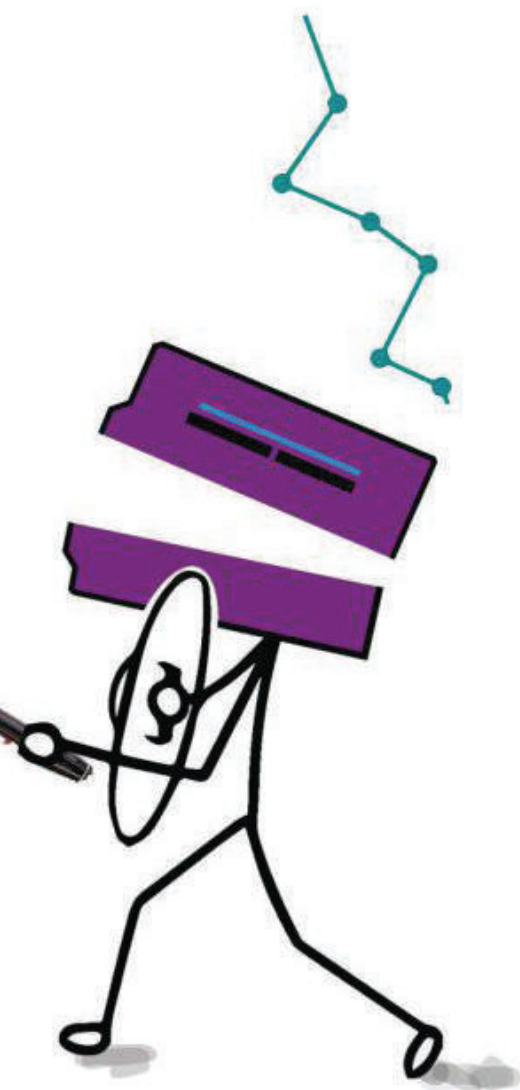
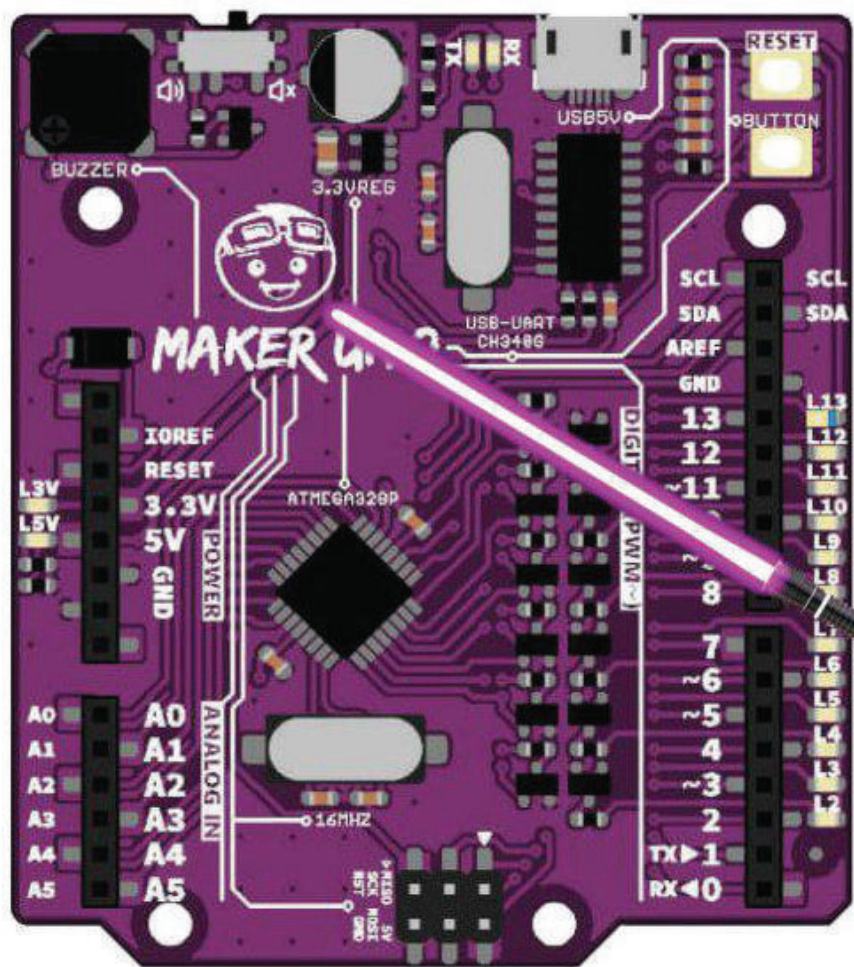


Getting Started With **MAKER UNO**

Edu Kit (Arduino Compatible)



Start Making Something!

TABLE OF CONTENTS



INTRODUCTION

Introduction to Components	
- Maker UNO	5
- Maker UNO Board	6
-	
Setting Up	
- Download Arduino IDE	7
- Install Maker UNO Drivers	11
- Install Maker UNO Board Package	13



LESSONS

1. The LED (Digital Output)	16
2. LED Blinking	19
3. LED Output (PWM)	22
4. Push Button as Digital Input	25
5. Serial Write	28
6. Serial Read	31
7. Tone Melody	34
8. Potentiometer as Analog Input	37
9. LDR as Analog Input	41
10. Controlling Motor	45

Editor :

Atifah Suad Anwar
Idris Zainal Abidin

Advisor and Supporter :

Phang Chin Yee
Ober Choo Sui Hong
Suhana Azmi
Lim Siaw Chiat
Tan Eng Tong



PROJECTS

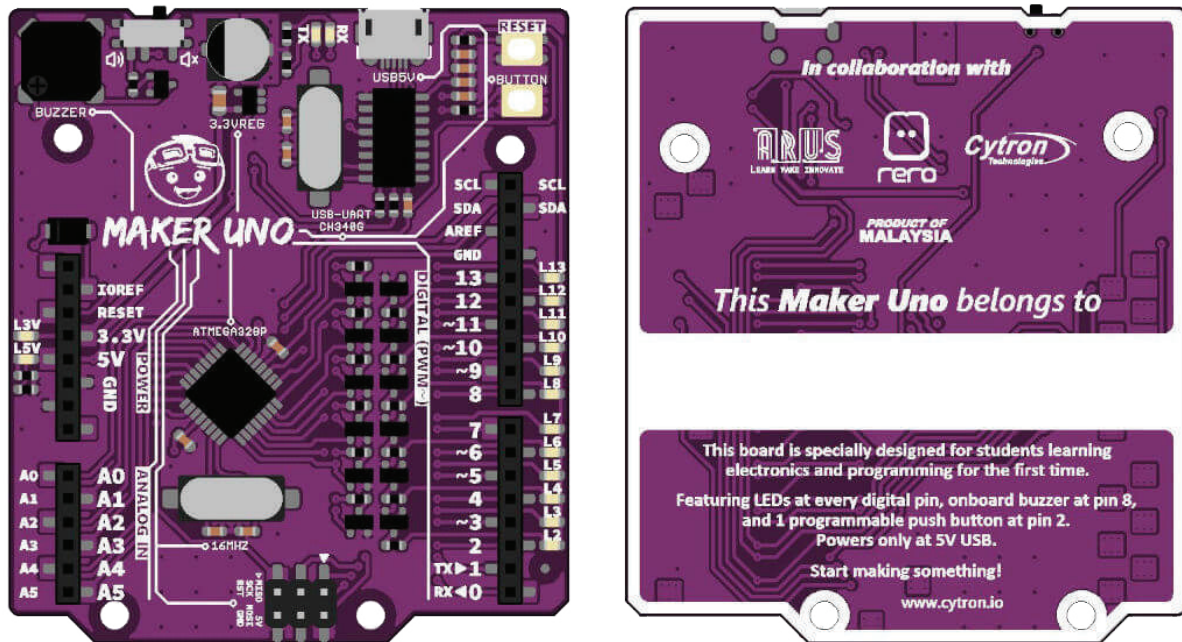
Interactive Traffic Light	50
Light Theremin	53



INTRODUCTION

MAKER-UNO

Maker UNO, an Arduino UNO compatible board designed and developed specially for students to learn coding and microcontroller. We named it Maker UNO to encourage everyone to be a maker by getting started with this amazing board..



MAKER - UNO Features:

- SMD ATmega328P microcontroller(the same microcontroller on Arduino UNO) with Optiboot (UNO) Bootloader.
- USB Programming facilitated by the CH340.
- Input voltage: USB 5V, from computer, power bank or standard USB adapter.
- 500mA (maximum) 3.3V voltage regulator.
- 0-5V outputs with 3.3V compatible inputs.
- 14 Digital I/O Pins (6 PWM outputs).
- 6 Analog Inputs.
- ISP 6-pin Header.
- 32k Flash Memory.
- 16MHz Clock Speed.
- R3 Shield Compatible.
- LED array for 5V, 3.3V, TX, RX and all digital pins.
- On board programmable push button (pin 2, need to configure as INPUT_PULLUP).
- On board piezo buzzer (pin 8).
- Utilize USB Micro-B socket.
- **PURPLE PCB!**

MAKER-UNO BOARD

Piezo Buzzer Slide Switch

Slide switch to connect between pin 8 to piezo buzzer. To use piezo buzzer, slide the switch on and program the buzzer. To use pin 8 for other purpose, slide the switch off.

Piezo Buzzer

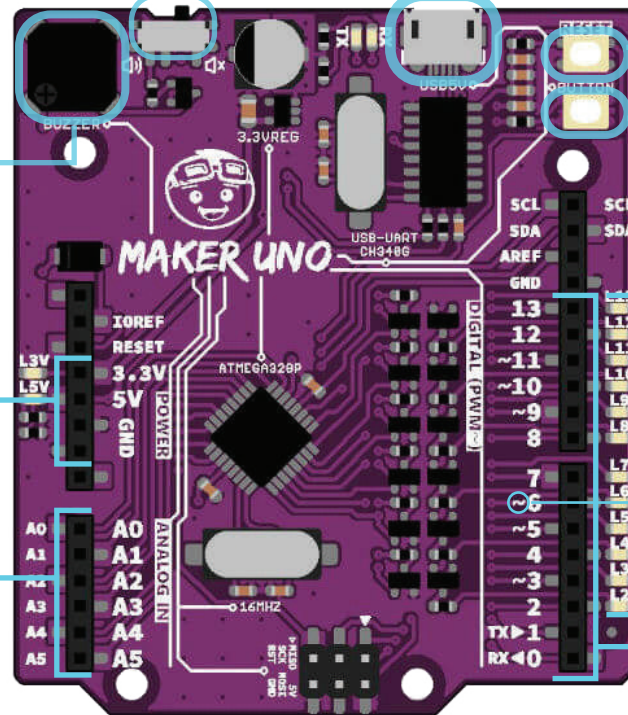
Piezo buzzer is connected to pin 8 through slide switch.

Power Pin

GND - Ground Pins
5V - Regulated 5V output
3V3 - Regulated 3.3v supply

Analog Pin

This pin can be used with `analogRead()`; to read an input in analog form (0-1023)



Micro USB B Type Connector (Female)

Main supply for Maker Uno. Used for program and debug purpose (Serial Monitor) too.

Reset Button

Button to restart Maker UNO program.

