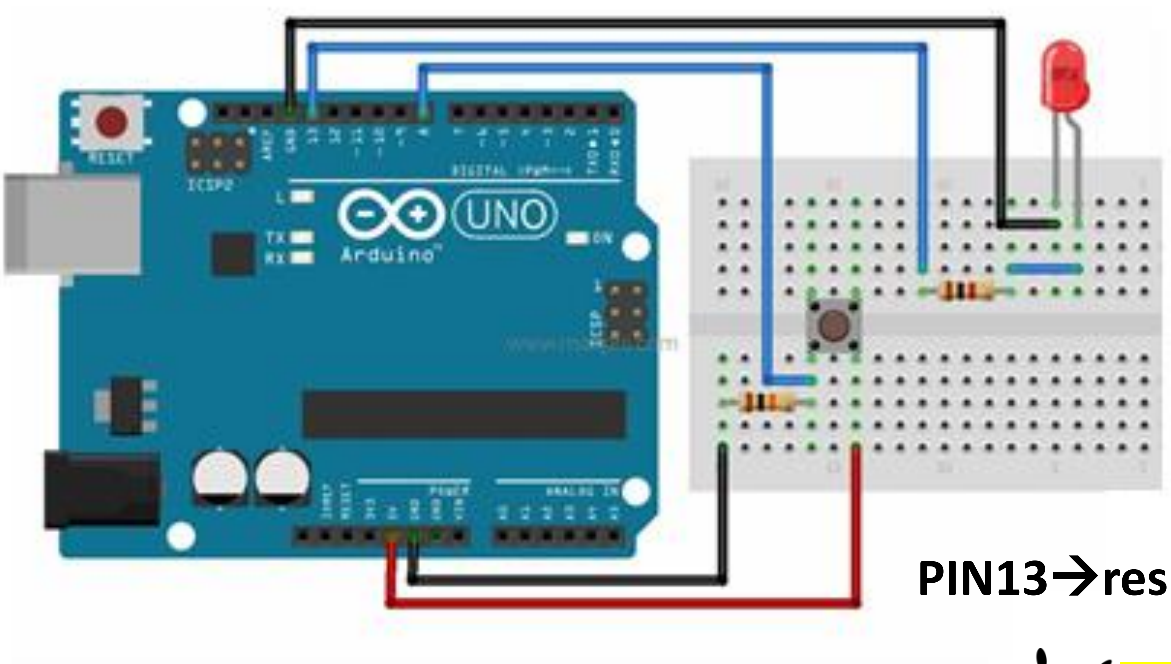
The LED will only light up if and when the switch is pressed as it will complete the circuit (closed loop)





Voltage divider

Circuit Diagram



LED = PIN13

LED resistor (150 Ω)

PIN13→resistor(150 Ω) → -ve LED→+ve LED→GND

Voltage divider – WHY?

