ARDUINO LAB

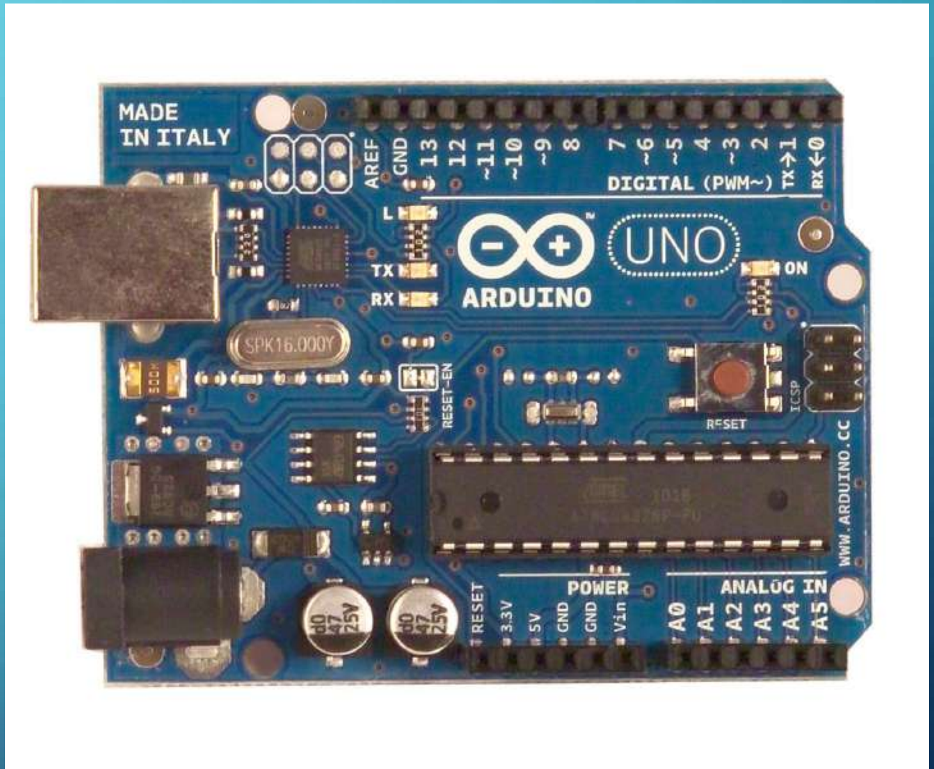
TEACHER: ENGR. MUHAMMAD SAQIB

OUTLINE:

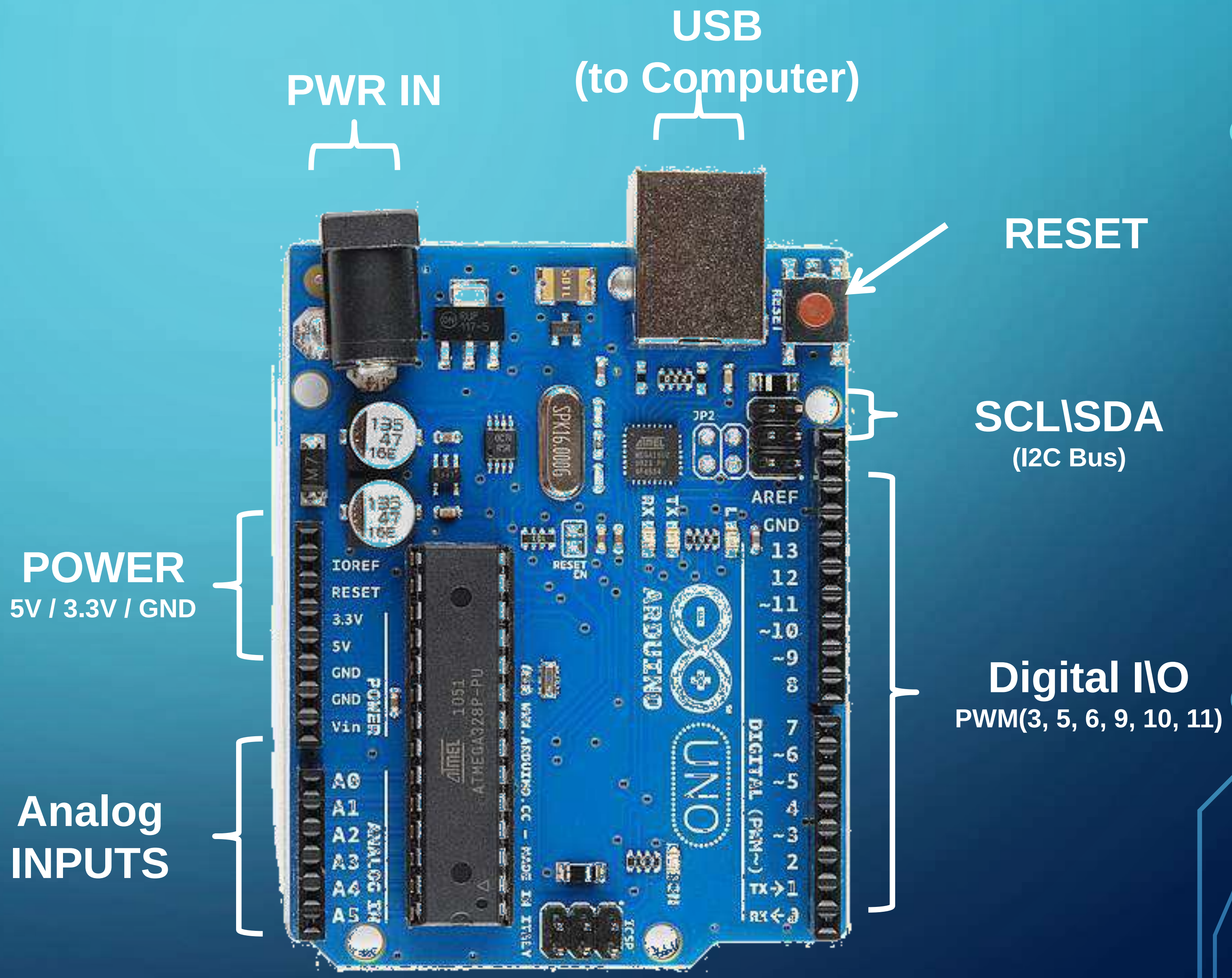
- **Introduction to Arduino**
- **Arduino Lab Practicals**

WHAT IS AN ARDUINO ?

- Open Source electronic prototyping platform based on flexible easy to use hardware and software.



PIN CONFIGURATION



PWR IN

USB
(to Computer)

RESET

SCL\SDA
(I2C Bus)

POWER
5V / 3.3V / GND

Digital I\O
PWM(3, 5, 6, 9, 10, 11)

Analog
INPUTS

BARE MINIMUM CODE

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


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

BARE MINIMUM CODE

- setup : It is called only when the Arduino is powered on or reset. It is used to initialize variables and pin modes

- loop : The loop functions runs continuously till the device is powered off. The main logic of the code goes here. Similar to while (1) for micro-controller programming.

PINMODE

- A pin on arduino can be set as input or output by using pinMode function.
- pinMode(13, OUTPUT); // sets pin 13 as output pin
- pinMode(11, INPUT); // sets pin 11 as input pin

CONCEPT: INPUT VS. OUTPUT

Referenced from the perspective of Arduino.

Inputs is a signal / information going into the board.

Output is any signal exiting the board.

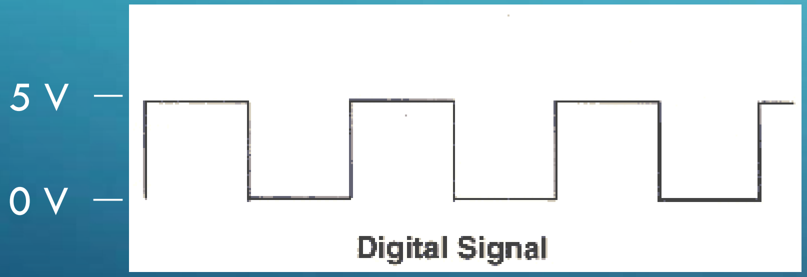
Examples: Switches, Light Sensors, Humidity Sensors, Temperature Sensors.

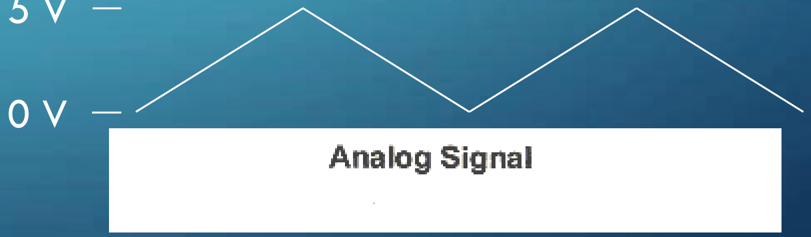
Examples: LEDs, DC motor, servo motor, a piezo buzzer, relay, an RGB LED.

CONCEPT: ANALOG VS. DIGITAL

- Microcontrollers are **digital** devices – ON or OFF. Also called discrete.

- **Analog** signals are anything that can be a full range of values.



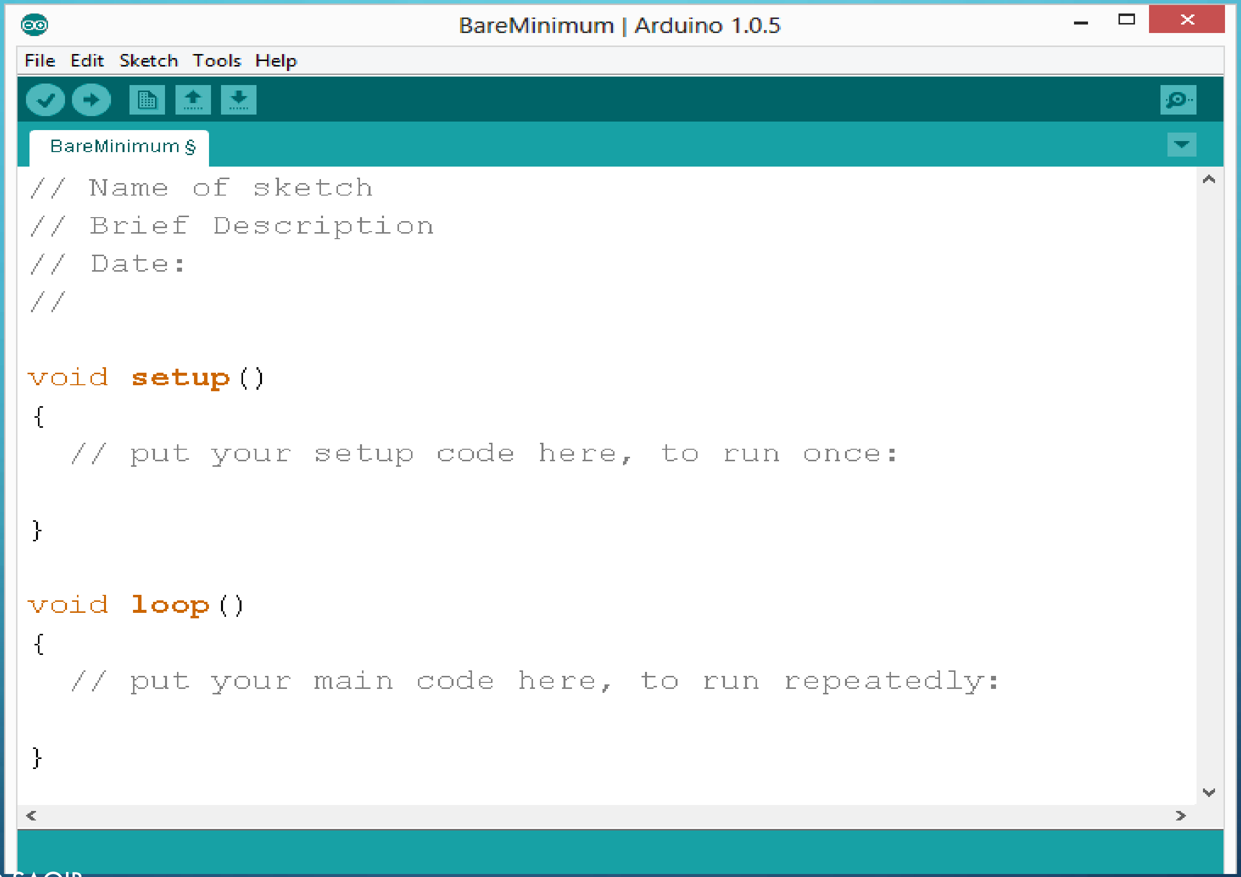


CONCEPT: COMMENTS

- Comments are for you – the programmer and your friends…or anyone else human that might read your code.