Programmable Button

This button is connected to pin 2 and GND. To use it, user need to configure it as `INPUT_PULLUP`.

Series of LED for Digital I/O

Every digital IO is equipped with LED, where you can control it or make it as indicator for input.

PWM Pin ~

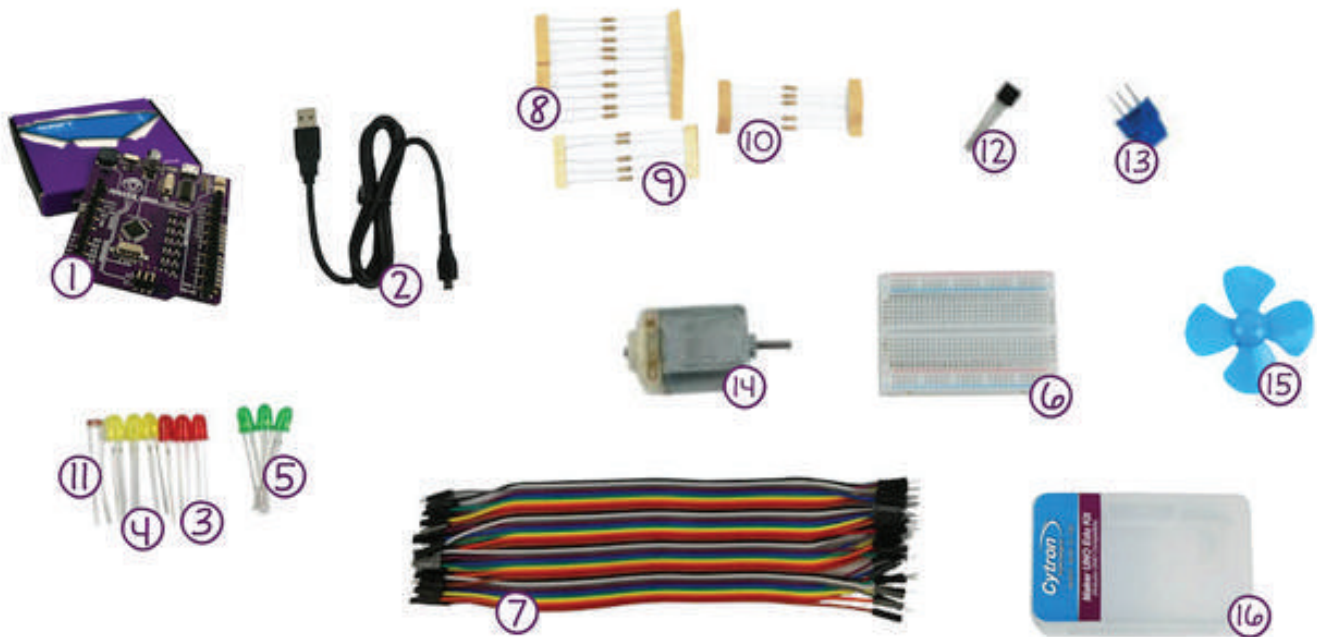
The digital pin that has this symbol can only use `analogWrite()`; to control the output. (0-255)

Digital Pin

This pin can be used with :
`digitalRead()`; as an input
`digitalWrite()`; as an output

MAKER UNO Edu Kit

The Most Affordable Arduino Kit to Kickstart Your Arduino Class



1. Maker UNO x 1

2. USB Micro B Cable x 1

3. LED 5mm Red x 3

4. LED 5mm Yellow x 3

5. LED 5mm Green x 3

6. Breadboard (Small) x 1

7. 40 Ways Male to Male Jumper Wire x 1

8. Resistor 0.25W 5% (220R) x 10

9. Resistor 0.25W 5% (1K) x 5

10. Resistor 0.25W 5% (10K) x 5

11. LDR (Small) x 1

12. Transistor 2N2222 x 1

13. Finger Adjust Preset 10K x 1

14. 3V Miniature Brush Motor w/o Gear x 1

15. DIY 4 Blades 56mm Motor Propeller x 1

16. Plastic Box x 1

DOWNLOADING ARDUINO IDE

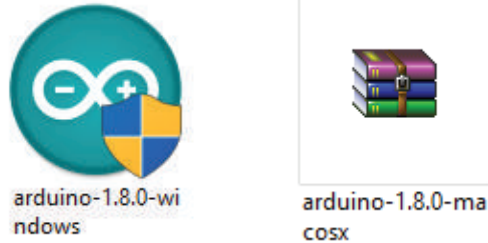
Maker UNO requires Arduino software to run. You can download the software from Arduino website (<http://arduino.cc/en/Main/Software>) and it is free to use.



Arduino IDE is compatible with Windows, Mac OS X and also Linux. You just need to choose the appropriate operating system installation package for your computer.

***Note:** If you are a Windows user, it is recommended that you choose Windows (installer).

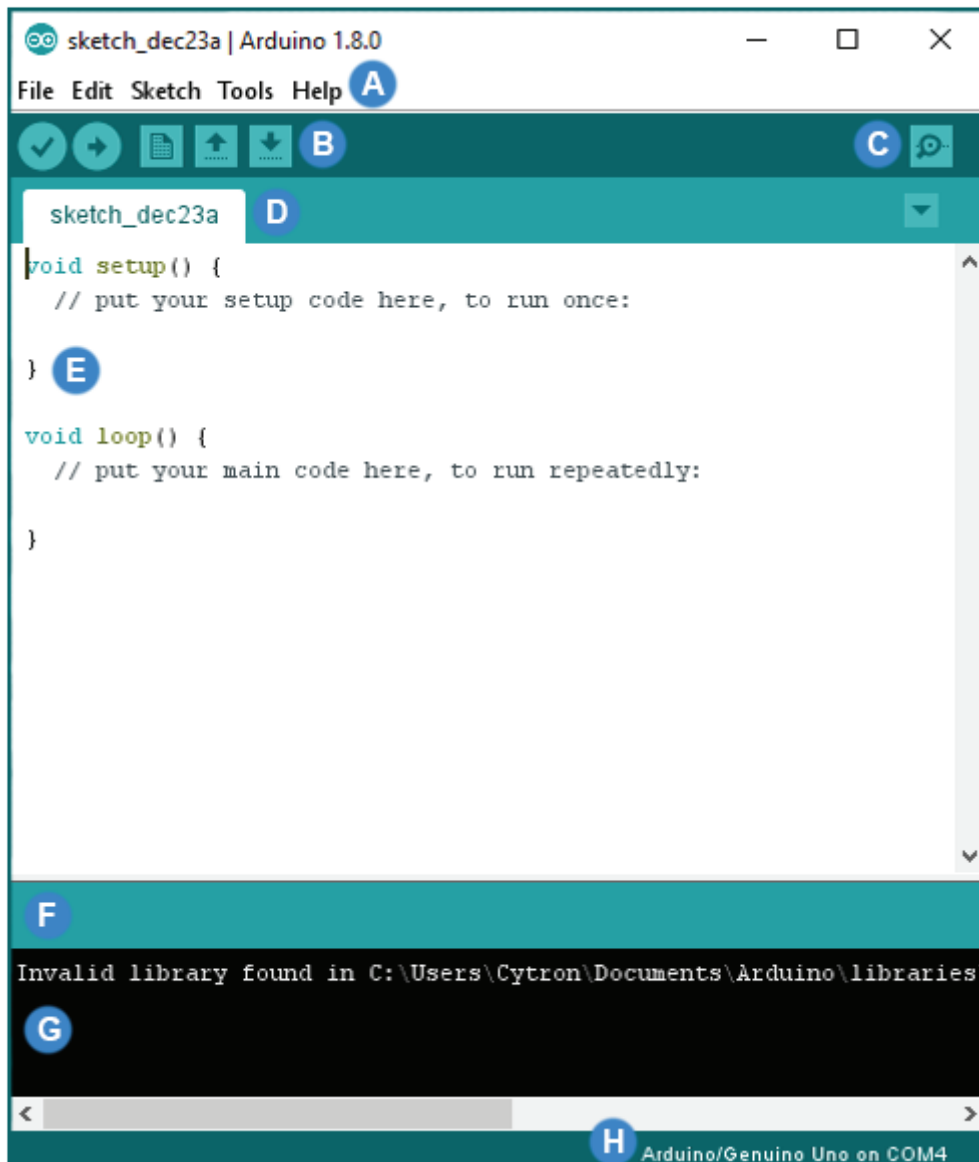
Choose the installer that compatible with your laptop OS and download the Arduino IDE. You will have arduino-1.8.x-windows.exe software after finish downloading for Windows OS user while for Mac OS user, you will get a zip file of arduino-1.8.x-macosx zip file as shown below :



***Note:** For latest version of Arduino IDE, go to <https://www.arduino.cc/en/Main/Software>

Double-click on the icon to install Arduino IDE. Complete the download, proceed with the installation as usual. After finish installing the software, you can start using it by double-click on the icon. Then, you will see this layout of Arduino IDE.





Label	Description	Label	Description
A	Menu Bar	E	Code Area
B	Button Bar	F	Status Bar
C	Serial Monitor	G	IDE Output
D	Sketch Name	H	Board Name and COM Number



Verify

Compiles and approves your code. It will detect errors in syntax (e.g. missing semi colon or parentheses).



Upload

Sends your code to the Maker UNO. When you click it, you should see the lights on your board blink rapidly.



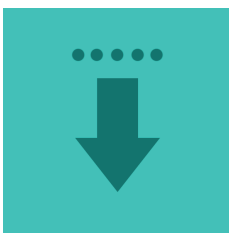
New Sketch

This button opens up a new code window tab.



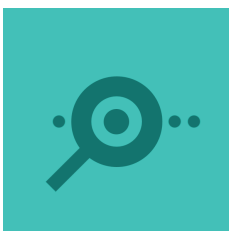
Open

This button will let you open an existing sketch.



Save

This saves the currently active sketch.

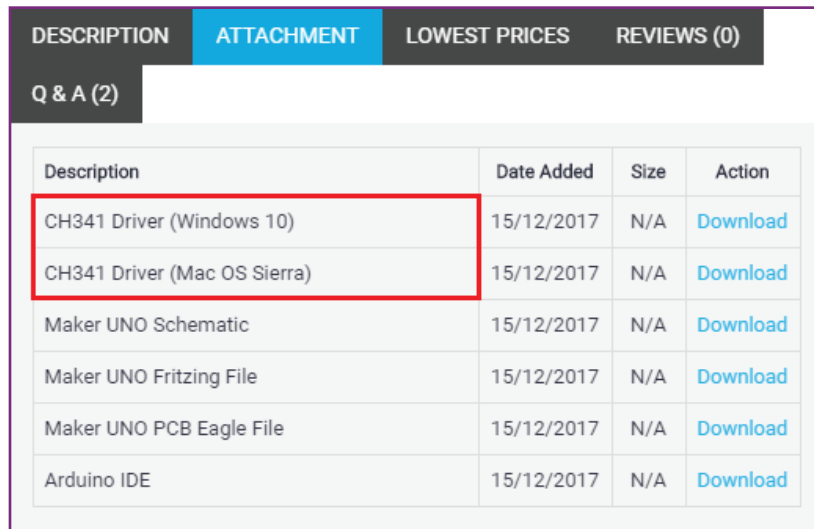


Serial Monitor

Open Serial Monitor.