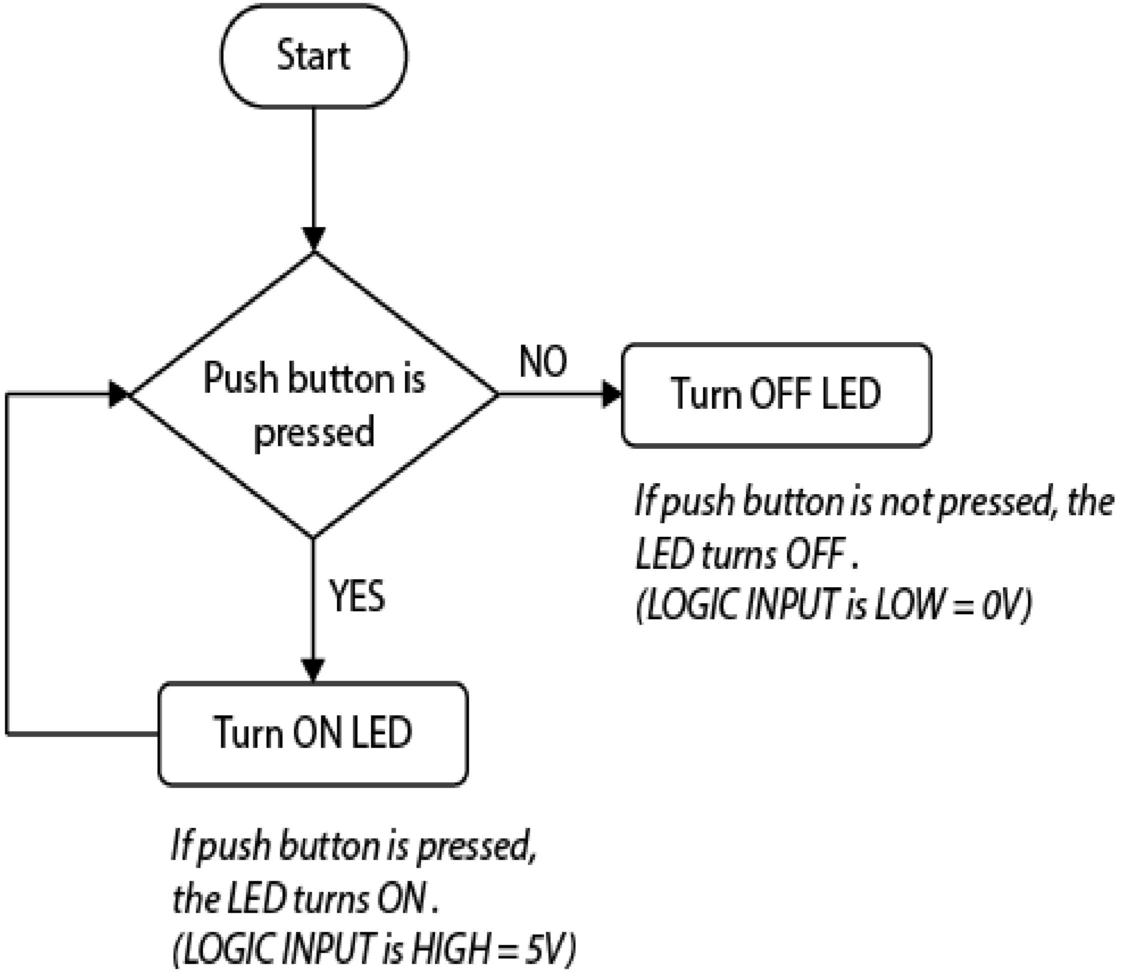
Voltage Divider at Switch

Resistor 1 kΩ

5V to PINswitch1

PINswitch2 to PIN8

1. Construct the Circuit
2. Sketch the flow chart


Start
Push button is pressed
NO
Turn OFF LED
YES
Turn ON LED
If push button is not pressed, the LED turns OFF.
(LOGIC INPUT is LOW = 0V)
If push button is pressed, the LED turns ON.
(LOGIC INPUT is HIGH = 5V)

```
void setup(){
  pinMode(9,INPUT);
  pinMode(5,OUTPUT);
}
```

Setup Digital pin 9 as input and setup Digital pin 5 as output.

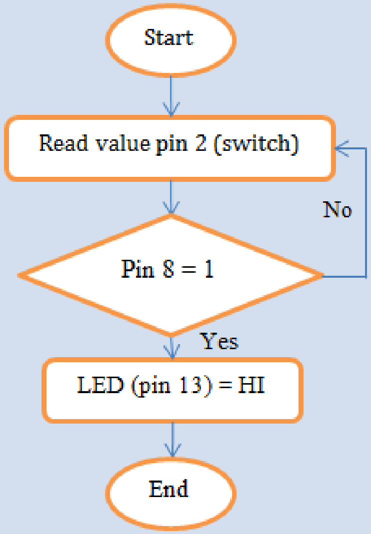
```
void loop()
{
  if(digitalRead(9) == HIGH)
  {
    digitalWrite(5,HIGH);
  }
  else
  {
    digitalWrite(5,LOW);
  }
}
```

Conditional statement:
To read Digital pin status, if it's `HIGH`,

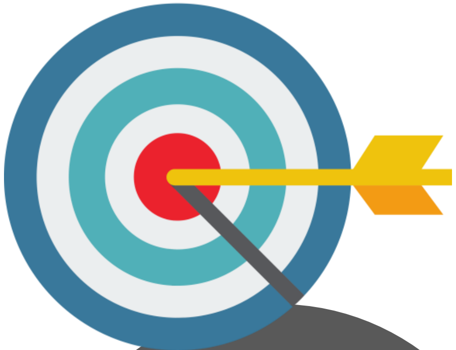
the action is to turn ON LED on Digital pin 5.

if the Digital pin status is other than `HIGH`, the action is to turn OFF LED on Digital pin 5.

Programming – Push Button

Pseudo-code	Flow Chart
1. Declare a. Switch as Input (pin 8) b. LED as output (pin 1) 2. If the switch is pressed, LED ON, otherwise LED off	


```
const int switch1=8;
const int LED=13;
int currentstate = LOW;

void setup()
{
  pinMode (switch1,INPUT);
  pinMode (LED,OUTPUT);
}

void loop()
{
  currentstate=digitalRead(switch1);
  if (currentstate==HIGH)
    digitalWrite (LED,HIGH);
  else
    digitalWrite (LED,LOW);
 delay (10);
}
```



Try It Out

Change the digital pins to connect LED and Pushbutton.