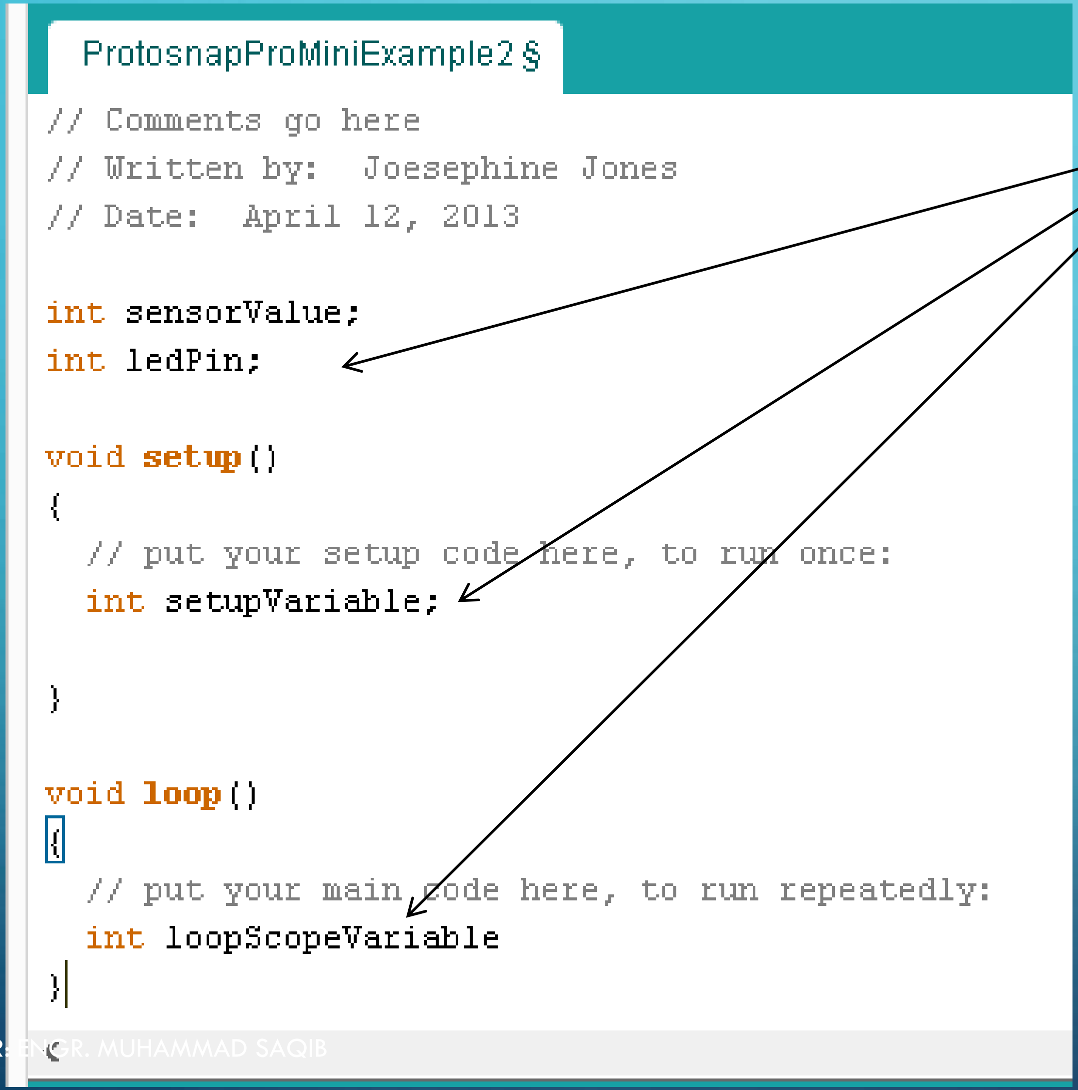
```
// this is for single line comments
// it’s good to put a description at the top and before anything ‘tricky’

/* this is for multi-line comments
   Like this…
  And this….
*/
```

CONCEPT: COMMENTS



```
// Name of sketch
// Brief Description
// Date:
//

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

}

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

}
```

CONCEPTS: VARIABLES



```
// Comments go here
// Written by:  Joesephine Jones
// Date:  April 12, 2013

int sensorValue;
int ledPin;

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

}

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

Variable Scope

Global

Function-level

CONCEPTS: VARIABLES

- **Variable Types:**

8 bits	16 bits	32 bits
byte	int	long
char	unsigned int	unsigned long
		float

READING/WRITING DIGITAL VALUES

- digitalWrite(13, LOW); // Makes the output voltage on pin 13 , 0V

- digitalWrite(13, HIGH); // Makes the output voltage on pin 13 , 5V

- int buttonState = digitalRead(2); // reads the value of pin 2 in buttonState

ANALOG TO DIGITAL CONVERSION

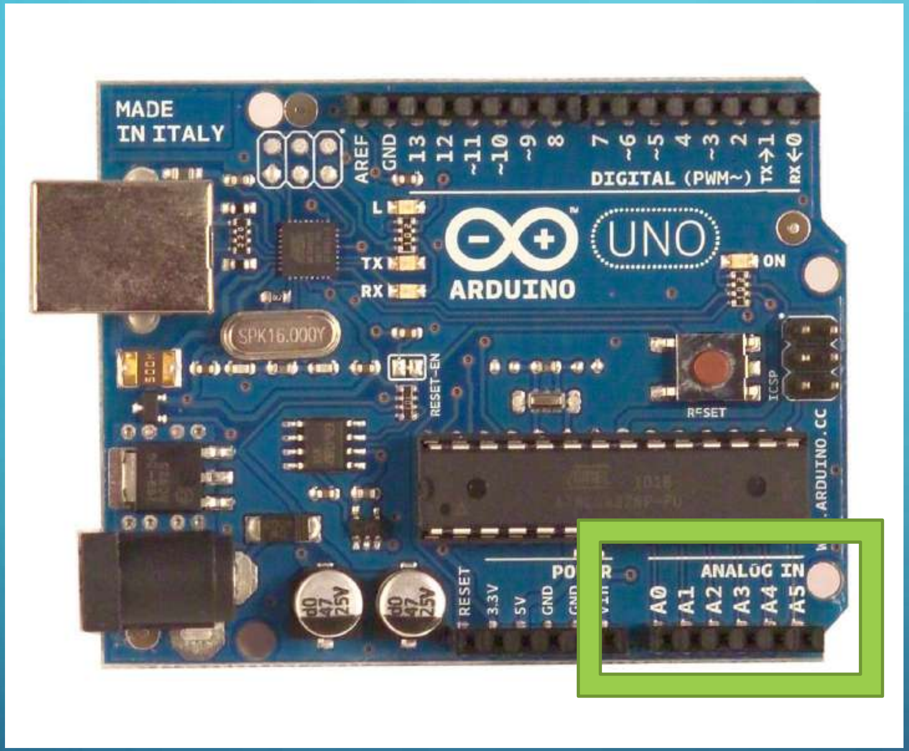
- What is analog ?

  It is continuous range of voltage values (not just 0 or 5V)

- Why convert to digital ?

  Because our microcontroller only understands digital.

ADC IN ARDUINO UNO



ADC IN ARDUINO UNO

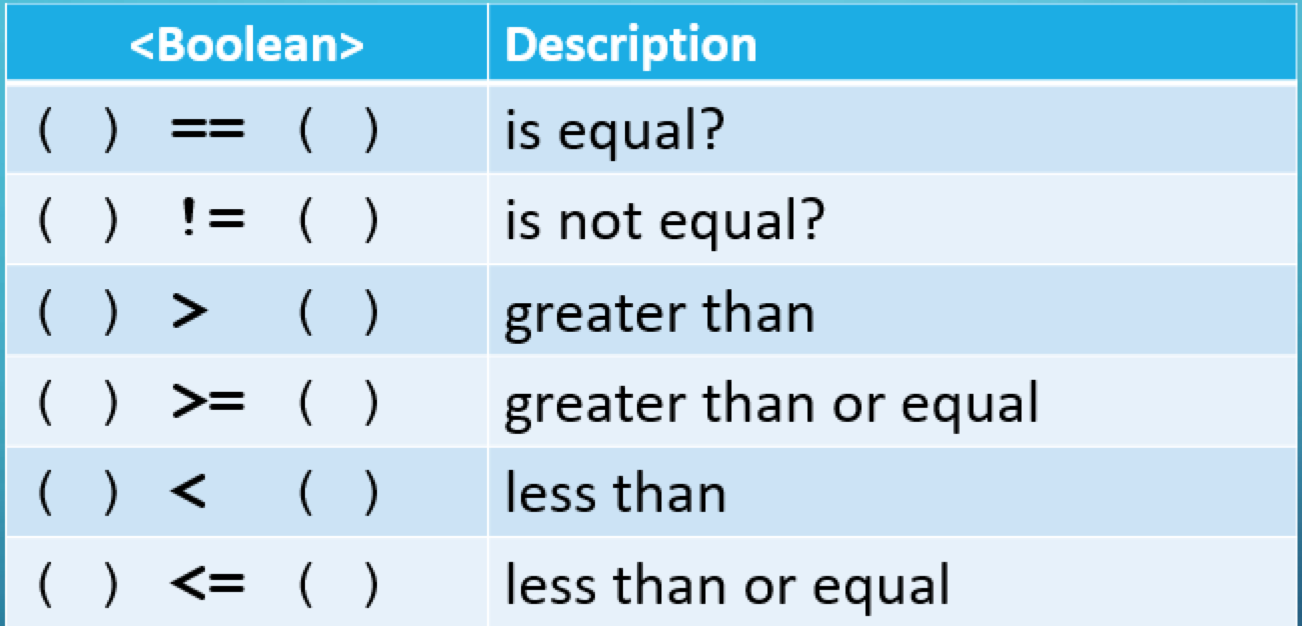
- The Arduino Uno board contains 6 pins for ADC (A0 to A5).
- 10-bit analog to digital converter.
- This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023

READING/WRITING ANALOG VALUES

- Int a= analogRead(A0); // used to read the analog value from the pin A0

- analogWrite(2,128);

BOOLEAN OPERATORS

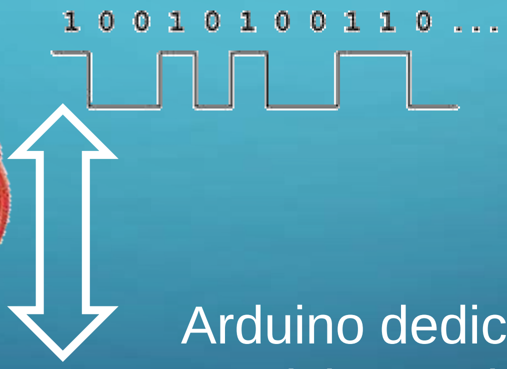


<Boolean>	Description
( ) == ( )	is equal?
( ) != ( )	is not equal?
( ) > ( )	greater than
( ) >= ( )	greater than or equal
( ) < ( )	less than
( ) <= ( )	less than or equal

CONCEPT: SERIAL COMMUNICATION

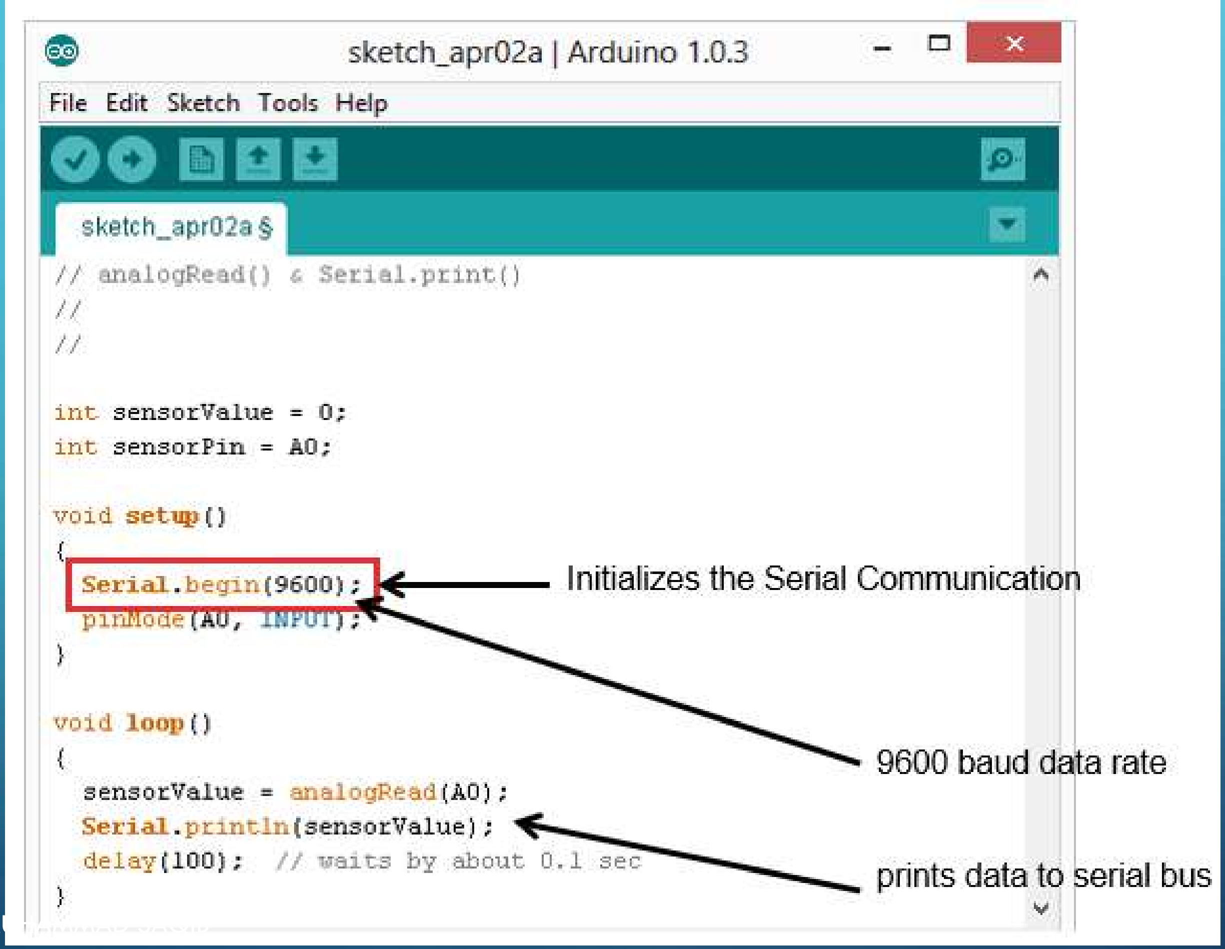
Method used to transfer data between two devices.

Data passes between the computer and Arduino through the USB cable. Data is transmitted as zeros (‘0’) and ones (‘1’) sequentially.



Arduino dedicates Digital I/O pin # 0 to receiving and Digital I/O pin #1 to transmit.

CONCEPT: SERIAL COMMUNICATION