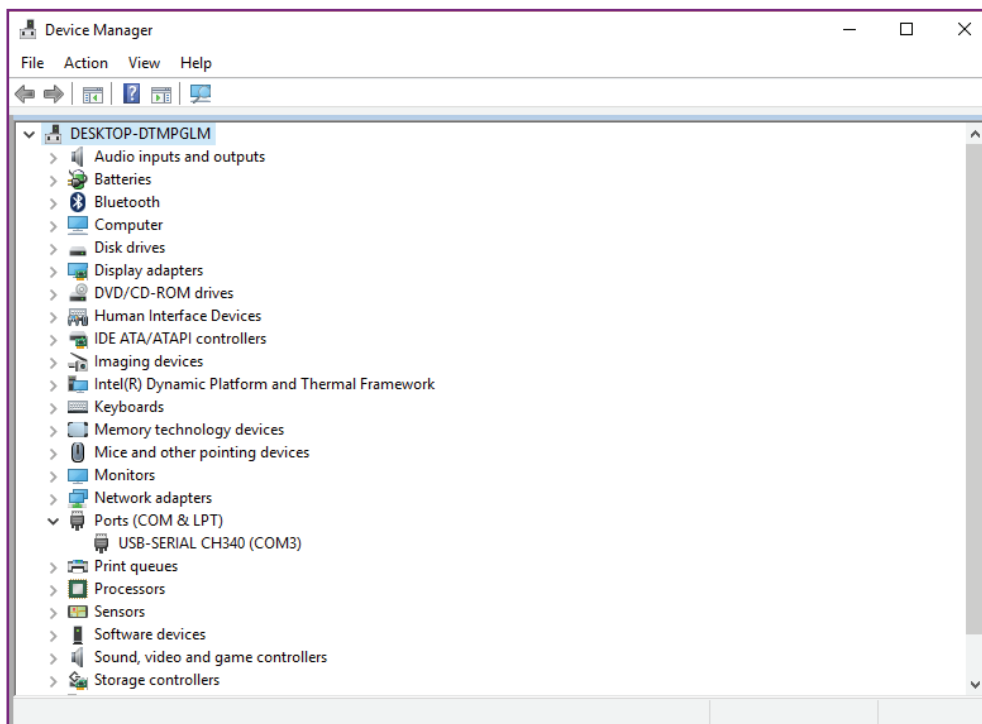
INSTALLING MAKER UNO DRIVER

Download Maker UNO driver at Maker Uno product page (under Attachment tab). Please choose appropriate driver depends on your OS. Complete the download, proceed with the installation as usual.

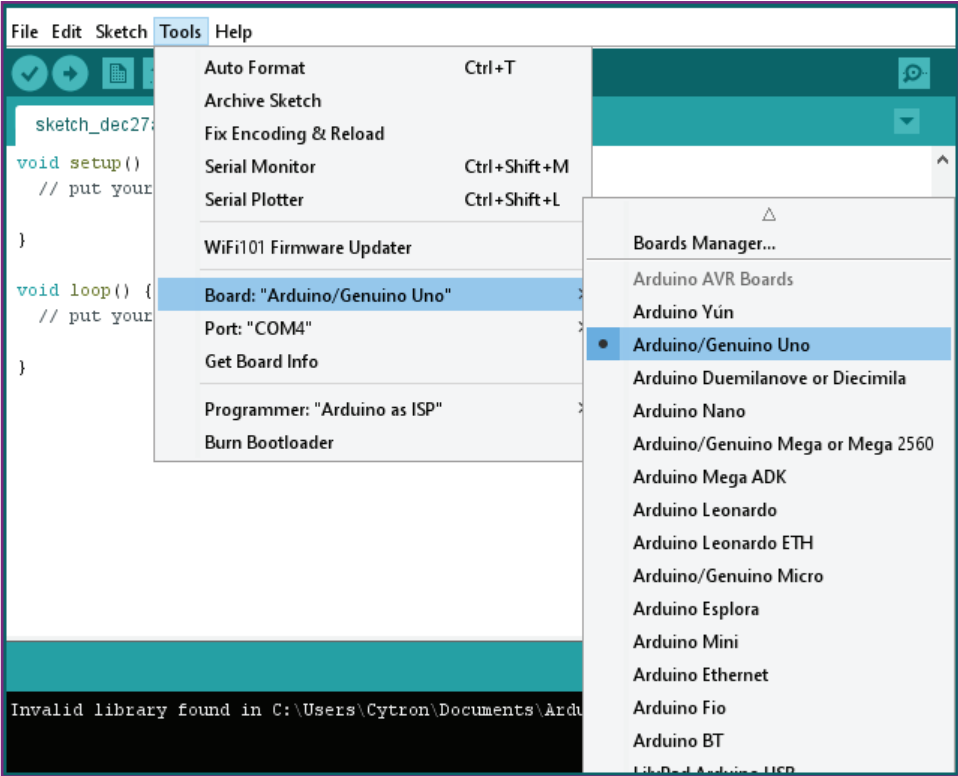


DESCRIPTION	ATTACHMENT	LOWEST PRICES	REVIEWS (0)
Q & A (2)			
Description	Date Added	Size	Action
CH341 Driver (Windows 10)	15/12/2017	N/A	Download
CH341 Driver (Mac OS Sierra)	15/12/2017	N/A	Download
Maker UNO Schematic	15/12/2017	N/A	Download
Maker UNO Fritzing File	15/12/2017	N/A	Download
Maker UNO PCB Eagle File	15/12/2017	N/A	Download
Arduino IDE	15/12/2017	N/A	Download

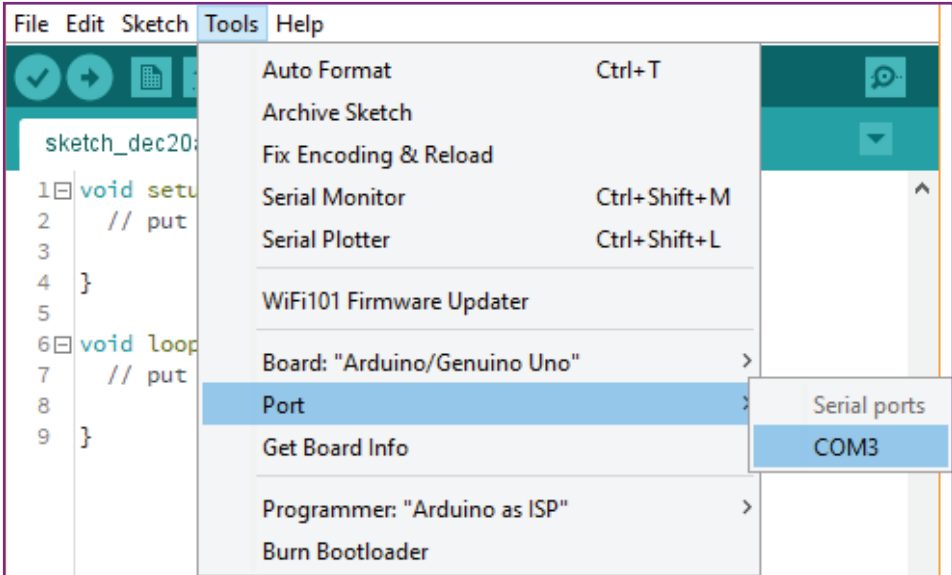
After installation is complete, your Maker UNO port should appear at Device Manager under Ports (COM & LPT) - e.g. USB-SERIAL CH340 (COM3). Please remember the port number.



Select Board :



Select Serial Port :