Add a buzzer that will be turned ON whenever the pushbutton is pressed.

Modify sketch when push release ON, then push release again will OFF the LED.

Answers

```
int inPin = 2;         // the number of the input pin
int outPin = 13;       // the number of the output pin

int state = HIGH;      // the current state of the output pin
int reading;           // the current reading from the input pin
int previous = LOW;    // the previous reading from the input pin

// the follow variables are long's because the time, measured in
miliseconds,
// will quickly become a bigger number than can be stored in an int.
long time = 0;         // the last time the output pin was toggled
long debounce = 200;   // the debounce time, increase if the output
flickers

void setup()
{
  pinMode(inPin, INPUT);
  pinMode(outPin, OUTPUT);
}
```

Answers

```
void loop()
{
  reading = digitalRead(inPin);

  // if the input just went from LOW and HIGH and we've waited long enough
  // to ignore any noise on the circuit, toggle the output pin and remember
  // the time
  if (reading == HIGH && previous == LOW && millis() - time > debounce) {
    if (state == HIGH)
      state = LOW;
    else
      state = HIGH;

    time = millis();
  }

  digitalWrite(outPin, state);

  previous = reading;
}
```

DATA TYPES

Arduino Data Types

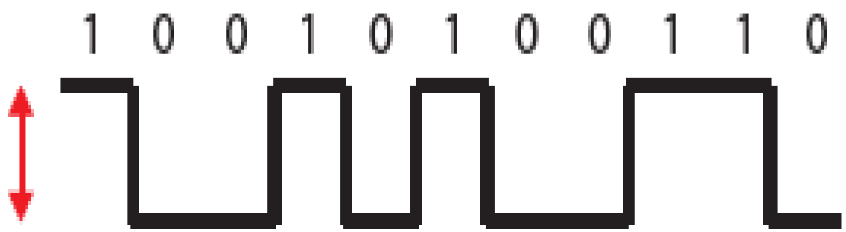
DATA TYPES	VALUE RANGES
int	-32768 ***to*** 32767
float	-3.4028235 E38 ***to*** 3.4028235 E38
long	-2,147,483,648 ***to*** 2,147,483,647
unsigned int (no negative value)	0 ***to*** 65535
char	-127 ***to*** 128
word	0 ***to*** 65535
byte	0 ***to*** 255
boolean	True ***or*** False
unsigner char	0 ***to*** 255

Comparison Operators

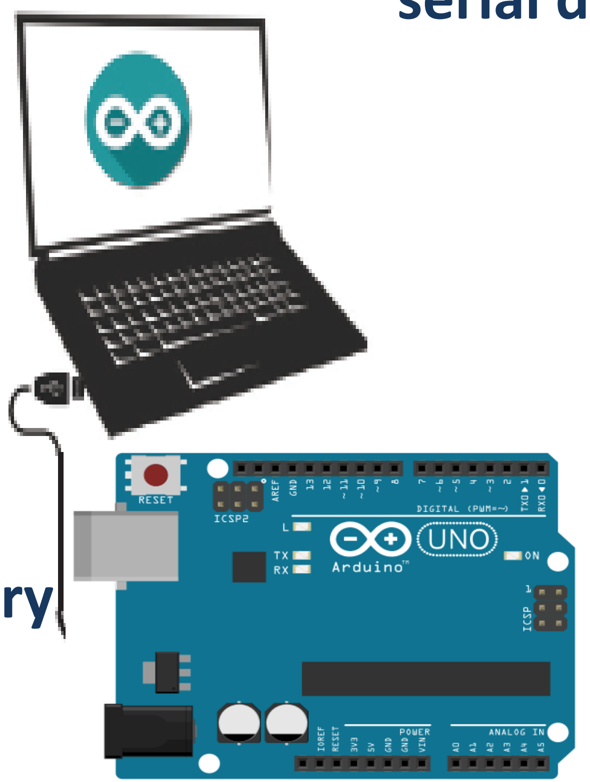
RELATIONSHIP	OPERATOR
Equal to	==
Not equal to	!=
Less than	<
Greater than	>
Less than or Equal to	<=
More than or Equal to	>=

Serial Communication

serial data transmission – 1 bit at a time



USB port

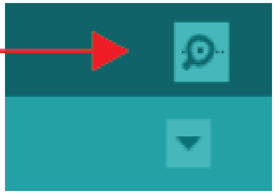


built in library (Serial)

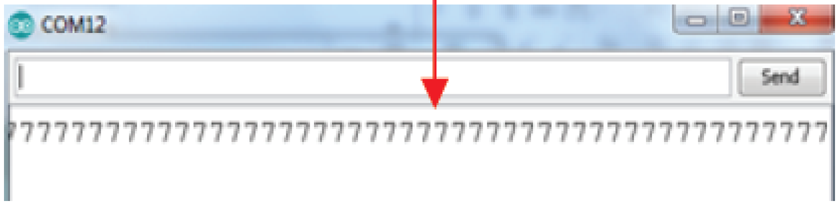
Informations passes between the computer and Arduino through the USB cable were transmitted as zeros ('0') and ones ('1')... also known as bits!