```
// analogRead() & Serial.print()
//
//

int sensorValue = 0;
int sensorPin = A0;

void setup()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
}

void loop()
{
  sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(100);  // waits by about 0.1 sec
}
```

Initializes the Serial Communication

9600 baud data rate

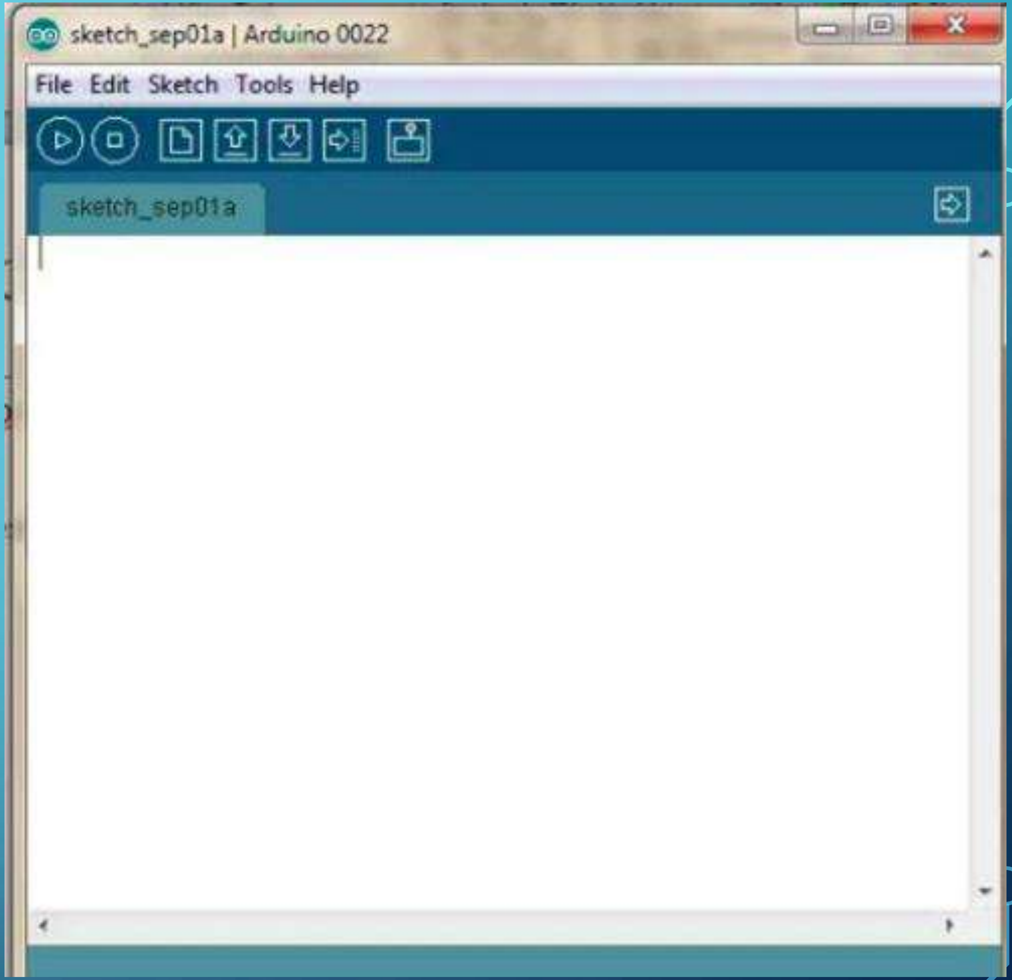
prints data to serial bus

ADUINO IDE INSTALLATION

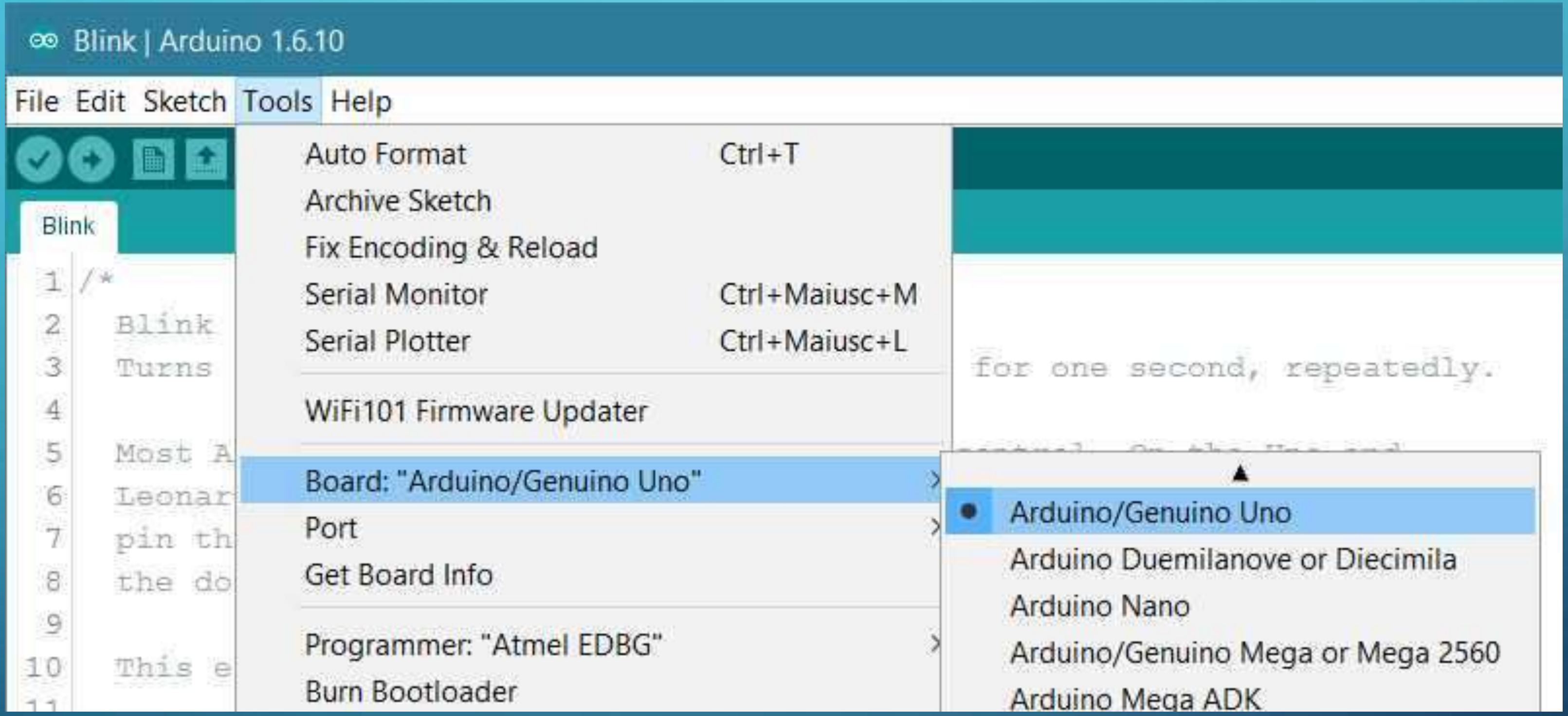
- https://www.youtube.com/watch?v=TbHsOgtCMDc&ab_channel=ProgrammingKnowledge2

ARDUINO IDE

- Arduino is an open source IDE developed from wiring.
- Arduino is programmed in C language.
- It provides the serial monitor to see the serial from the USB and virtual COM port.
- Compiling, verification and burning can be done through a single click.



SELECTING ARDUINO BOARD



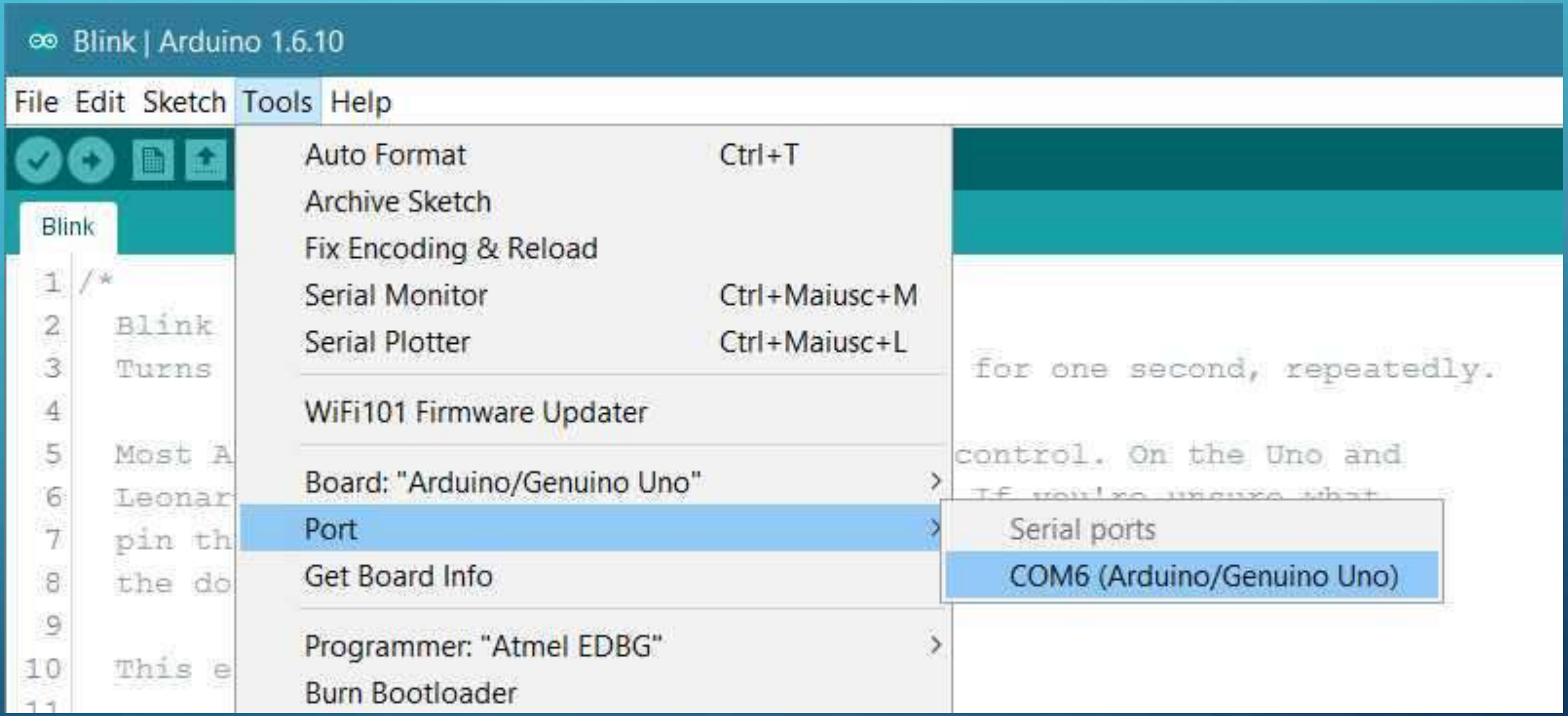
Board: "Arduino/Genuino Uno"

● Arduino/Genuino Uno

Programmer: "Atmel EDBG"

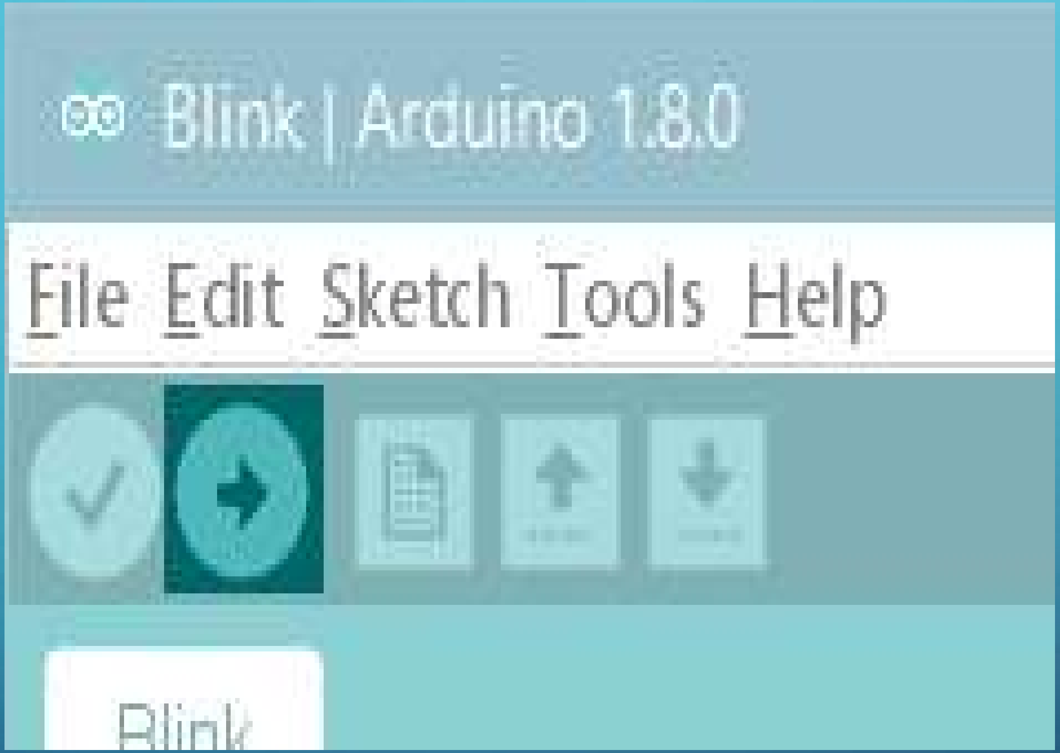
```
/*
  Blink
  Turns                                for one second, repeatedly.