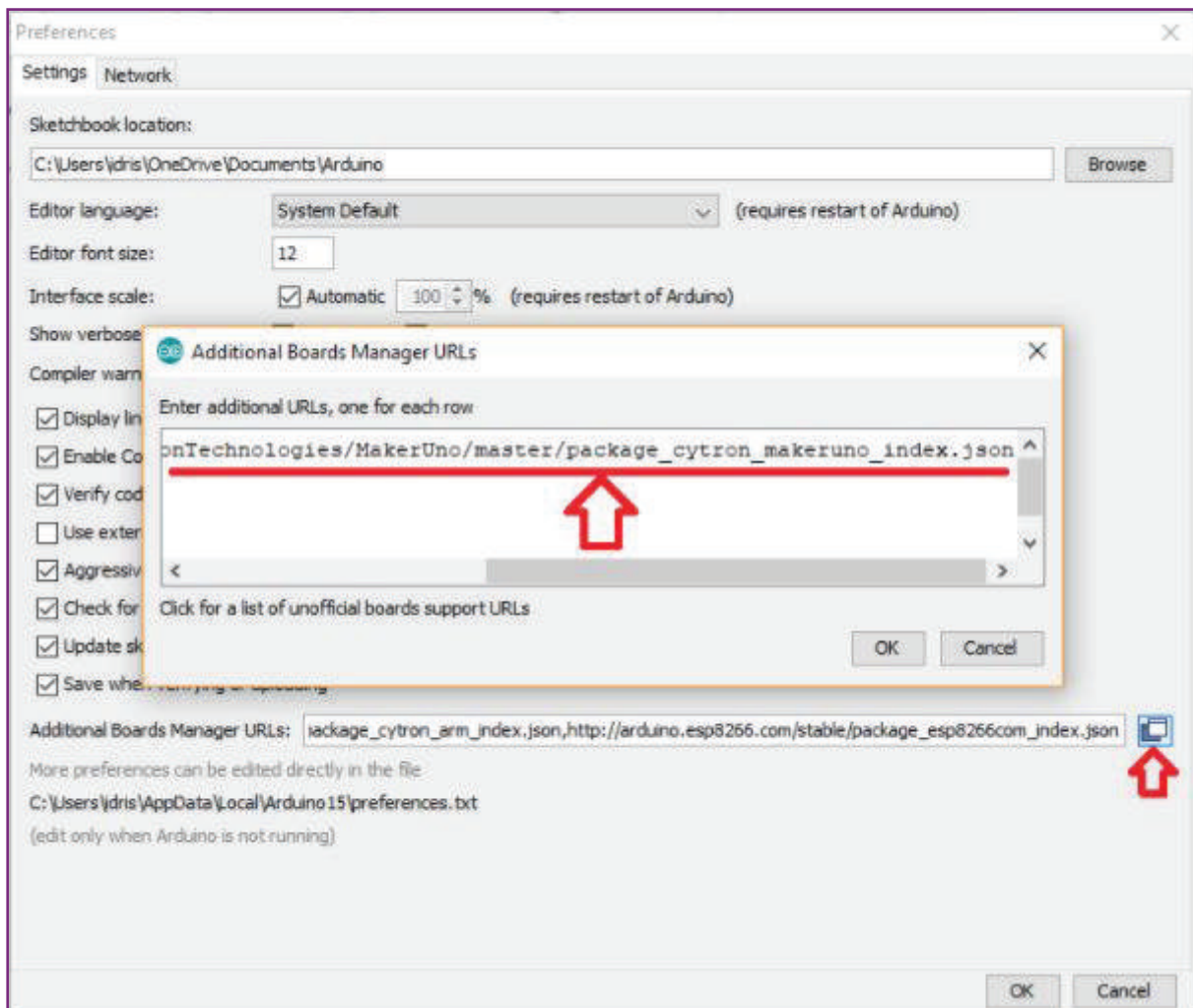


INSTALLING MAKER UNO BOARD PACKAGE

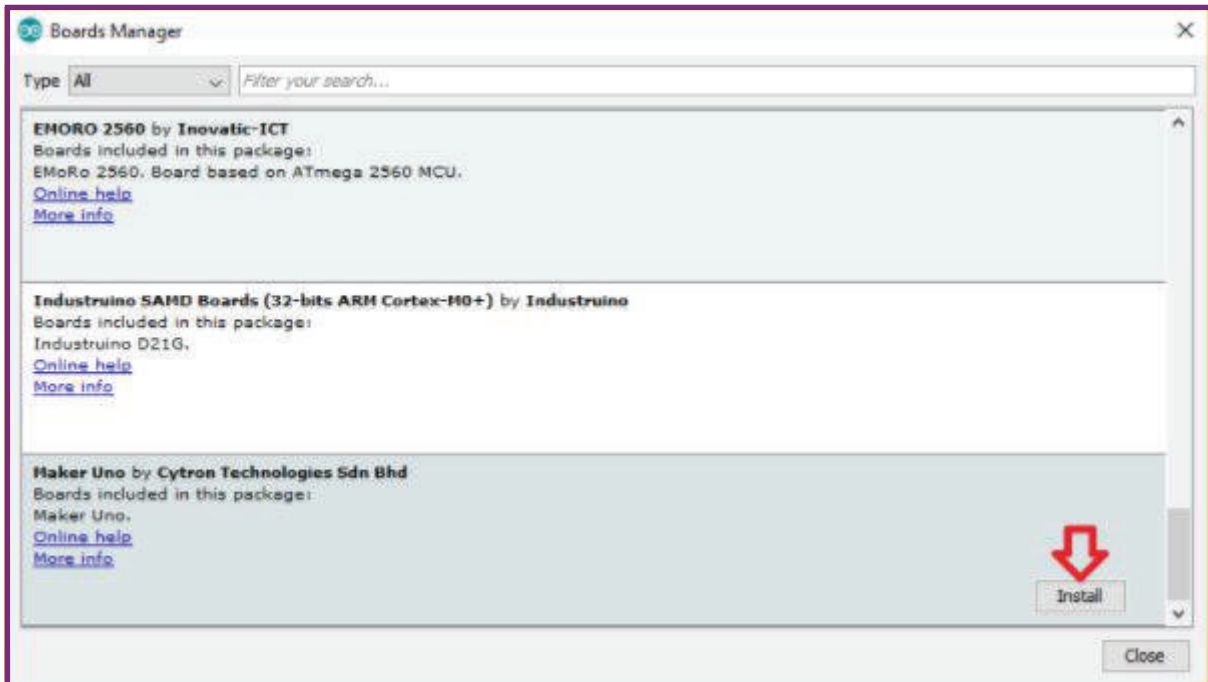
Maker UNO comes with board package too. You can install it using the URL link below. Copy the URL link below and paste it into the Additional Boards Manager URLs (Go to File > Preferences). Then click OK.

The link:

https://cytrontechnologies.github.io/package_cytron_makeruno_index.json



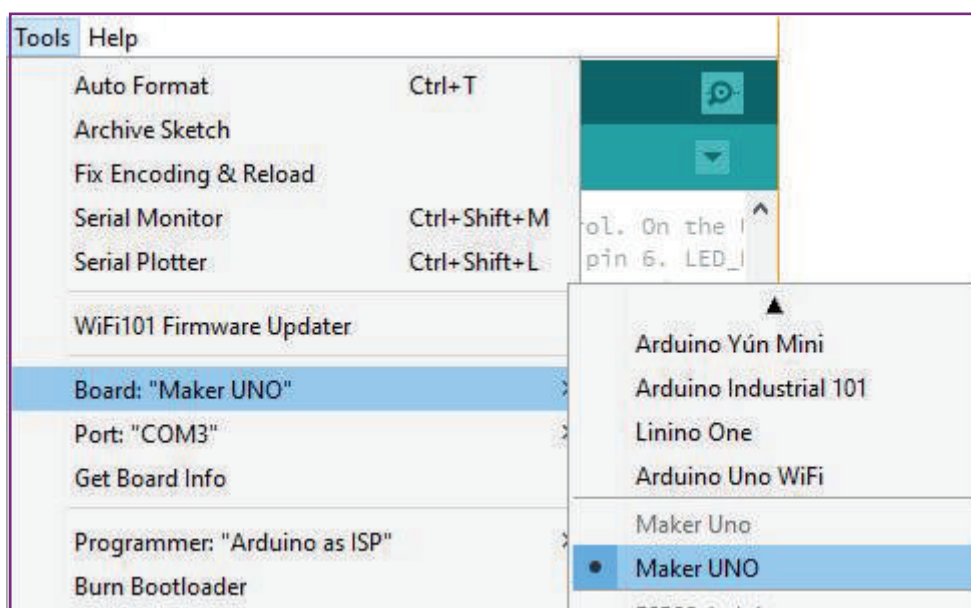
Open Boards Manager (Go to Tools > Board > Boards Manager...), find Maker Uno board package from Cytron Technologies and install.



NOTE

You need an internet connection during board package installation.

Once installed, you should be able to select Maker UNO as your Board (Go to Tools > Board), and proceed with uploading as usual.



CAUTIONS

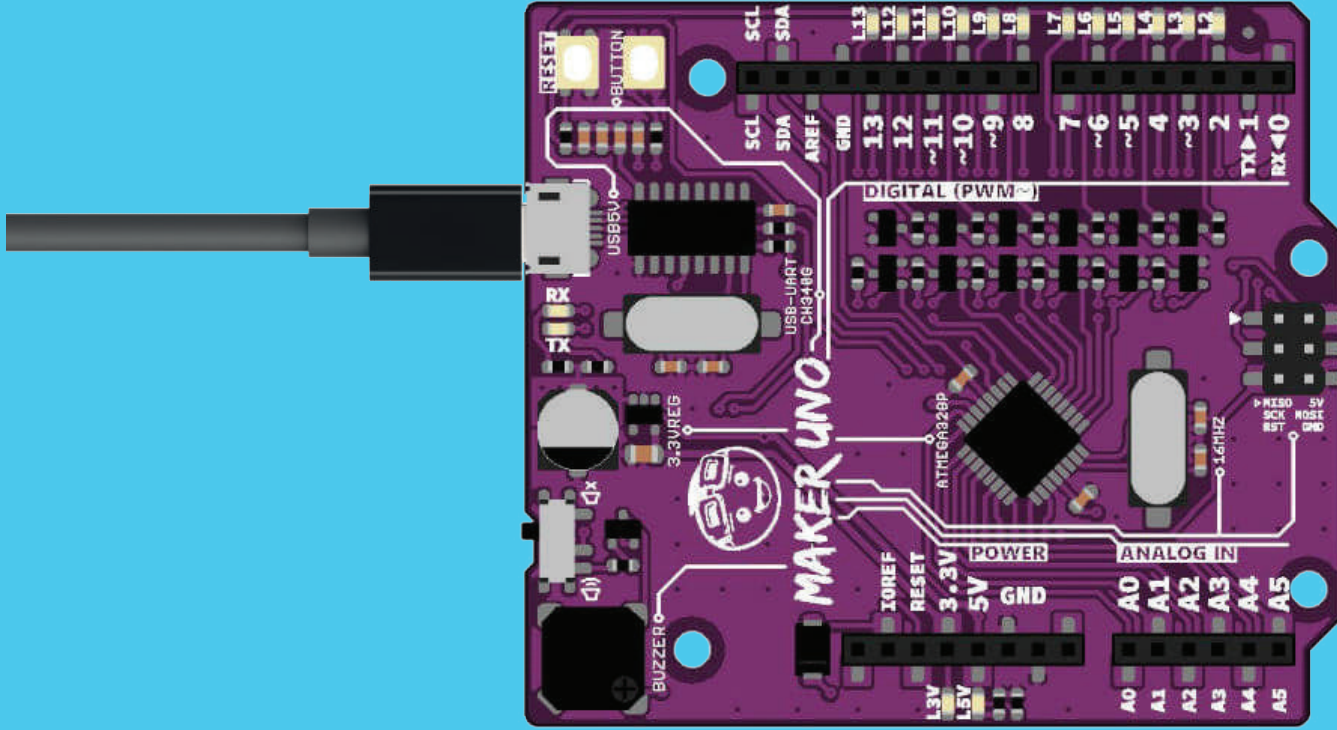
What happens if I just select Arduino/Genuino Uno instead of Maker UNO?

Arduino/Genuino Uno by default will set all IO pins to INPUT. If you upload the Blink example to Maker UNO, LED 13 will start to blink and also other LEDs light up randomly, (i.e. sometimes all LED turns on). This is because the voltage on the pins set to INPUT are floating. If you don't set those pins to OUTPUT or connect them to an external circuit, it will produce a floating voltage (i.e. 1V, 2V or any voltage in range of 0-5V). This floating voltage is enough to light up the LED occasionally. Actually this is not a problem, just confusing for newbies/beginners trying to learn using a Maker UNO.

We create a custom board package for Maker UNO to eliminate this confusion. The Maker UNO board package initializes all digital IO pins to OUTPUT by default. Now when you upload the Blink example, only pin 13 will blink, other pins will remain off. We feel this approach is better for a newbie/beginner, HOWEVER we advise to use this method only for early lessons. Once you understand the reasons to set a pin as INPUT and OUTPUT, we advise you to select Arduino/Genuino Uno as a board for safety purposes.



LESSON 1: THE LED (DIGITAL OUTPUT)





LESSON 1: LIGHT UP THE LED (IDE)

```
File Edit Sketch Tools Help
Lesson_1 $
1 void setup() {
2   // put your setup code here, to run once:
3 }
4
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8 }
9
10
Done uploading.
Sketch uses 724 bytes (2%) of program storage space. Maximum is 32.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039
Invalid library found in C:\Users\Cytron\Documents\Arduino\librar
Invalid library found in C:\Users\Cytron\Documents\Arduino\librar
8 Arduino/Genuino Uno on COM4
```

i LED is a light emitting diode. It will light up when a proper voltage is applied in correct direction.

1 Open new sketch on Arduino IDE.

2 **Write** this code to your sketch :

```
void setup()
{ // put your setup code here, to run once:
  pinMode(7, OUTPUT);
}

void loop()
{ // put your main code here, to run
  repeatedly:
  digitalWrite(7, HIGH);
}
```

```
File Edit Sketch Tools Help
Lesson_1
1 void setup()
2 { // put your setup code here, to run once:
3   pinMode(7, OUTPUT);
4 }
5
6 void loop()
7 { // put your main code here, to run repeatedly:
8   digitalWrite(7, HIGH);
9 }
```

Done uploading.

Sketch uses 724 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes free.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries\...
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries\...

8 Arduino/Genuino Uno on COM4

3 **Compile** the file.

4 **Upload** the sketch.