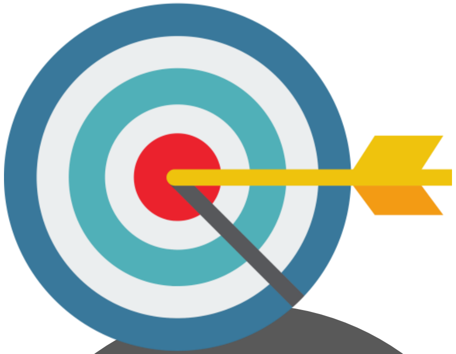
```
int myduino = 7;

void setup()
{
   Serial.begin(9600);
}
void loop()
{
  //Send the value of myduino to Serial port.
  Serial.print(myduino);
}
```

Once uploaded, look at the right hand corner of the IDE and click on the Serial Monitor button.



We will now see the value 7 streaming horizontally across the Serial Monitor.





Try It Out

Using the push button sketch, print "Button Pressed" whenever the button is press

When the button is not press, print "Release"

Print the status of button directly on Serial Monitor

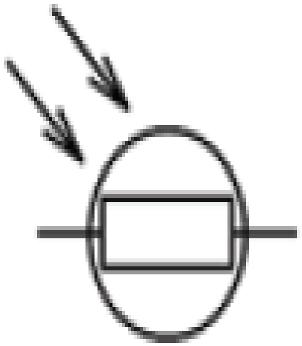
Objective

- To incorporate analogRead() in Arduino Programming

3. ANALOG INPUT AND DIGITAL OUTPUT

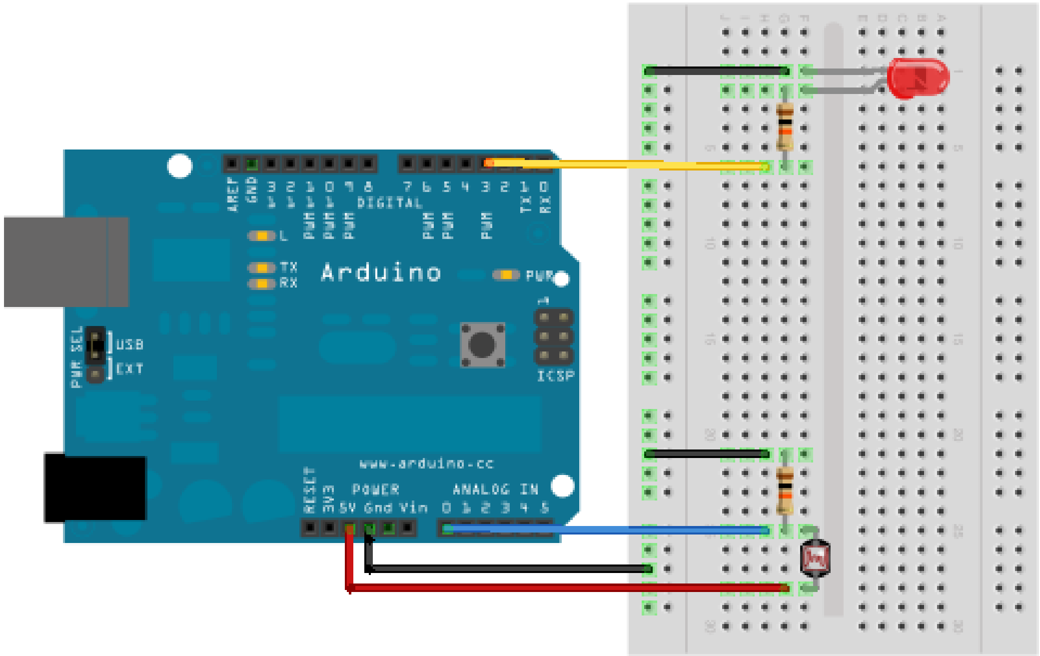
LIGHT SENSOR

A Light Dependent Resistor (LDR) or also known as Photo-resistor is a cheap and easy to use light sensor that changes its resistance with light intensity. Reducing the amount of light it received will increase its resistance, thus making it inversely proportional.