  Most A
  Leonar
  pin th
  the do

  This e
```

SELECTING PORT



```
/*
  Blink
  Turns                                  for one second, repeatedly.

  Most A                              control. On the Uno and
  Leonar
  pin th
  the do

  This e
```

UPLOAD PROGRAM



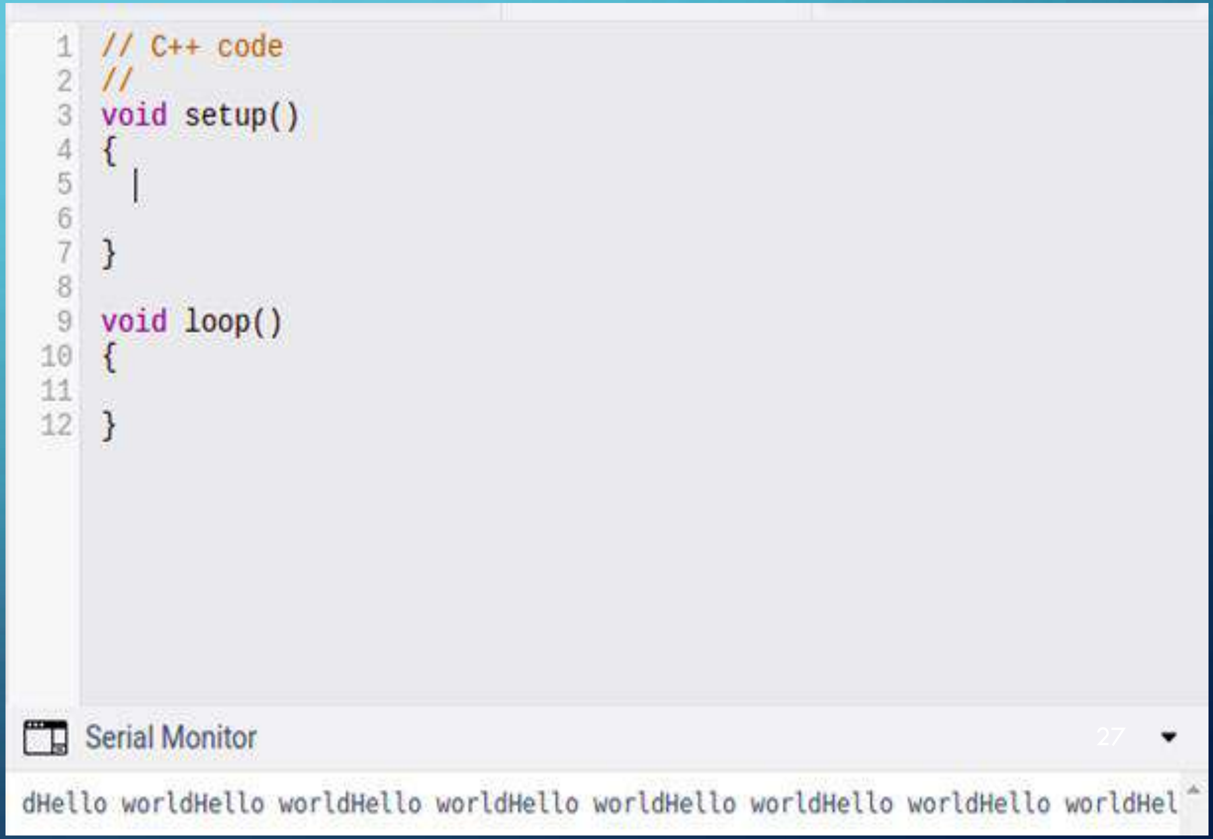
ARDUINO PROGRAM

The basic Arduino program structure consists of two function

- **Setup** is use to program setup and variable initialization.

  Code in this section is executed once.

- **Loop** contain code for the working of program and execute this code in loop.



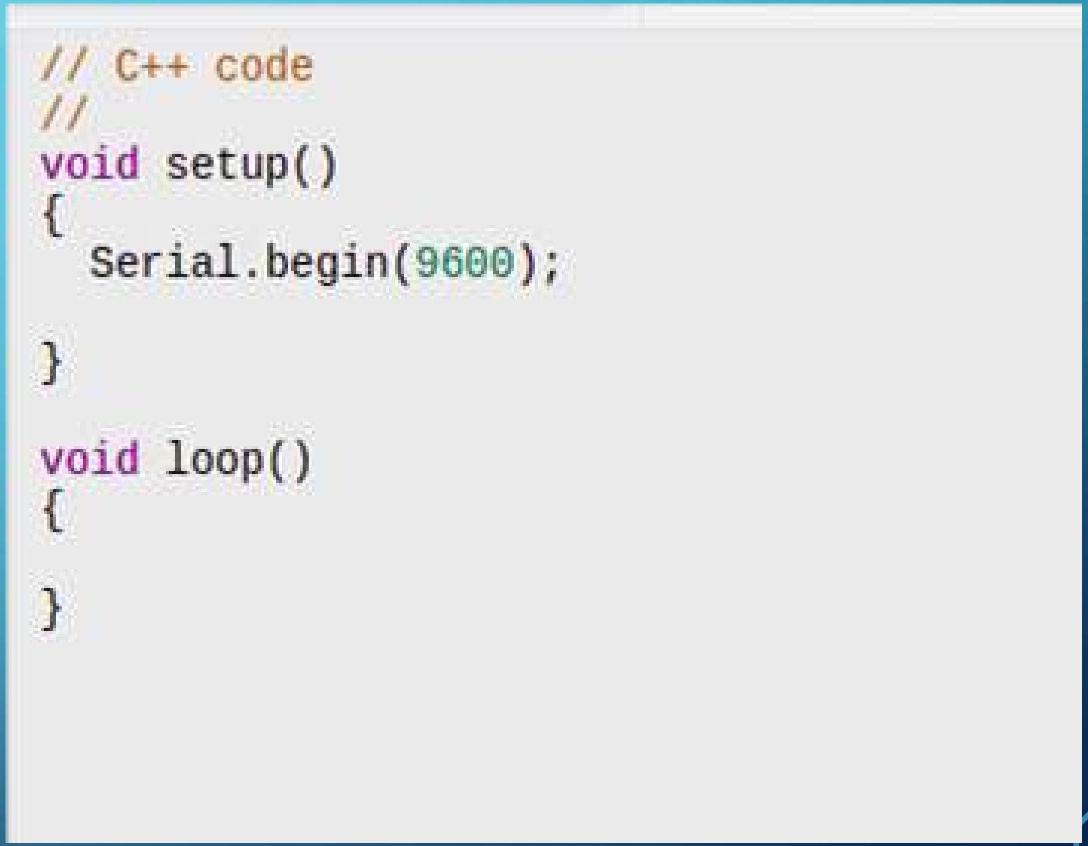
```
// C++ code
//
void setup()
{
  |