5 You will see status of **“Done Uploading”** if everything is correct your **LED** at pin 7 will **light up**.

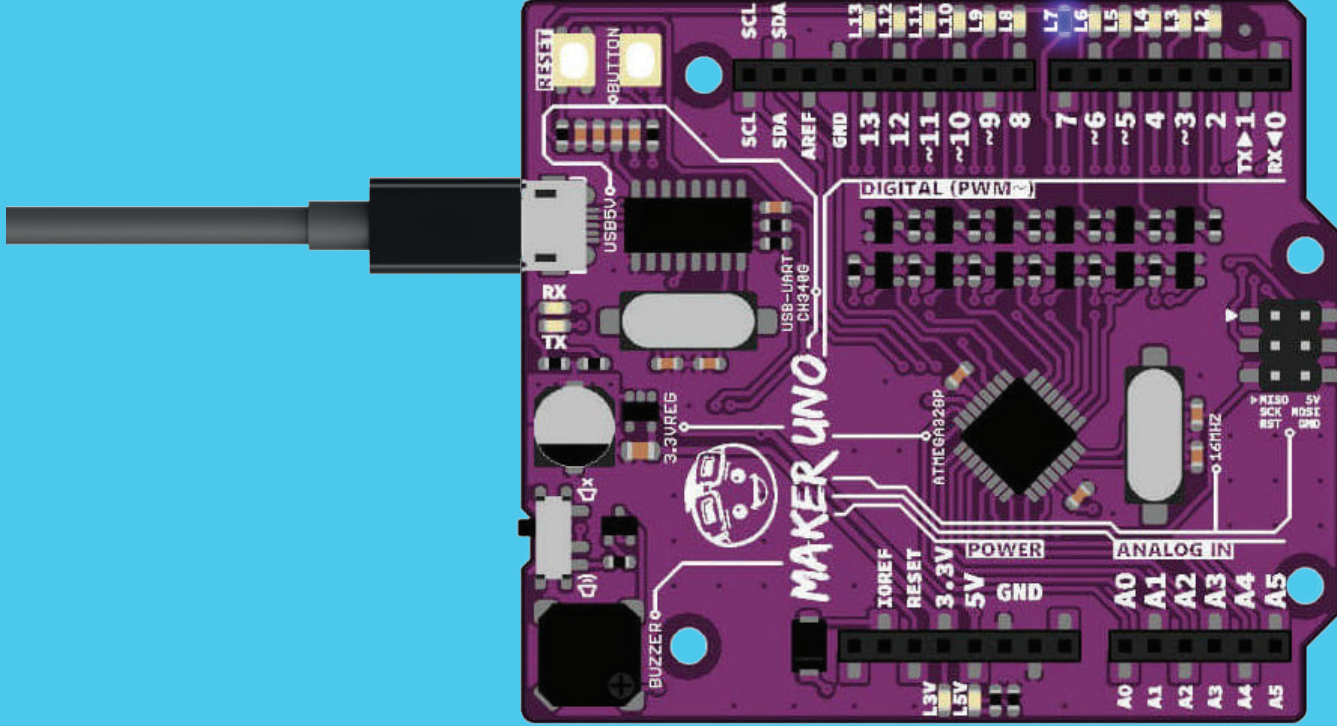
The **void setup()** runs once when the Maker UNO is powered on. The code in the void setup() usually use to configure the pin as INPUT or OUTPUT using **pinMode()**;

The **void loop()** runs continuously after the void setup() has complete. The code in the void loop() usually use to control the INPUT and OUTPUT.

The digitalWrite(); is used to set the digital OUTPUT of the pin number to HIGH or LOW



LESSON 2: LED (BLINKING)





LESSON 2: LED BLINKING (IDE)

```
1 void setup()
2 {
3   pinMode(7, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(7, HIGH);
9   delay(1000);
10  digitalWrite(7, LOW);
11  delay(1000);
12 }
```

Done uploading.

Sketch uses 928 bytes (2%) of program storage space. Maximum is 32768 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes free.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries

10 Arduino/Genuino Uno on COM4

i

LED will blink when **delay** is applied between ON and OFF. Then it will be blinking!

1

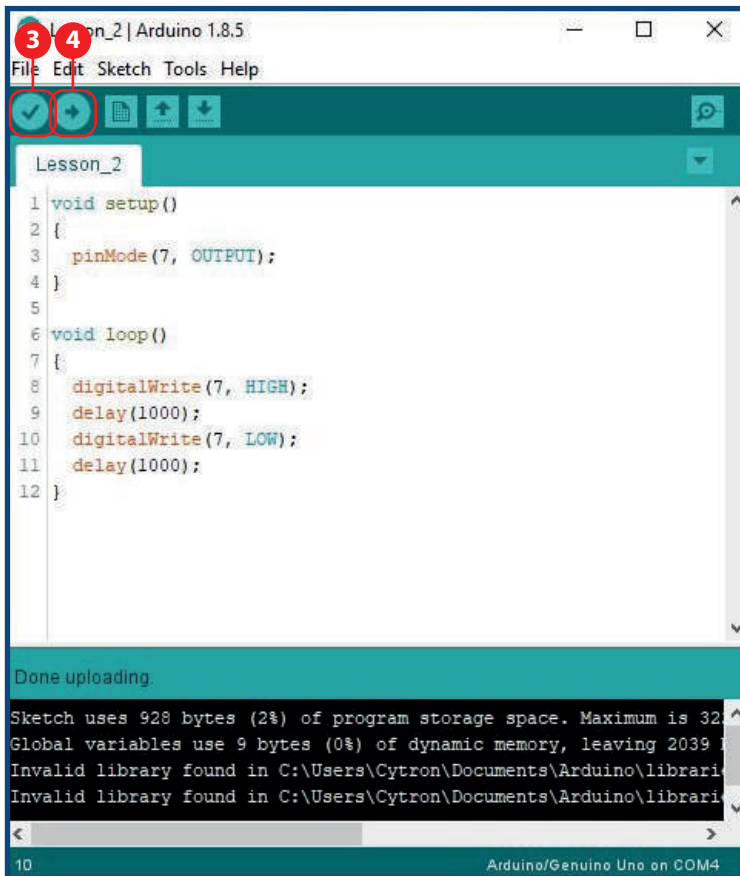
Open new sketch on Arduino IDE.

2

Write this code to your sketch :

```
void setup()
{
  pinMode(7, OUTPUT);
}

void loop()
{
  digitalWrite(7, HIGH);
  delay(1000);
  digitalWrite(7, LOW);
  delay(1000);
}
```



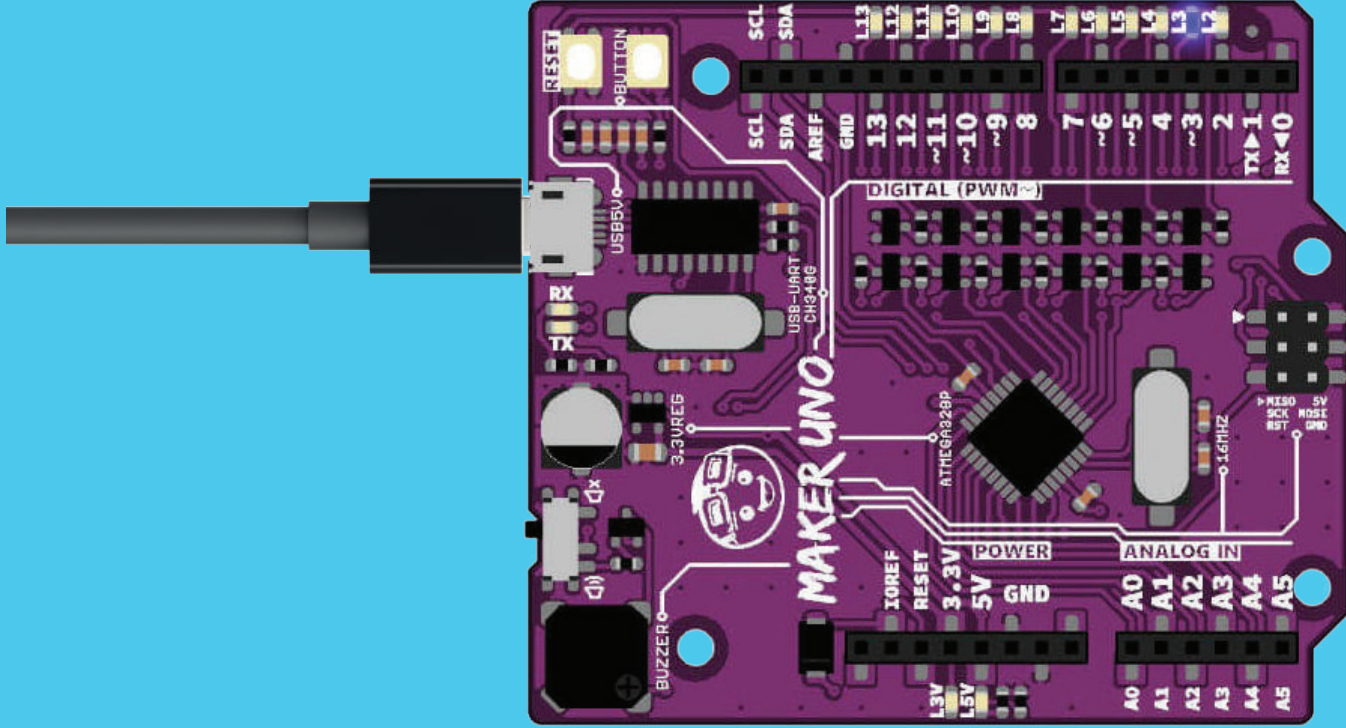
- 3 **Compile** the file.
- 4 **Upload** the sketch.
- 5 You will see status of **“Done Uploading”** if everything is correct and your **LED** will **blink**.

The **digitalWrite(7, HIGH);** digital pin number 7 is set to HIGH which is to turn ON the LED while the **digitalWrite(7, LOW);** digital pin number 7 is set to LOW which is to turn OFF the LED.

The **delay();** is a function to make the Maker UNO execute anything for the time set in milliseconds. 1000 is equal to 1second.



LESSON 3: FADE AN LED





LESSON 3: FADE AN LED(IDE)

```
File Edit Sketch Tools Help
Lesson_3
1 int LED = 3;
2 int brightness = 0;
3 int fadeAmount = 5;
4
5 void setup()
6 {
7   pinMode(LED, OUTPUT);
8 }
9 void loop()
10 {
11   analogWrite(LED, brightness);
12   brightness = brightness + fadeAmount;
13   if (brightness <= 0 || brightness >= 255)
14   {
15     fadeAmount = -fadeAmount;
16   }
17   delay(30);
18 }
```

Done uploading.

Sketch uses 1150 bytes (3%) of program storage space. Maximum is 32768 bytes.
Global variables use 13 bytes (0%) of dynamic memory, leaving 2035 bytes free.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries\

7 Arduino/Genuino Uno on COM4

i

The LED will fade using `analogWrite()` function using Pulse Width Modulation (PWM) which make a digital output acting as analog output.

1

Open new sketch on Arduino IDE.