Light incident increases	=	Voltage output increases
Light incident decreases	=	Voltage output decreases

Circuit Diagram



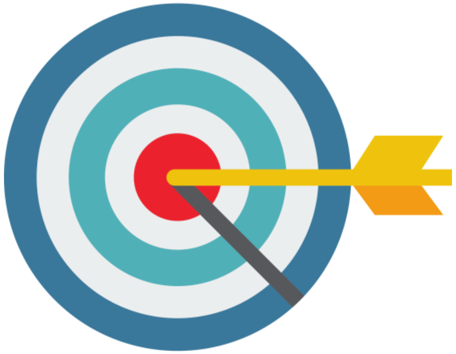
PIN3 to resistor 150 Ohm
Resistor to +ve LED
-ve LED to GND

Light Detected Resistor (LDR)

5V to LDRPIN1
PINA0 to LDRPIN2
LDRPIN2 to Resistor 1kΩ
Resistor 1kΩ to GND

Resistor 1kΩ (brown,black,orange)

1. Construct the Circuit



Try It Out

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

void loop()
{
  int dout = analogRead(A0);
  Serial.print(dout);
  delay(300);
}
```

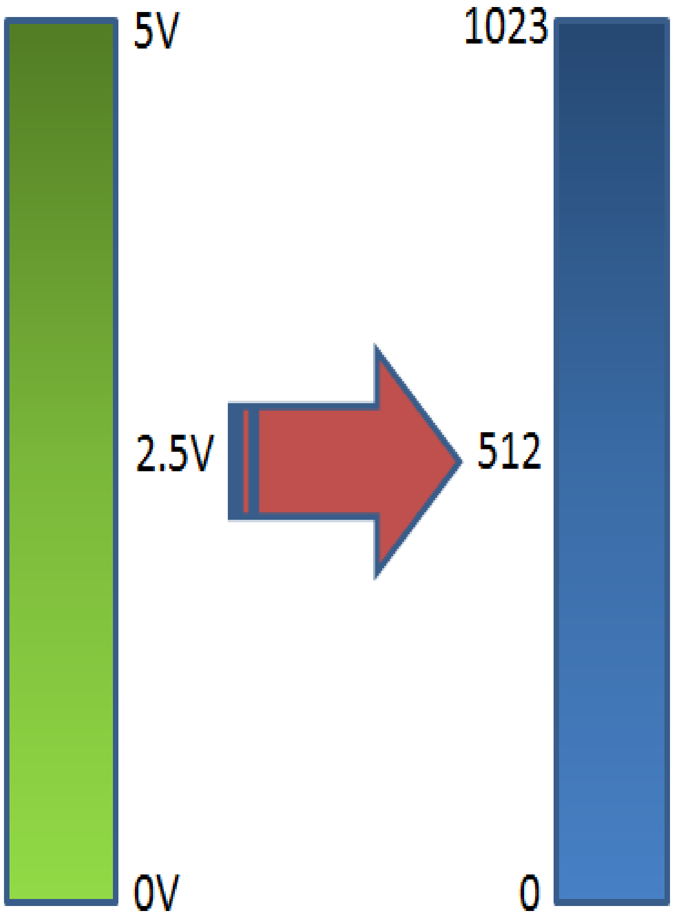
SCENARIO	OBSERVATION (Vout)
LDR exposed to room ambient light	
Closed the top surface of the LDR with your hand.	
Shine a torch light on the LDR.	
Close the light of the room.	

SCENARIO	OBSERVATION
LDR exposed to room ambient light	
Closed the top surface of the LDR with your hand.	
Shine a torch light on the LDR.	
Close the light of the room.	

Analogue Input

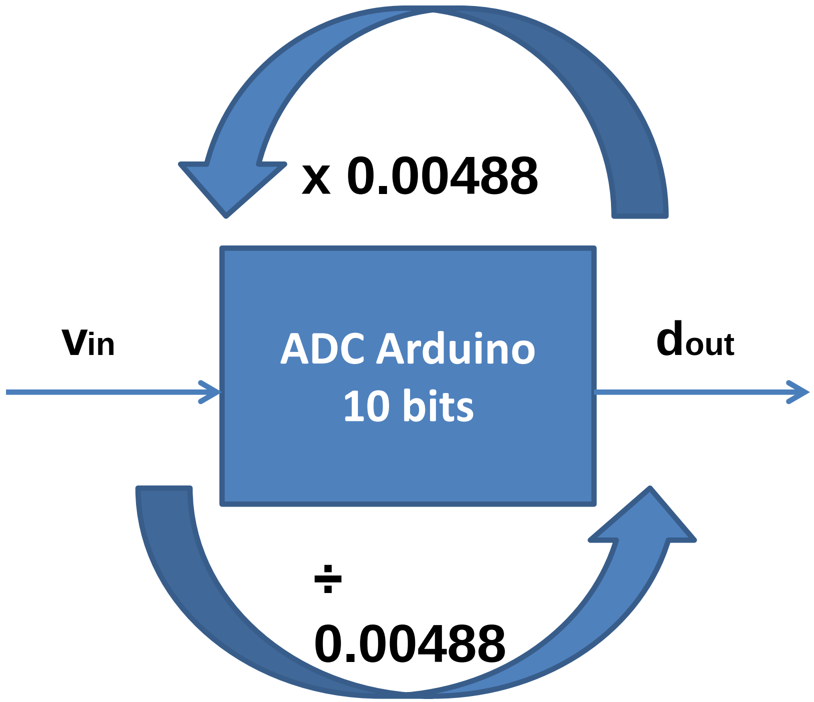
- This is the difference between an on/off sensor (which tells us whether something is there) and an analogue sensor, whose value continuously changes.