}

void loop()
{

}
```

```
dHello worldHello worldHello worldHello worldHello worldHello worldHello worldHel
```

BAUD RATE

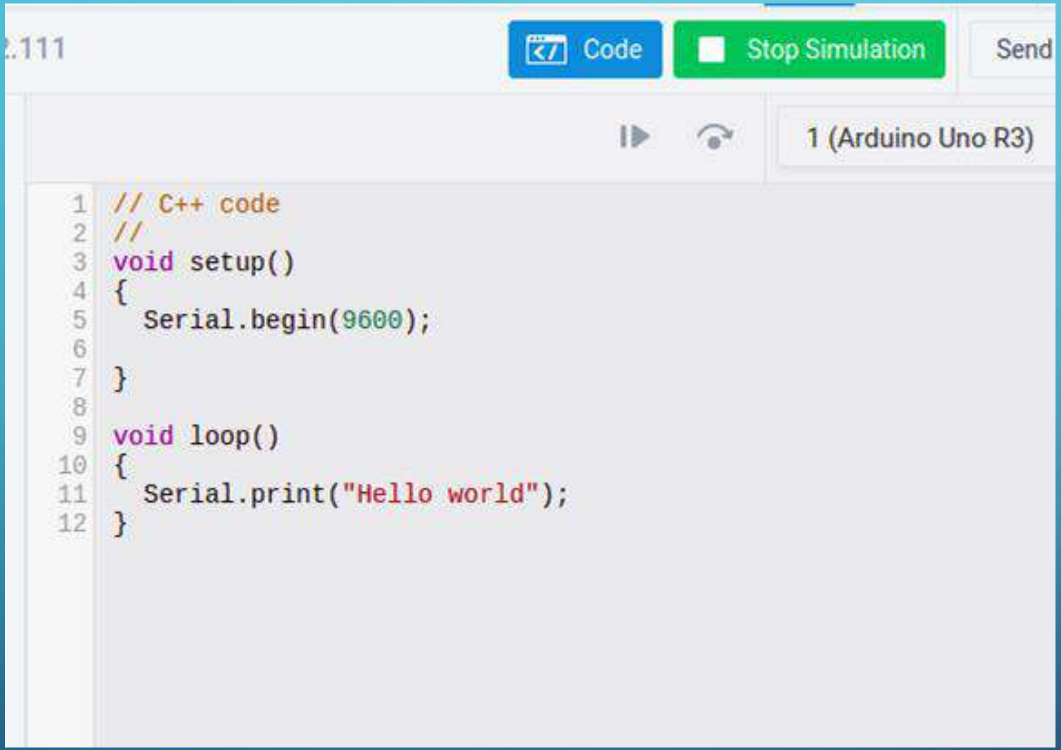
The baud rate is the rate at which information is transferred in a communication channel.

"9600 baud" means that the serial port is capable of transferring a maximum of 9600 bits per second.



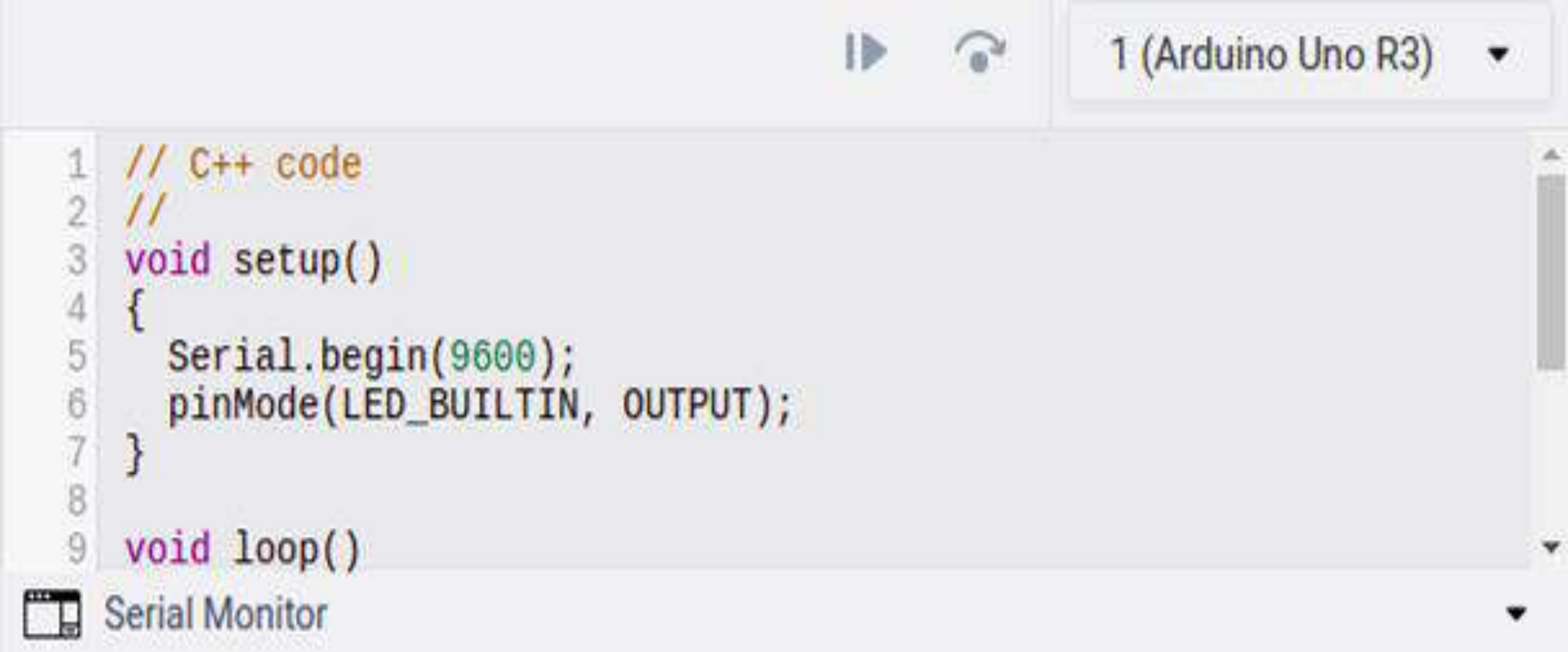
```
// C++ code
//
void setup()
{
  Serial.begin(9600);

}

void loop()
{

}
```

HELLO WORLD



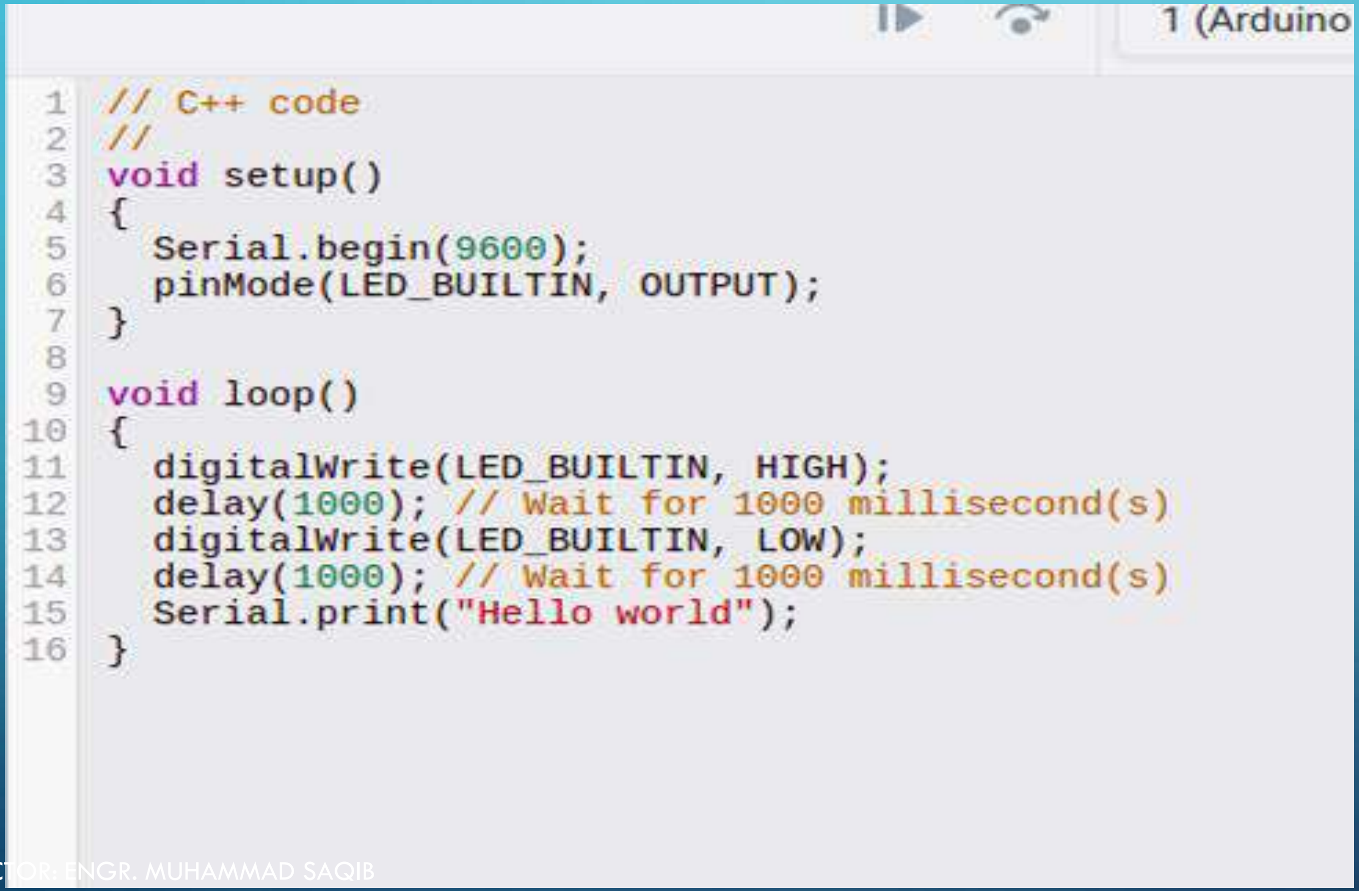
```
// C++ code
//
void setup()
{
  Serial.begin(9600);

}

void loop()
{
  Serial.print("Hello world");
}
```

INITIALIZING WHICH PIN TO USE AS INPUT OR OUTPUT



```
// C++ code
//
void setup()
{
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
```

LED BLINKING