2

Write this code to your sketch :

```
int LED = 3;
int brightness = 0;
int fadeAmount = 5;

void setup()
{
  pinMode(3, OUTPUT);
}

void loop()
{
  analogWrite(LED, brightness);
  brightness = brightness + fadeAmount;
  if (brightness <= 0 || brightness >= 255)
  {
    fadeAmount = -fadeAmount;
  }
  delay(30);
}
```

```
1 int LED = 3;
2 int brightness = 0;
3 int fadeAmount = 5;
4
5 void setup()
6 {
7   pinMode(LED, OUTPUT);
8 }
9 void loop()
10 {
11   analogWrite(LED, brightness);
12   brightness = brightness + fadeAmount;
13   if (brightness <= 0 || brightness >= 255)
14   {
15     fadeAmount = -fadeAmount;
16   }
17   delay(30);
18 }
```

Done uploading.

Sketch uses 1150 bytes (3%) of program storage space. Maximum is 32768 bytes.
Global variables use 13 bytes (0%) of dynamic memory, leaving 2035 bytes free.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries

7 Arduino/Genuino Uno on COM4

3 **Compile** the file.

4 **Upload** the sketch.

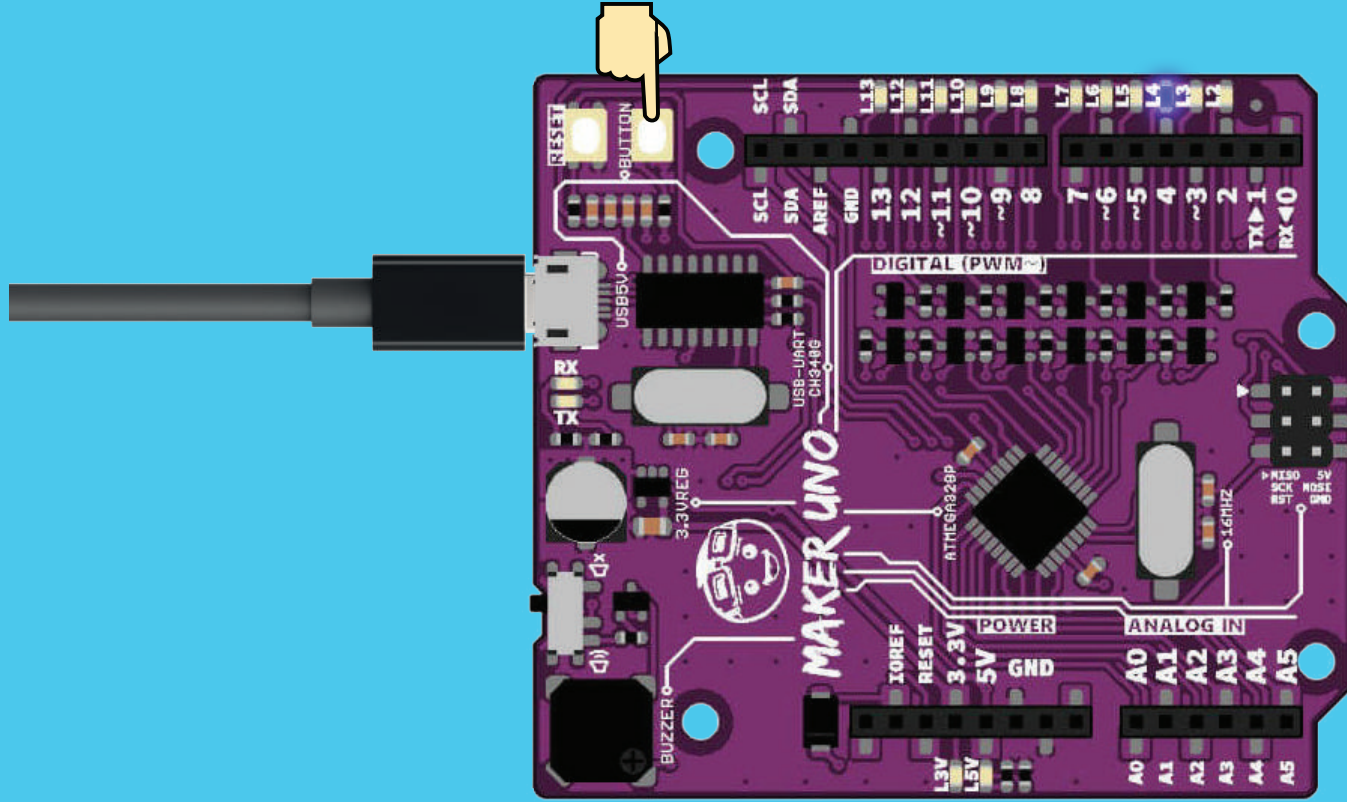
5 You will see status of **“Done Uploading”** if everything is correct and your **LED** will **fade**.

The **analogWrite()** function uses **PWM**, so if you want to change the pin you're using, be sure to use another PWM capable pin. On most Arduino, the PWM pins are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.

The **analogWrite(LED, brightness);** set OUTPUT of the pin number 3 to variable **“brightness”**. The LED will light up based on the amount of variable **“brightness”**.



LESSON 4 : PUSH BUTTON (DIGITAL INPUT)





LESSON 4: PUSH BUTTON (IDE)

```
File Edit Sketch Tools Help
Lesson_4
1 int LED = 4;
2 int Button = 2;
3
4 void setup()
5 {
6   pinMode(4, OUTPUT);
7   pinMode(2, INPUT_PULLUP);
8 }
9
10 void loop()
11 {
12   if (digitalRead(Button) == LOW)
13     digitalWrite(LED, HIGH);
14
15   else if (digitalRead(Button) == HIGH)
16     digitalWrite(LED, LOW);
17 }
Done uploading.
Sketch uses 944 bytes (2%) of program storage space. Maximum is 32.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039
Invalid library found in C:\Users\Cytron\Documents\Arduino\librar
Invalid library found in C:\Users\Cytron\Documents\Arduino\librar
6 Arduino/Genuino Uno on COM4
```

i

Push button act as a digital input device. Maker UNO is able to sense 2 states for digital input, i.e. HIGH and LOW. Push the button and the LED will turn ON!

1

Open new sketch on Arduino IDE.

2

Write this code to your sketch :

```
int LED = 4;
int Button = 2;

void setup()
{
  pinMode(4, OUTPUT);
  pinMode(2, INPUT_PULLUP);
}

void loop()
{
  if (digitalRead(Button) == LOW)
    digitalWrite(LED, HIGH);

  else if (digitalRead(Button) == HIGH)
    digitalWrite(LED, LOW);
}
```

```
Lesson_4
1 int LED = 4;
2 int Button = 2;
3
4 void setup()
5 {
6   pinMode(4, OUTPUT);
7   pinMode(2, INPUT_PULLUP);
8 }
9
10 void loop()
11 {
12   if (digitalRead(Button) == LOW)
13     digitalWrite(LED, HIGH);
14
15   else if (digitalRead(Button) == HIGH)
16     digitalWrite(LED, LOW);
17 }
```

Done uploading.

Sketch uses 944 bytes (2%) of program storage space. Maximum is 32768 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes free.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries\...
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries\...

Arduino/Genuino Uno on COM4

- 3 **Compile** the file.
- 4 **Upload** the sketch.
- 5 You will see status of **“Done Uploading”** if everything is correct, when **button is pressed**, the **LED** pin 4 will light up.

Using **pinMode(INPUT_PULLUP)**, there is an internal 20K-ohm resistor is pulled to 5V. This configuration causes the input to read HIGH when the switch is open, and LOW when it is closed.

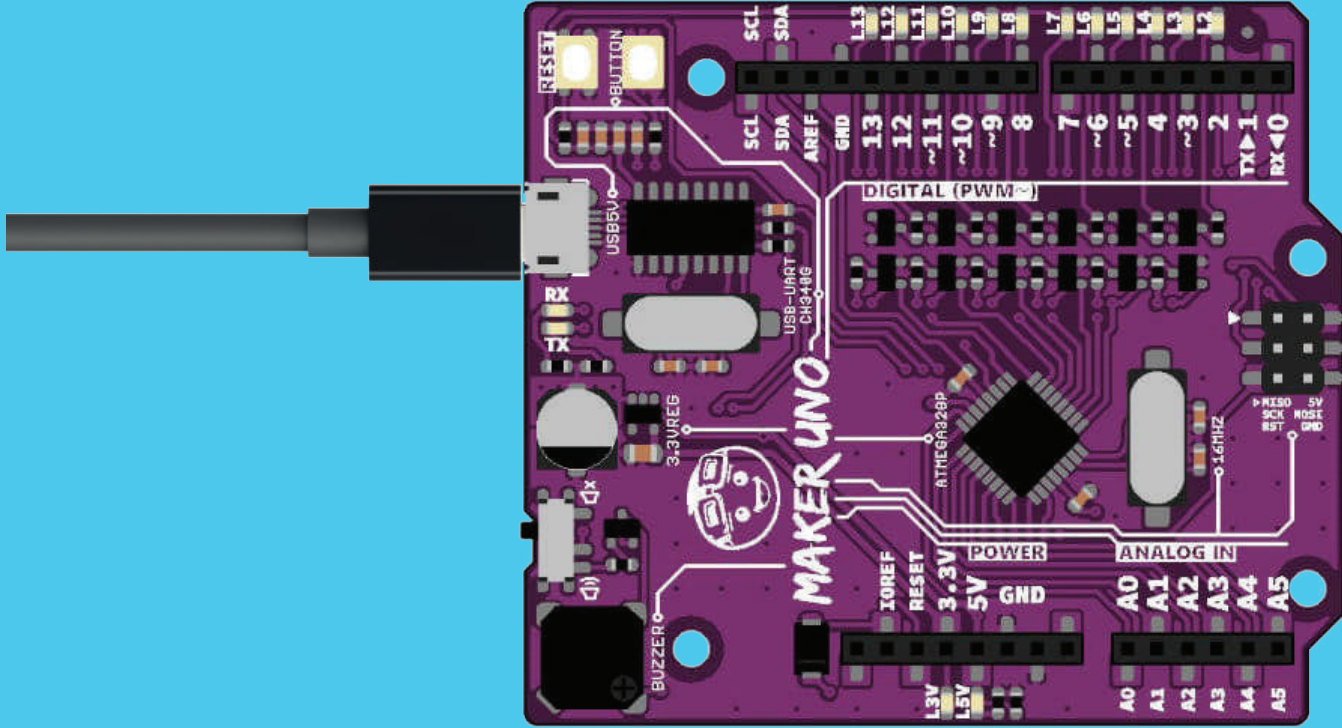
The **if()** statement is use to compare a condition whether it is **TRUE** or **FALSE**.

The **else if()** statement is use to set other condition than if() statement.

The **digitalRead(Button) == LOW;** will read the button input. If the button is pushed, the INPUT will be LOW.