- By using the analogRead() function, we can read the voltage applied to one of the pins. This function returns a number between 0 and 1023, which represents voltages between 0 and 5 volts

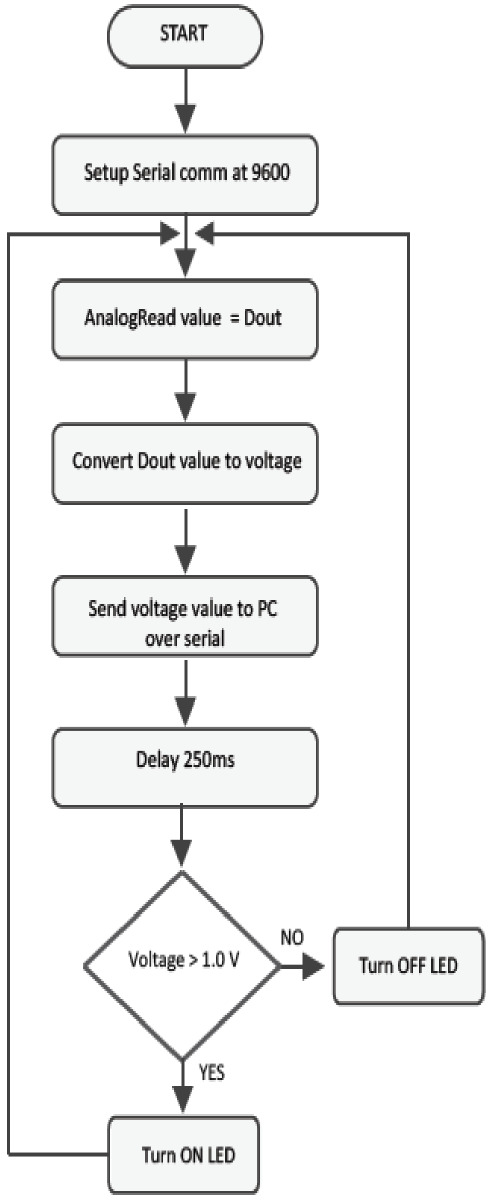


Data Conversion

$$\text{ADC Output } (d_{out}) = \frac{V_{in}}{\text{Resolution}} = \frac{3.3}{0.00488} = 676$$



V_{in} = d_{out} x Resolution = 676 x 0.00488 = 3.3


START
Setup Serial comm at 9600
AnalogRead value = Dout
Convert Dout value to voltage
Send voltage value to PC over serial
Delay 250ms
Voltage > 1.0 V
NO
Turn OFF LED
YES
Turn ON LED

Example: Complete System (Smart Street Light system)

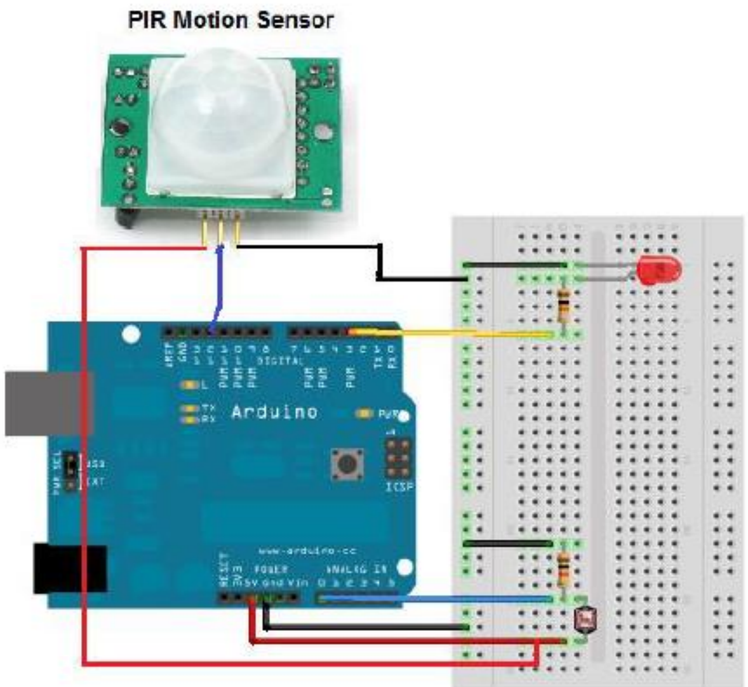
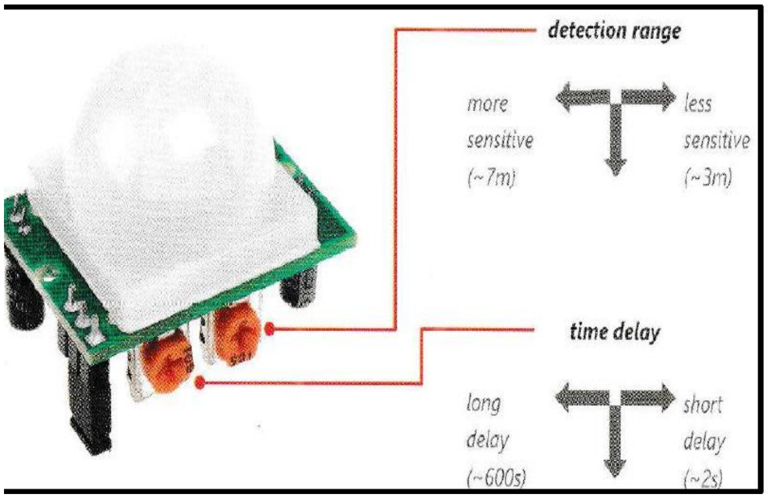


Figure 13 : Complete circuit



Example: Complete System (Smart Street Light system)

<u>Cicuit and coding</u> : Use previous circuit

Code	(Application: Smart Street Light system)

`int analogvalue;`
`const int ldrPin=A0;`
`const int ledPin=3;`
`const int PIR = 12;`
`int currentstate=0;`
	• declare analogvalue as variable
	• declare ldrpin as constant integer port A0
	• declare ledpin = port digital 3
	• PIR sensor pin 12