```
// C++ code
//
void setup()
{
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  Serial.print("Hello world");
}
```

INTEGERS, FLOAT AND VARIABLES

- **Integers -> int**

  0, 1, 2, 3, 4, ……….

- **rational numbers-> float**

  0.1, 0.3, 0.45, ………..

- **Character**

  ‘a’, ‘A’, ‘c’

- **Strings**

  “Hello”, “Tea”, “class” etc.

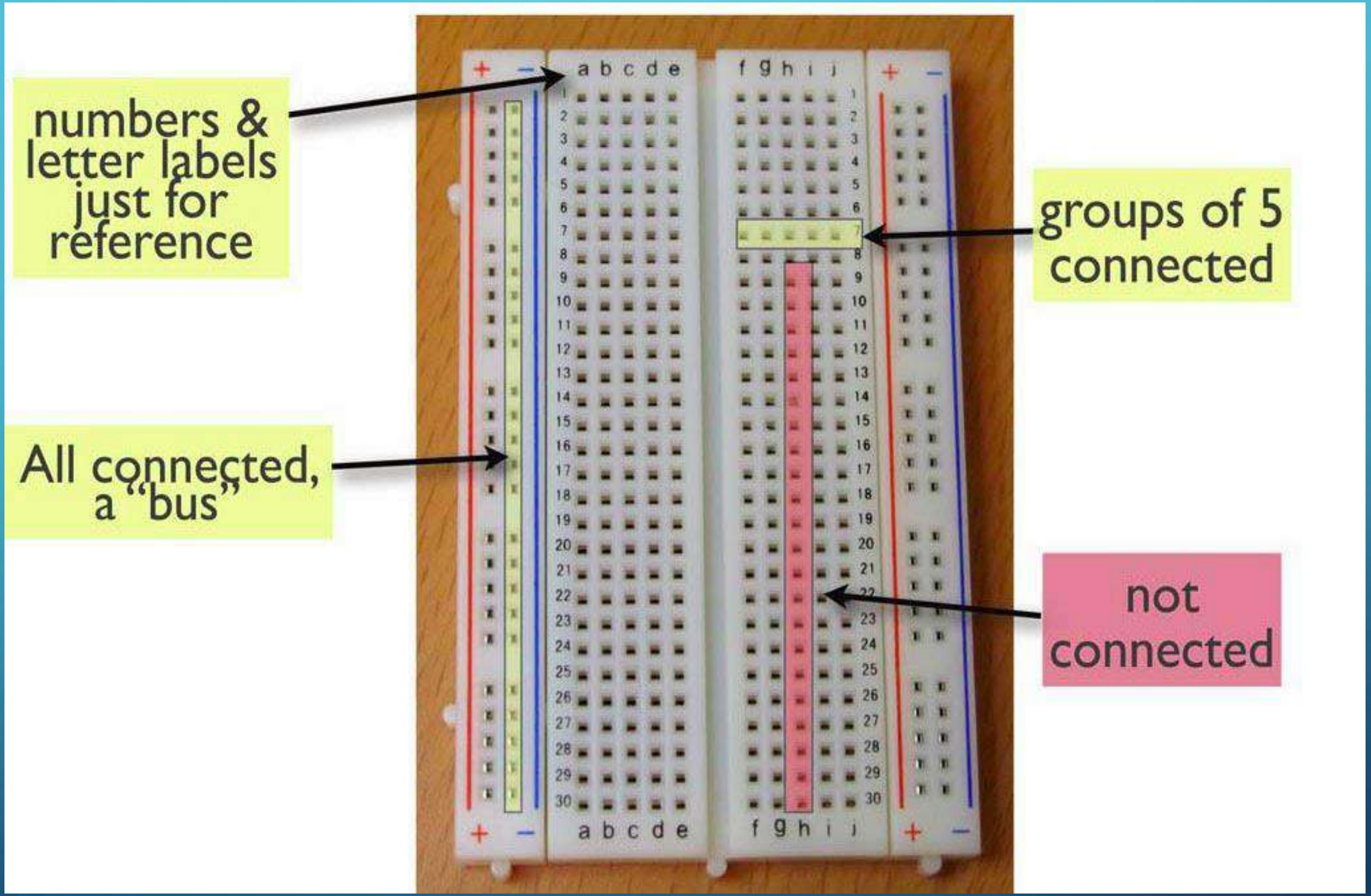
BREADBOARD

A breadboard, or protoboard, is a construction base for prototyping of electronics.

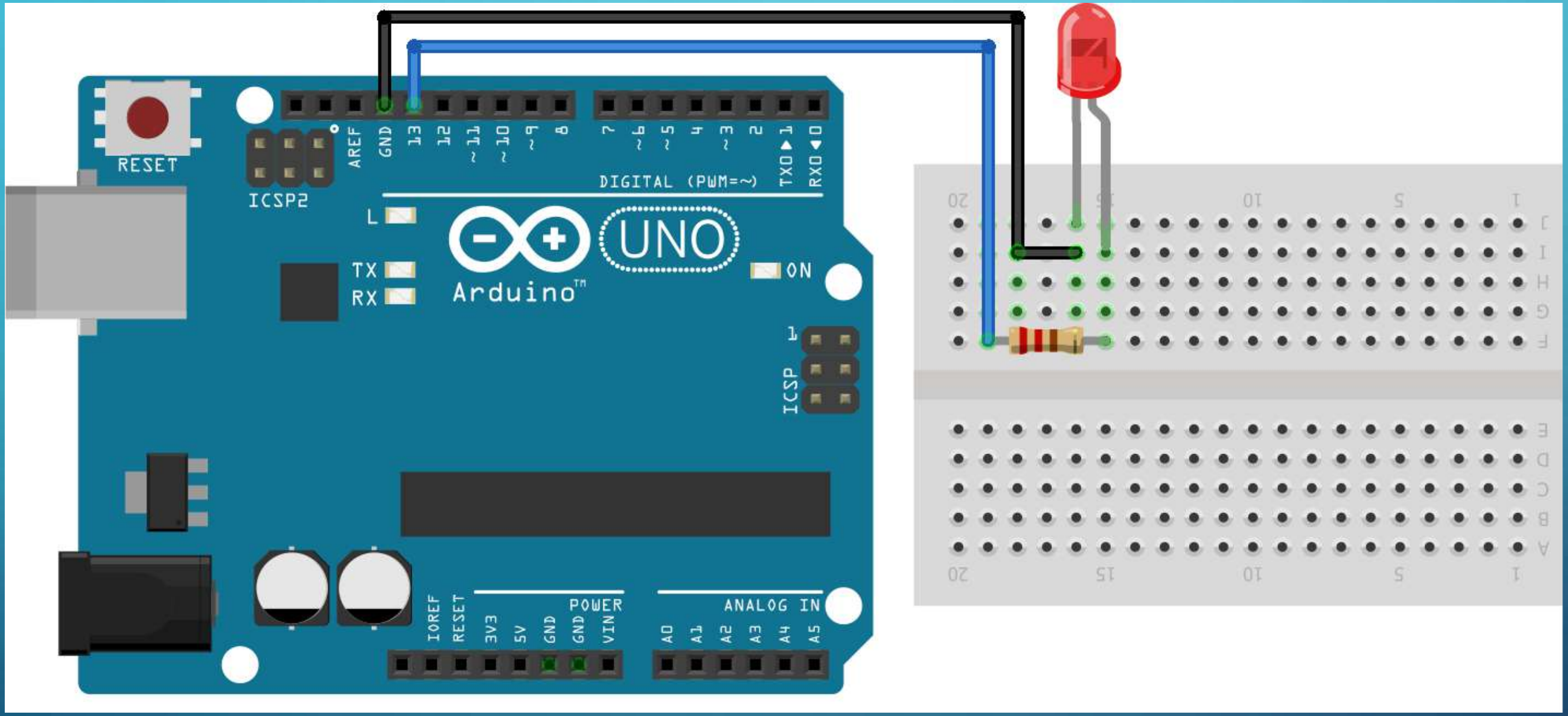
Breadboard does not require soldering, it is reusable.

The holes in a breadboard are connected by metal clips that span five holes, horizontally.

BREADBOARD CONNECTION



LED CIRCUIT


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino™
ON
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
20
15
10
5
1
J
I
H
G
F
E
D
C
B
A

VARIABLES

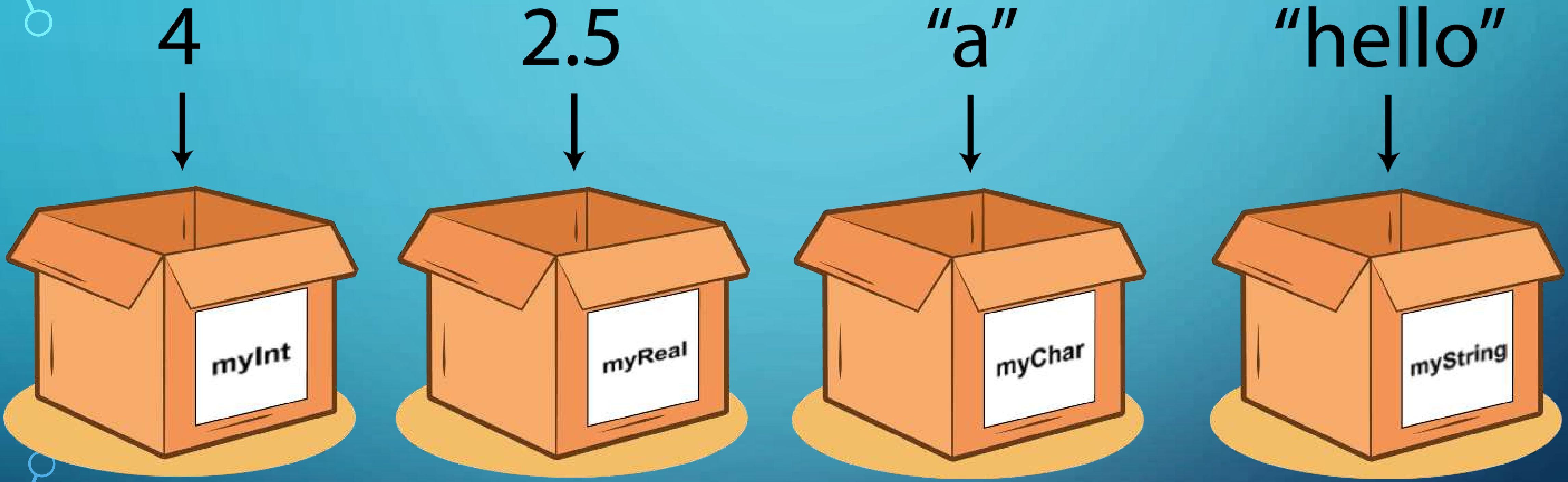
Why use variable?

What if you want to use another pin for LED, you will change the pin number in all the places where it has been used before.

Variables are handy.

We avoid defining Constants.

VARIABLES



int

float

char

String

CONDITIONAL STATEMENTS DECISION MAKING

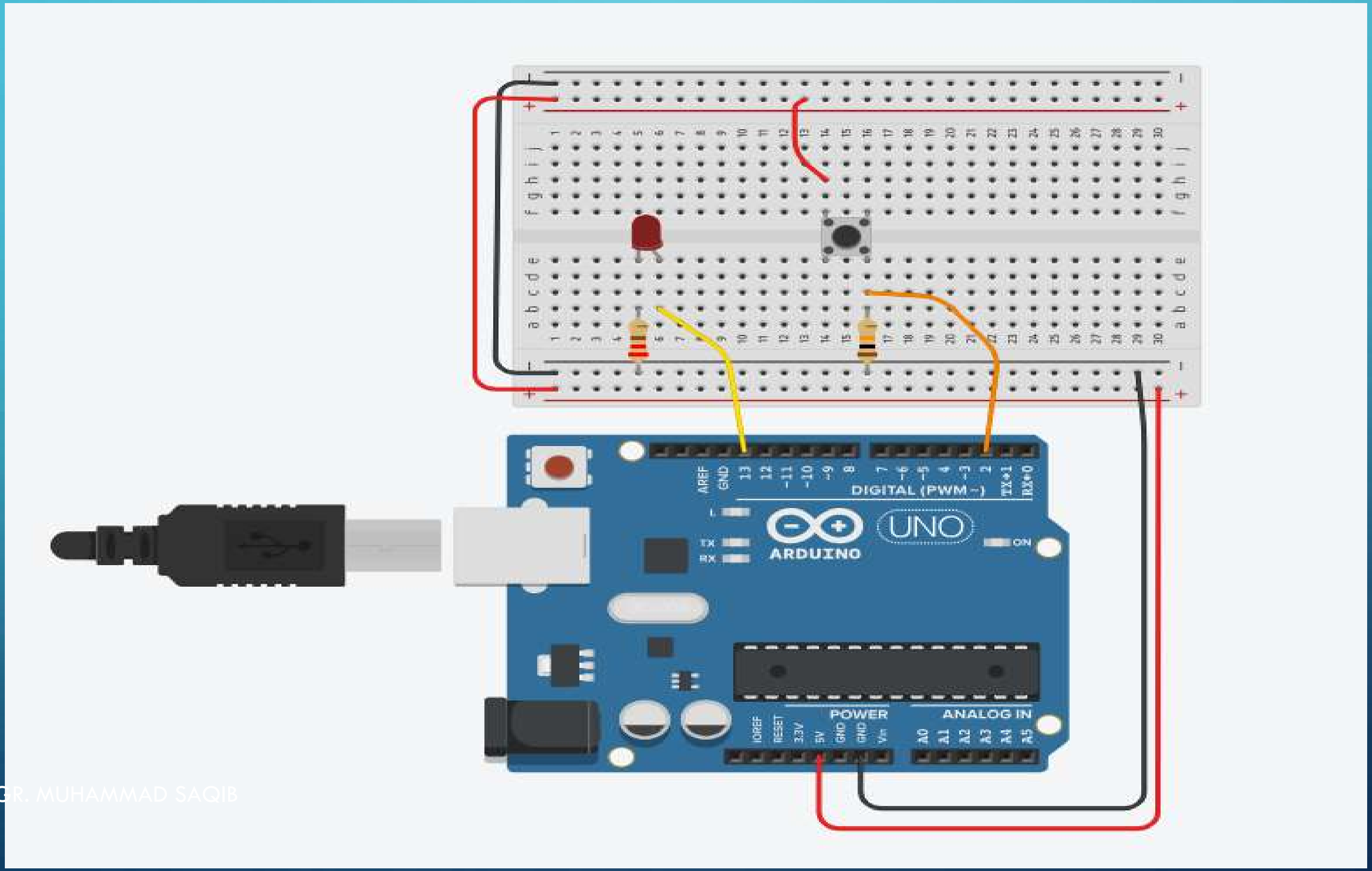
If and else statements

- Decision making
- If the first condition is false, only then the second condition will be checked.

SYNTAX OF CONDITIONAL STATEMENT