LESSON 5 : SERIAL WRITE





LESSON 5: SERIAL WRITE (IDE)

```
File Edit Sketch Tools Help
Lesson_5
1 void setup()
2 {
3   Serial.begin(9600);
4   Serial.print("Hello, World!");
5 }
6 void loop()
7 {
8
9 }
```

Done uploading.
Invalid library found in C:\Users\Cytron\Documents\Arduino\librari
Invalid library found in C:\Users\Cytron\Documents\Arduino\librari

8 Maker UNO on COM4

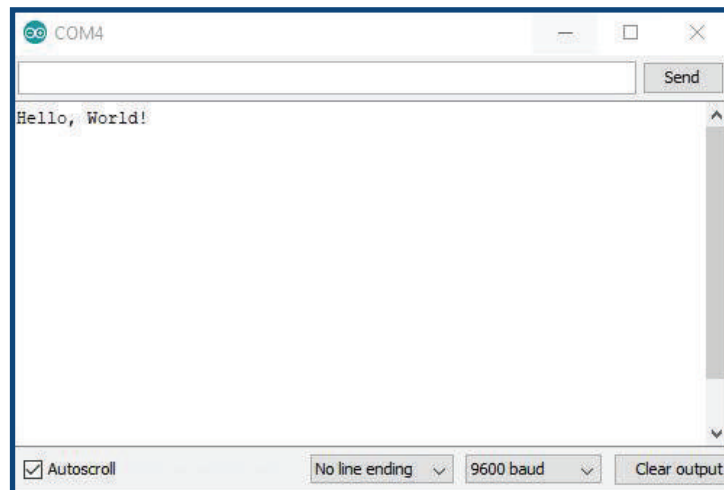
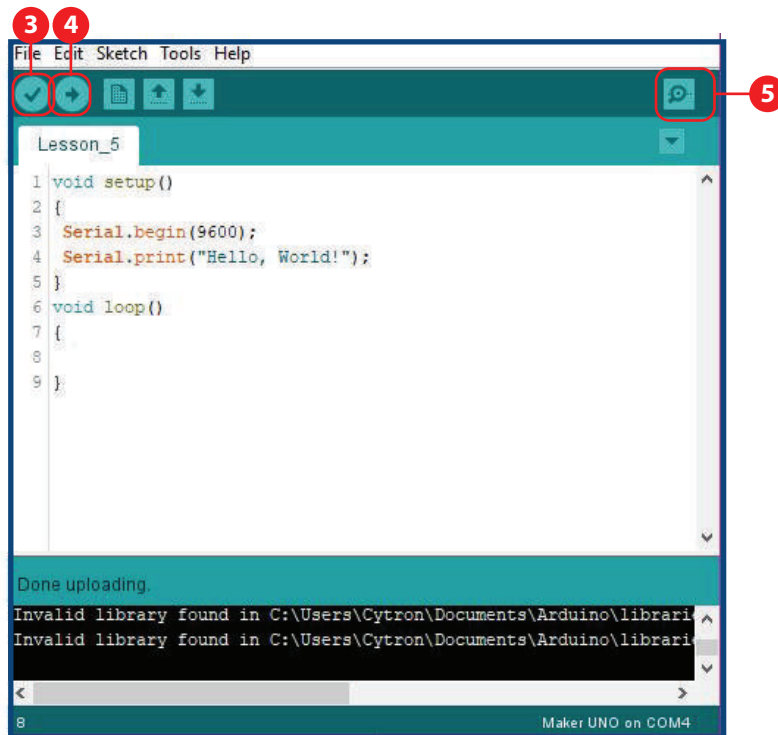
1

Open new sketch on Arduino IDE.

2

Write this code to your sketch :


```
void setup()
{
  Serial.begin(9600);
  Serial.print("Hello, World!");
}
void loop()
{
}
```



3 **Compile** the file.

4 **Upload** the sketch.

5 You will see the Button status through the Serial Monitor. Press the button to see the result!

i Click on the  symbol to see the result!

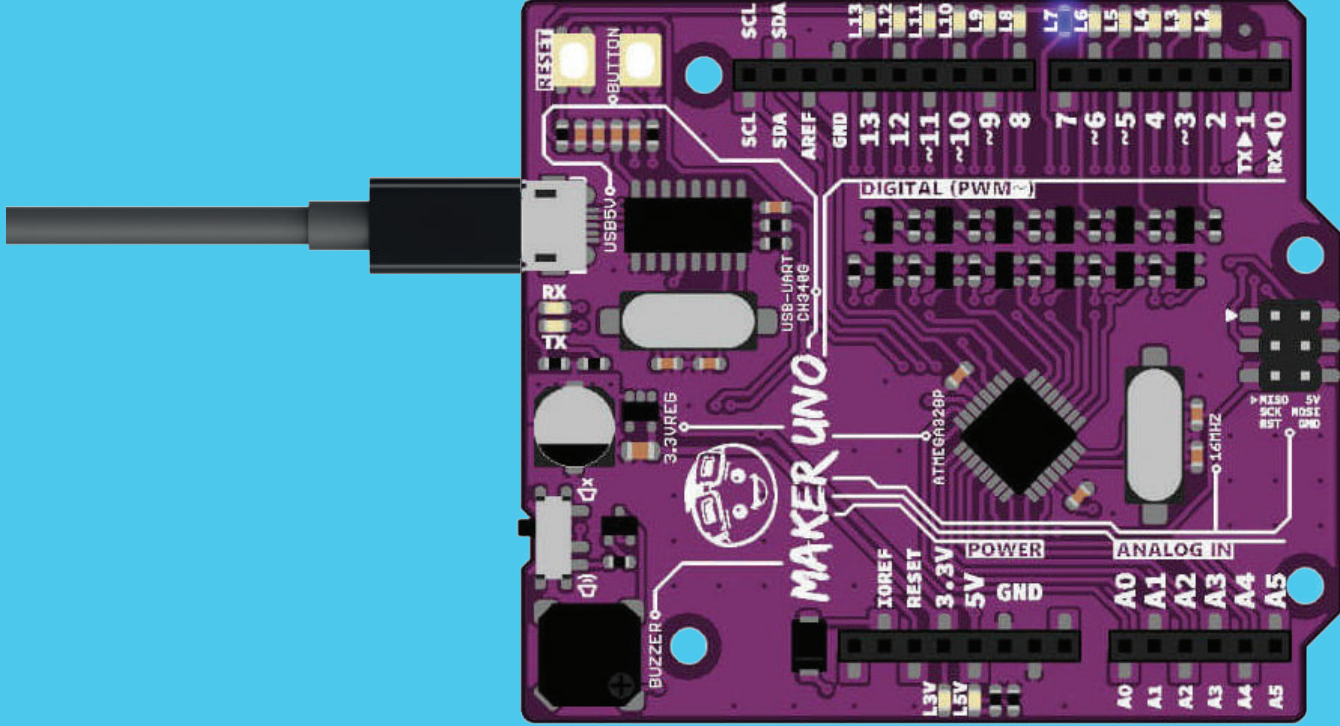
The **Serial.begin()**; open a serial communication between the Maker UNO and the computer. **9600** is the baud rate of the communication. The serial monitor must use the same baud rate to view the information.

The **Serial.print()**; sends information from Maker UNO to the connected computer. The information will be in the serial monitor.

The **Serial.println()**; sends information from Maker UNO to the connected computer. The information will be in the serial monitor and print out line by line.



LESSON 6 : SERIAL READ






LESSON 6: SERIAL READ (IDE)

```
File Edit Sketch Tools Help
Lesson_6
1 int LED = 7;
2 int data = 0;
3
4 void setup()
5 {
6   pinMode(LED, OUTPUT);
7   Serial.begin(9600);
8 }
9 void loop()
10 {
11   if(Serial.available() > 0 )
12   {
13     data = Serial.read();
14     if(data == '1')
15     {
16       digitalWrite(LED, HIGH);
17     }
18     else if(data == '0')
19     {
20       digitalWrite(LED, LOW);
21     }
22   }
23 }
```

Done uploading.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries

7 Arduino/Genuino Uno on COM4

i

Serial display can display numbers and characters (based on ASCII data) on the Arduino Serial Monitor. Click on the  symbol to enter the input.

1

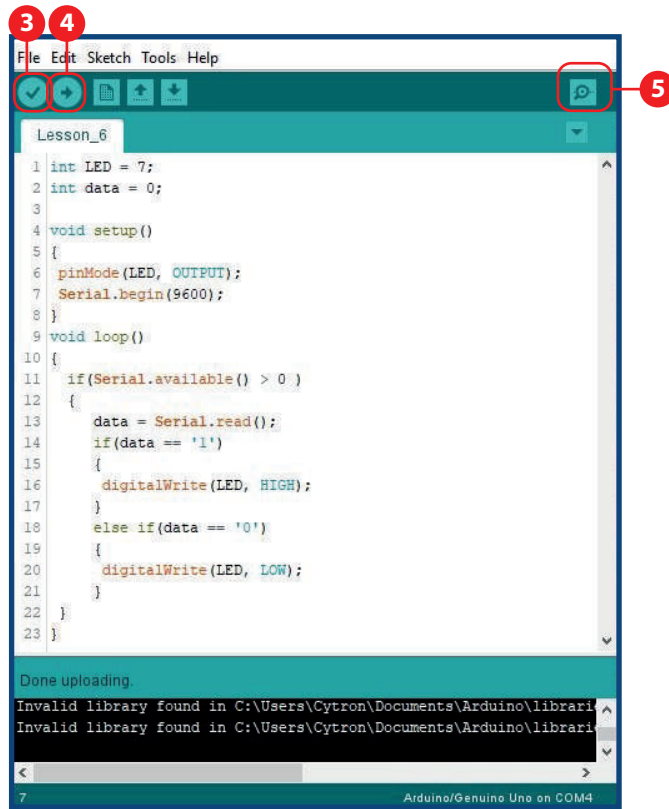
Open new sketch on Arduino IDE.

2

Write this code to your sketch :

```
int LED = 7;
int data = 0;

void setup()
{
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  if(Serial.available() > 0 )
  {
    data = Serial.read();
    if(data == '1')
    {
      digitalWrite(LED, HIGH);
    }
    else if(data == '0')
    {
      digitalWrite(LED, LOW);
    }
  }
}
```