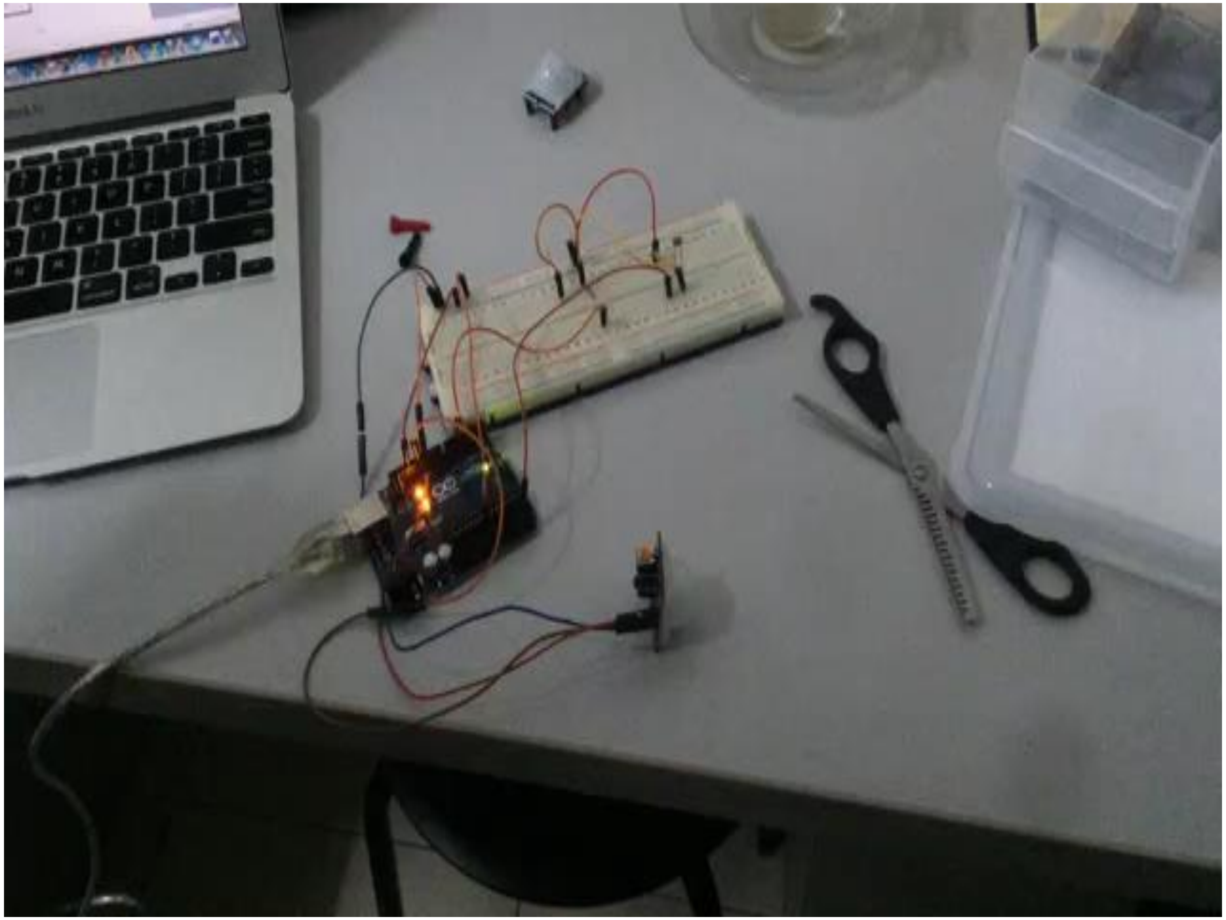
`void setup()`
`{`
`pinMode(PIR,INPUT);`
`Serial.begin(9600);`
`pinMode (ledPin,OUTPUT);`
`}`
	• the speed(baud rate) at which data is transferred to serial monitor

`void loop()`
`{`
`currentstate=digitalRead (PIR);`
`analogvalue=analogRead (ldrPin);`
`Serial.println (analogvalue);`
	• Read analog port A0 and assign value to analogvalue. Value ranges from 0-1023 corresponding value 0-5V
	• display value to serial monitor

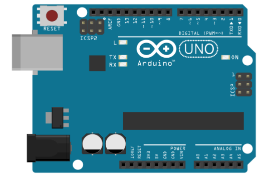
```
if (analogvalue<120)
{
   if (currentstate==HIGH)
   {
       analogWrite(ledPin,255);
   }
  else
       analogWrite(ledPin,5);
}
 else
{
  digitalWrite (ledPin,LOW);
}}
```

- if analog value <120 (day noon) ,LED will OFF
- if analog value >120 (night) && PIR detect, LED will ON and give the maximum brightness
- if analog value >120 (night) && PIR not detect(no Car available), LED will ON and give the half

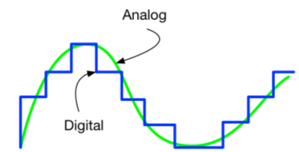
Result:



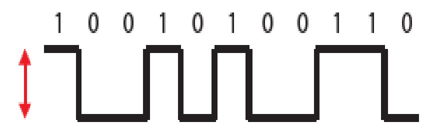
Let's Recap Today's Session



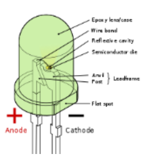
1. Arduino Microcontroller and its programming language