```
if (variable==condition){
//Thing you want to do when above condition is satisfied
}

else {
//If above condition is False come to else
}
```

CONTROL LED USING PUSH BUTTON



CODE

- // C++ code
- //
- int buttonState = 0;
- void setup()
- {
-   pinMode(2, INPUT);
-   pinMode(LED_BUILTIN, OUTPUT);
- }
- void loop()
- {
-

- // read the state of the pushbutton
-   buttonState = digitalRead(2);
-   // check if pushbutton is pressed. if it is, the
-   // button state is HIGH
-   if (buttonState == HIGH) {
-     digitalWrite(LED_BUILTIN, HIGH);
-   } else {
-     digitalWrite(LED_BUILTIN, LOW);
-   }
-   delay(10); // Delay a little bit to improve simulation performance
- }

PWM AND DUTY CYCLE

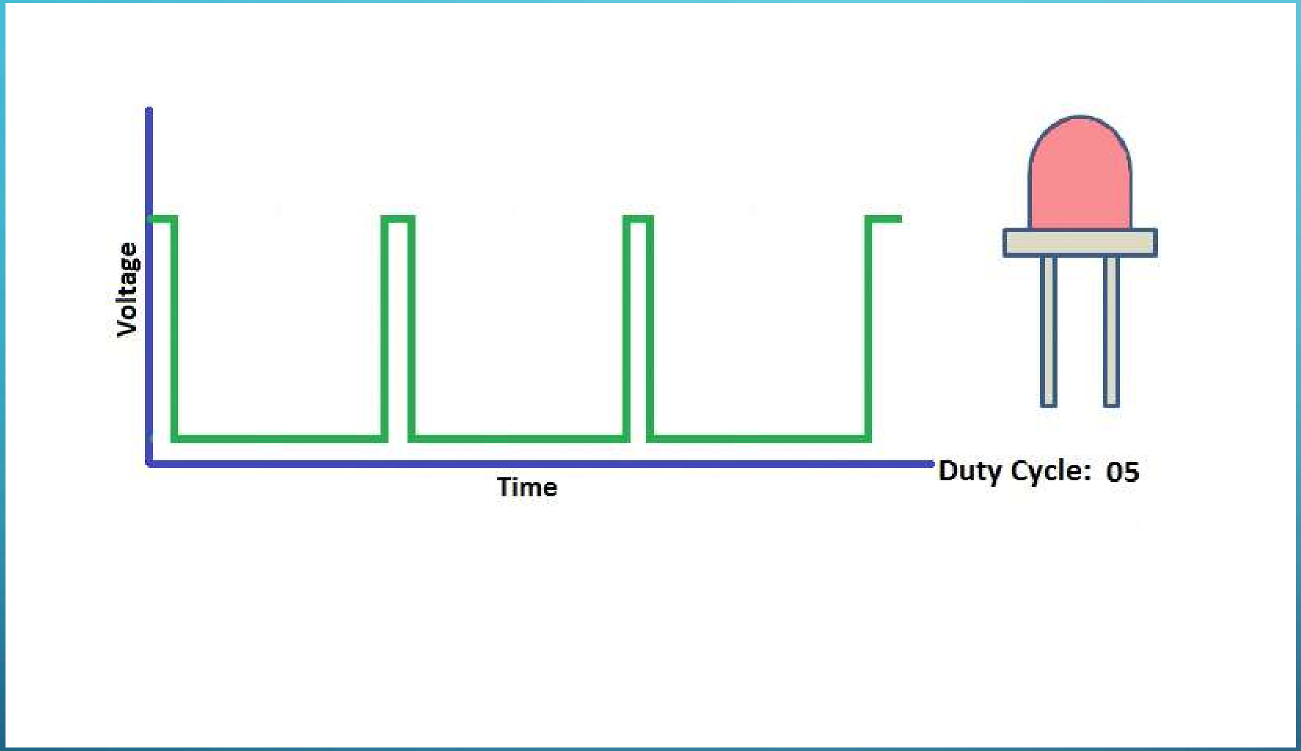
PWM stands for Pulse Width Modulation and it is a technique used in controlling the brightness of LED, speed control of DC motor, controlling a servo motor or where you have to get analog output with digital means.

A duty cycle or power cycle is the fraction of one period in which a signal or system is active.

$$D = \frac{PW}{T}$$

where D is the duty cycle, P W is the pulse width (pulse active time), and T is the total period of the signal.

PWM



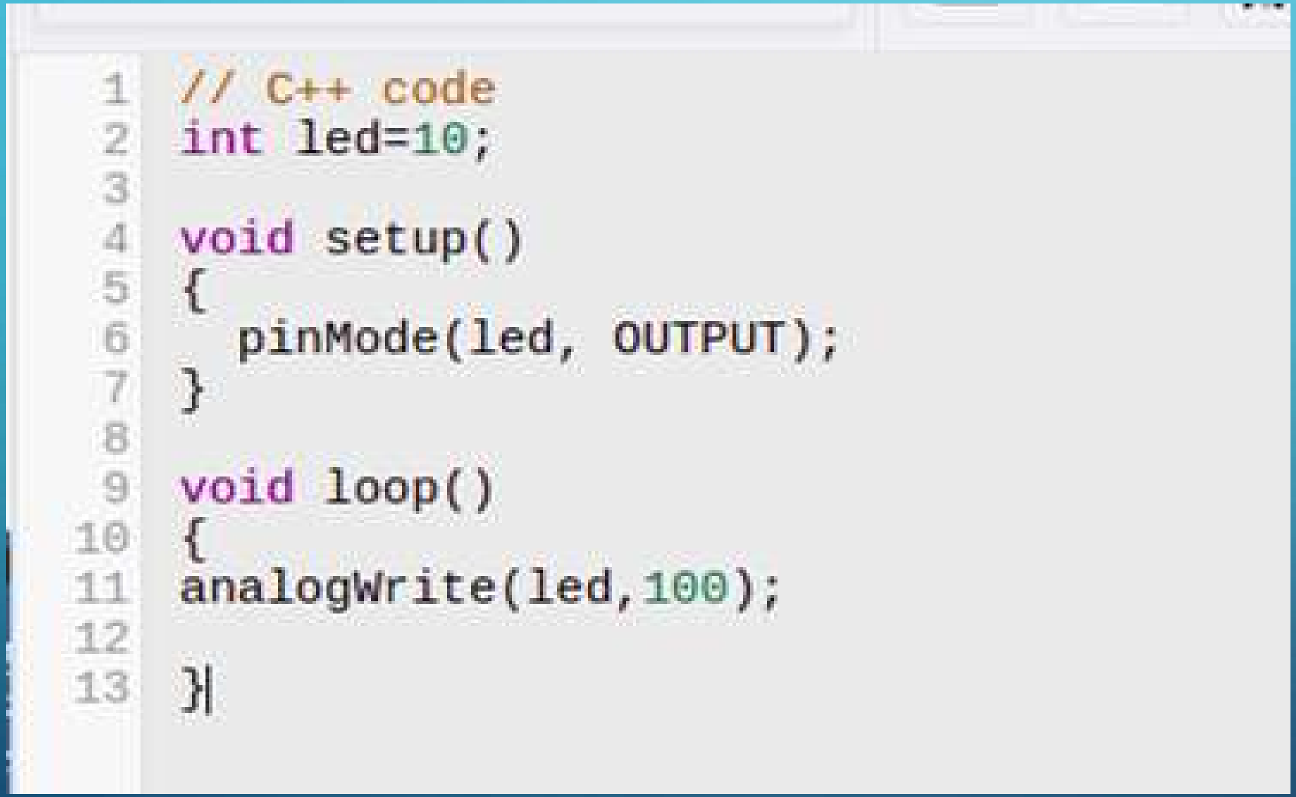
PWM

- Only digital pins with tilde (~) sign is use for giving PWM signal

- Digital pin is use to write Analog output (analogWrite(pin,value))

- value range is 0 to 255

LED BRIGHTNESS CONTROLL USING PWM



```
// C++ code
int led=10;

void setup()
{
  pinMode(led, OUTPUT);
}

void loop()
{
analogWrite(led,100);

}
```

FOR LOOP

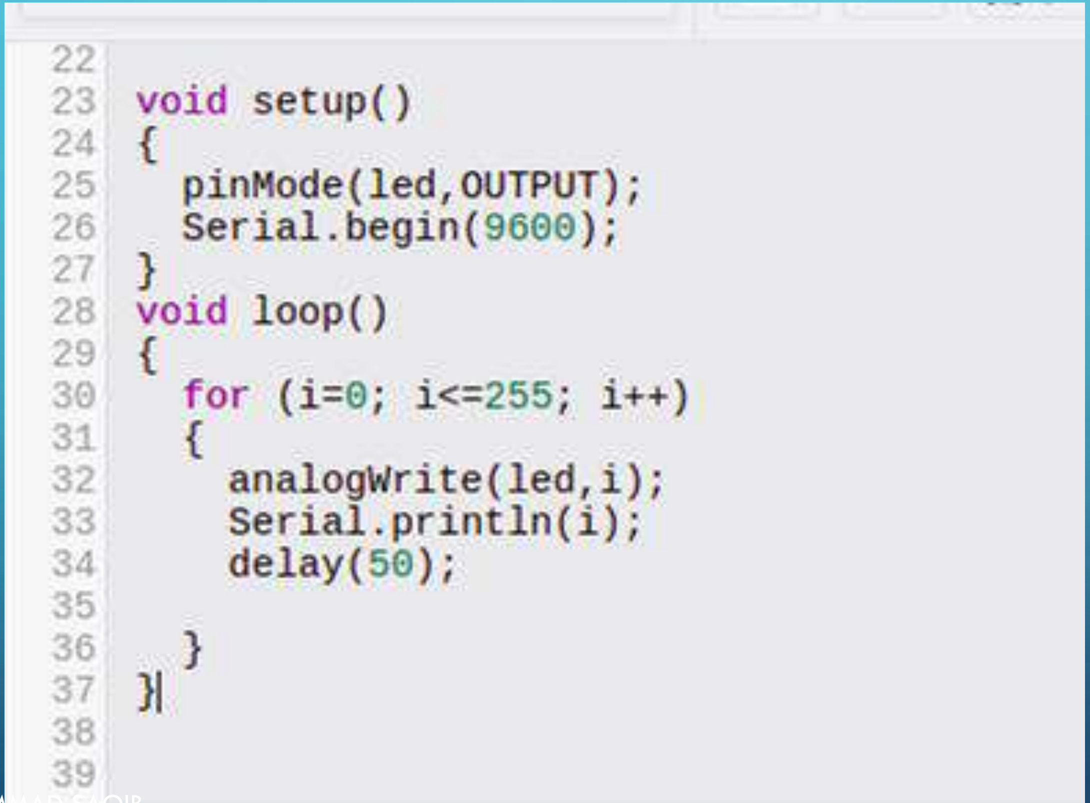
A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

syntax:

```
for (statement 1; statement 2; statement 3){
  }
```

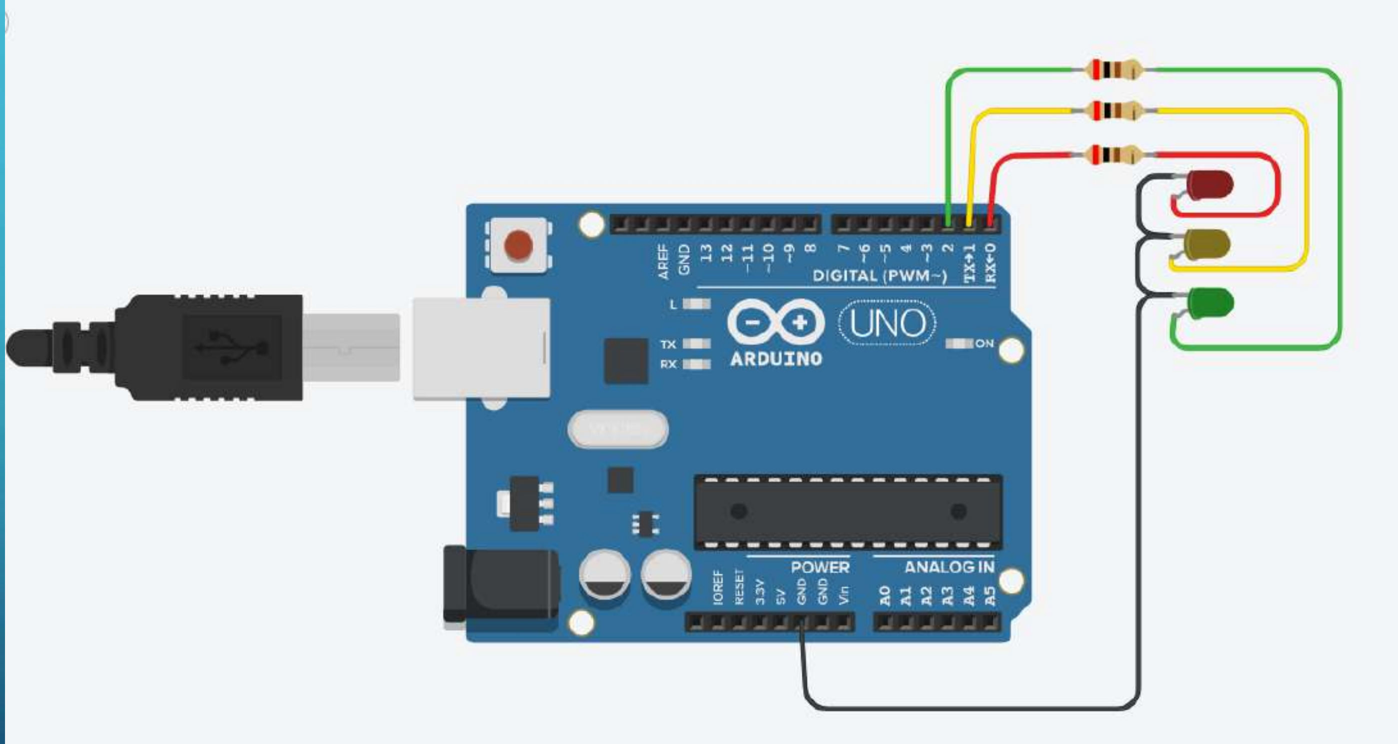
- Statement 1 sets a variable before the loop starts (int i = 0).
- Statement 2 defines the condition for the loop to run (e.g: i must be less than 5). If the condition is true, the loop will start over again, if it is false, the loop will end.
- Statement 3 increases a value (i++) each time the code block in the loop has been executed

LED BRIGHTNESS CONTROL USING FOR LOOP

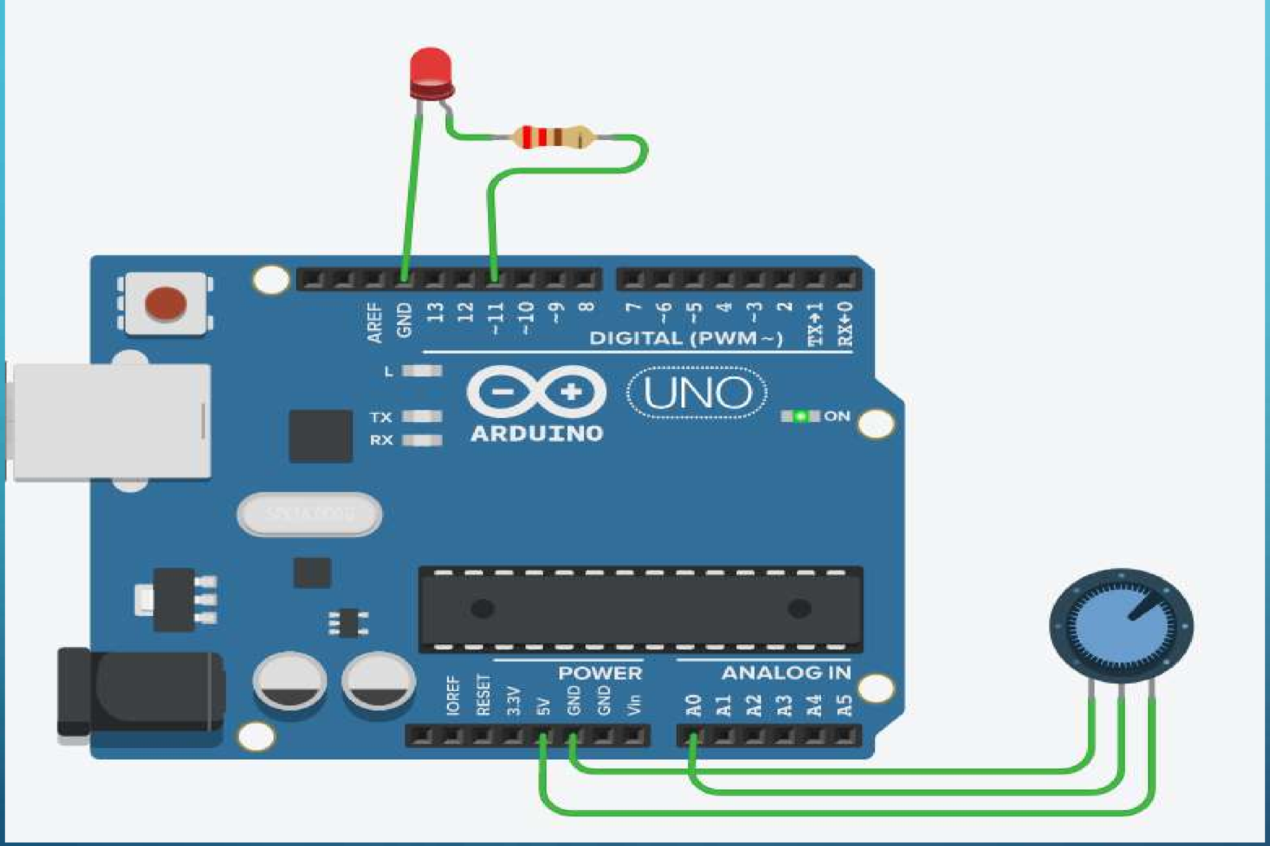


```
void setup()
{
  pinMode(led,OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  for (i=0; i<=255; i++)
  {
    analogWrite(led,i);
    Serial.println(i);
    delay(50);

  }
}
```

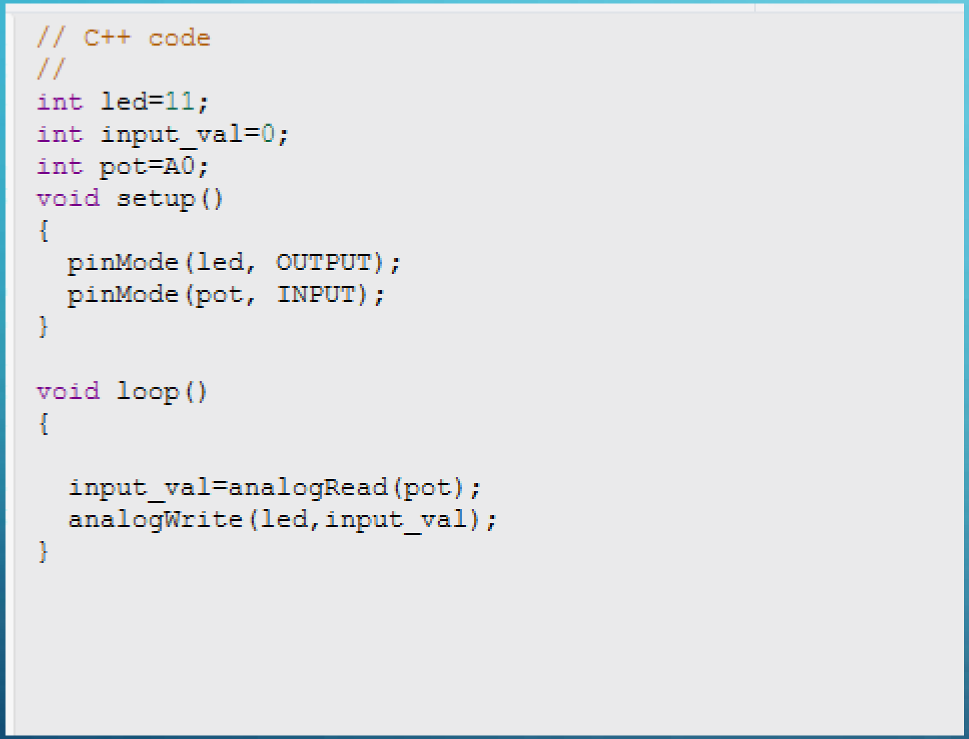
LED BLINKING



LED WITH POTENTIOMTER

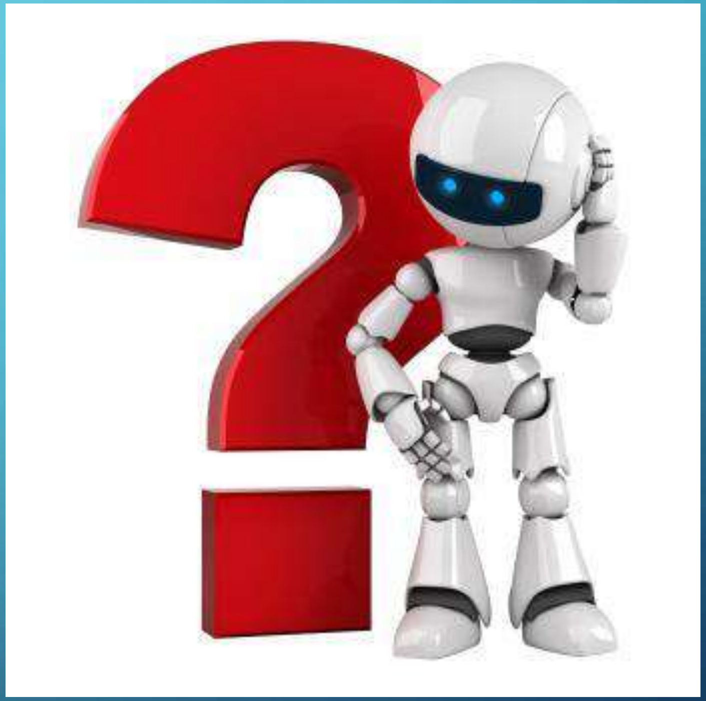


CODE



```
// C++ code
//
int led=11;
int input_val=0;
int pot=A0;
void setup()
{
  pinMode(led, OUTPUT);
  pinMode(pot, INPUT);
}

void loop()
{

  input_val=analogRead(pot);
  analogWrite(led,input_val);
}
```

Any Questions?



Thank You!