2. Electronic Components and its properties



3. PWM, Serial Monitor, ADC Conversion



Digital Output

Exp 1
Blinking LED (out)

Digital Input

Exp 3
Push Button (in)
LED (out)

Analog Output

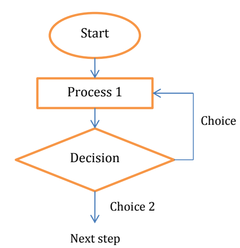
Exp 2
Dimmer LED (out)

Analog Input

Exp 4
LDR (in)
LED (out)

Exp 6
IR Sensor (in)
Motor (out)
Autonomous

Exp 5
Bluetooth to control Motor (out)



4. Programming Concepts – conditional, for loops and sequential

Arduino & You

padlet

Please log into my padlet again ☺:-

- What could an Arduino do?
- What would you do with an Arduino?

https://padlet.com/nurulhazlina/ArduinoWeek4

Next..

CO 3

mBlock

- week 2
- Computational Thinking

CO 3

Design Thinking

- week 3
- STEM outreach

CO 2

Arduino

- week 4
- Input outputs

CO 2

Coaching Technique

- week 5

CO 1

School Engagement

- week 6 - 13

School Engagement Schedule

28 March 3 pm mBlock	29 March 3 pm Arduino	11 April 3 pm App Inv	12 April 3 pm App Inv

3 April 2.30 pm mBlock	10 April 2.30 pm Arduino	17 April 2.30 pm App Inv

Time	7 May 3 pm mBlock	8 May 3 pm Arduino	9 May 3 pm App Inv	10 May 3 pm App Inv
AM				
PM				