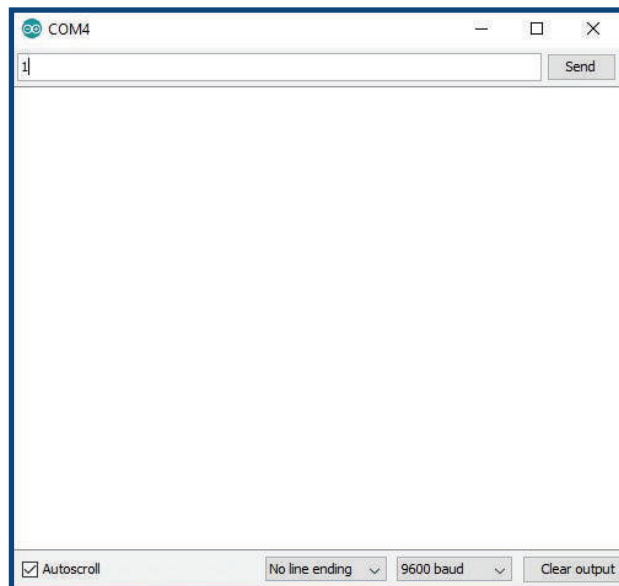



3 Compile the file.

4 Upload the sketch.

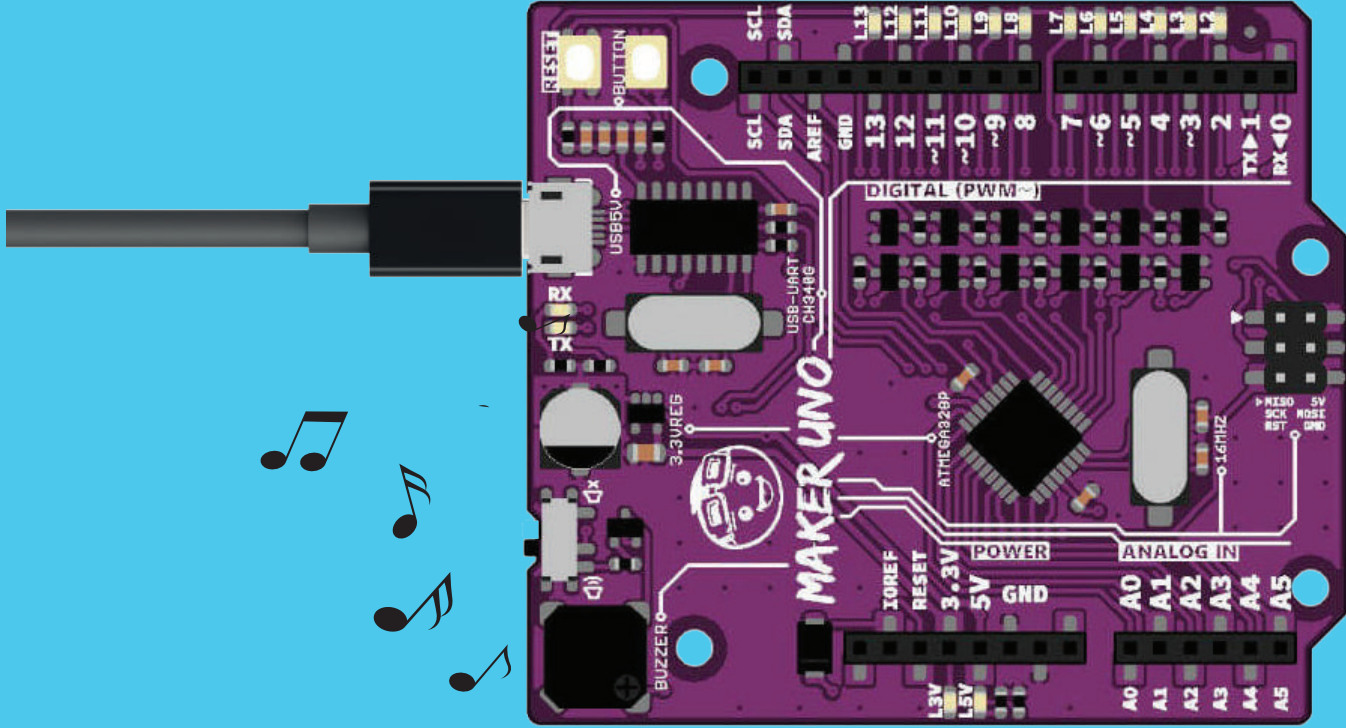
5 You can turn on the LED pin 7 by inserting the letter "1" and turn it off using "0" at the Serial Monitor.

The **Serial.available()** get the number of bytes (characters) available for reading from the serial port.
The **Serial.read()** reads all the incoming data in Maker UNO.



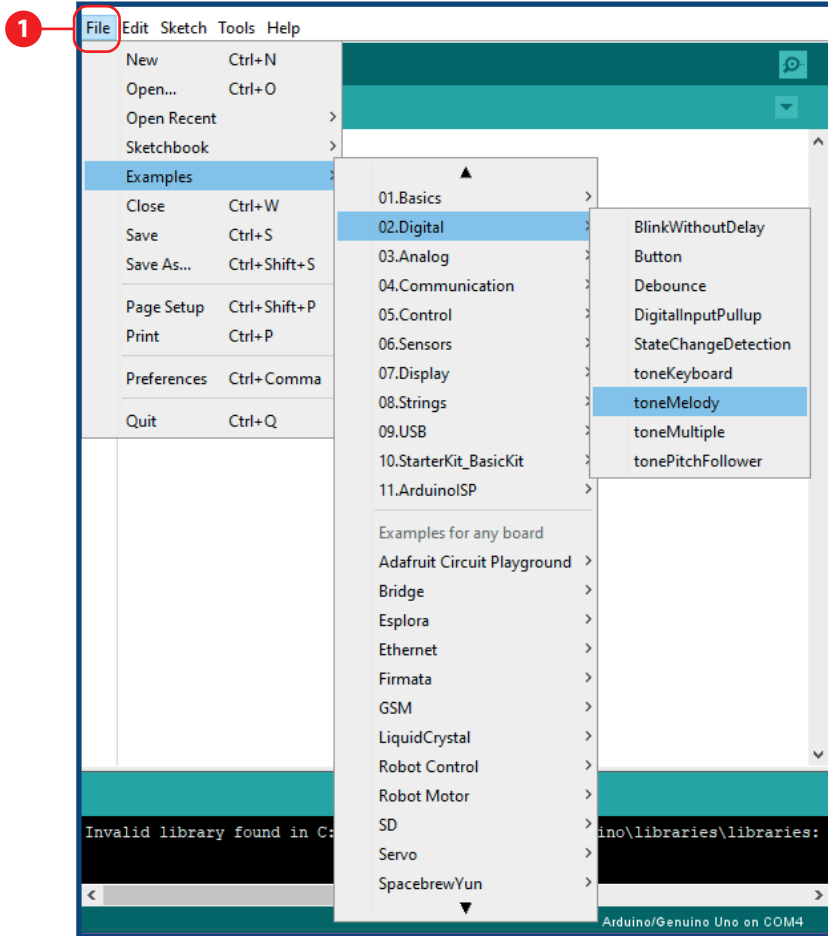


LESSON 7: TONE MELODY





LESSON 7 : TONE MELODY (IDE)



1

Go to File > Examples > 02. Digital > toneMelody

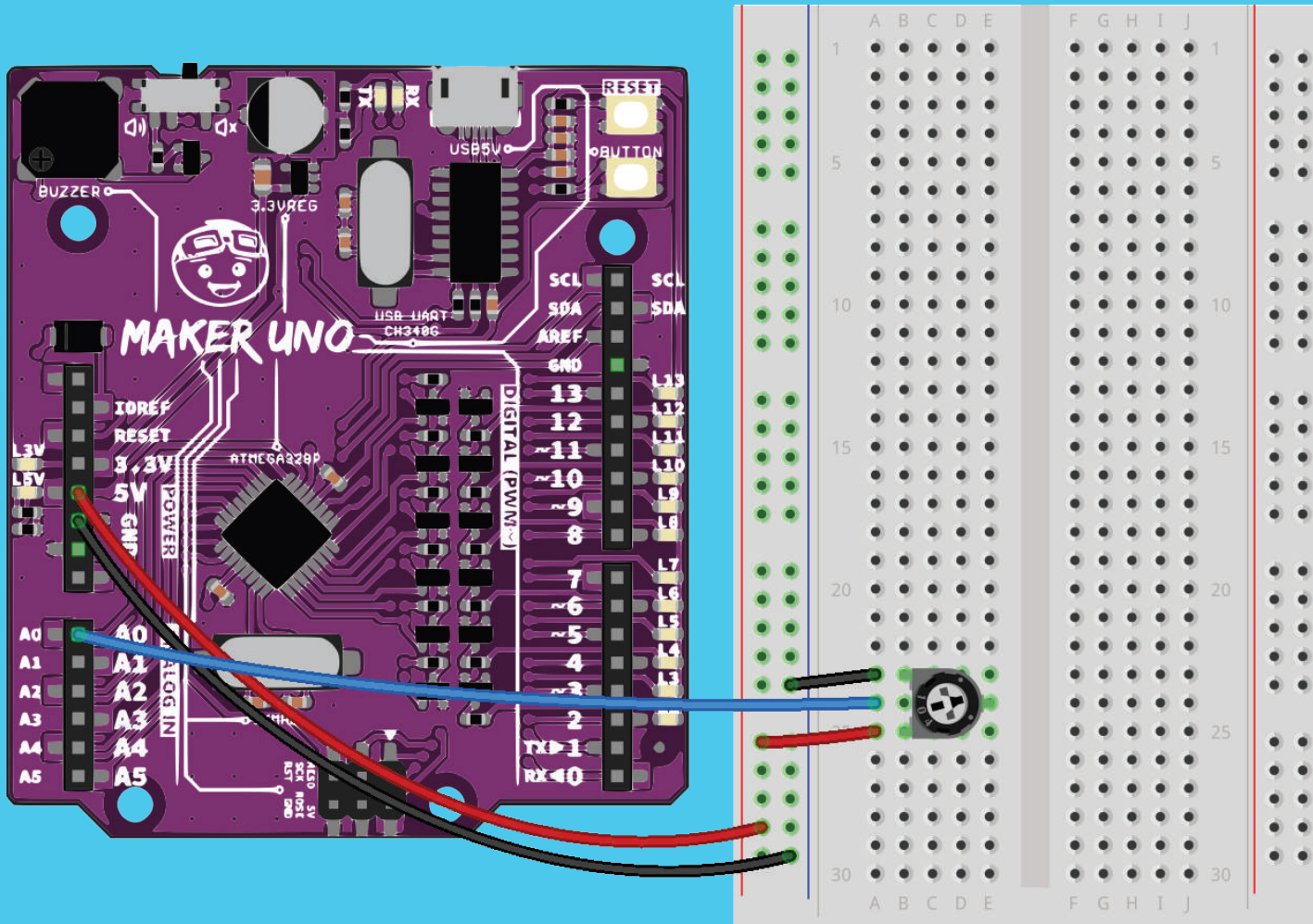
```
19 #include "pitches.h"
20
21 // notes in the melody:
22 int melody[] = {
23   NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
24 };
25
26 // note durations: 4 = quarter note, 8 = eighth note, etc.:
27 int noteDurations[] = {
28   4, 8, 8, 4, 4, 4, 4, 4
29 };
30
31 void setup() {
32   // iterate over the notes of the melody:
33   for (int thisNote = 0; thisNote < 8; thisNote++) {
34
35     // to calculate the note duration, take one second divided
36     //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
37     int noteDuration = 1000 / noteDurations[thisNote];
```

```
4
5 #define NOTE_B0 31
6 #define NOTE_C1 33
7 #define NOTE_CS1 35
8 #define NOTE_D1 37
9 #define NOTE_DS1 39
10 #define NOTE_E1 41
11 #define NOTE_F1 44
12 #define NOTE_FS1 46
13 #define NOTE_G1 49
14 #define NOTE_GS1 52
15 #define NOTE_A1 55
16 #define NOTE_AS1 58
17 #define NOTE_B1 62
18 #define NOTE_C2 65
19 #define NOTE_CS2 69
20 #define NOTE_D2 73
21 #define NOTE_DS2 78
22 #define NOTE_E2 82
23 #define NOTE_F2 87
```

- 2 Compile the file.
- 3 Upload the sketch.
- 4 You can change the music note based on your preference and enjoy the music tone.
- i You can refer to the next tab **pitches.h** for more music note!

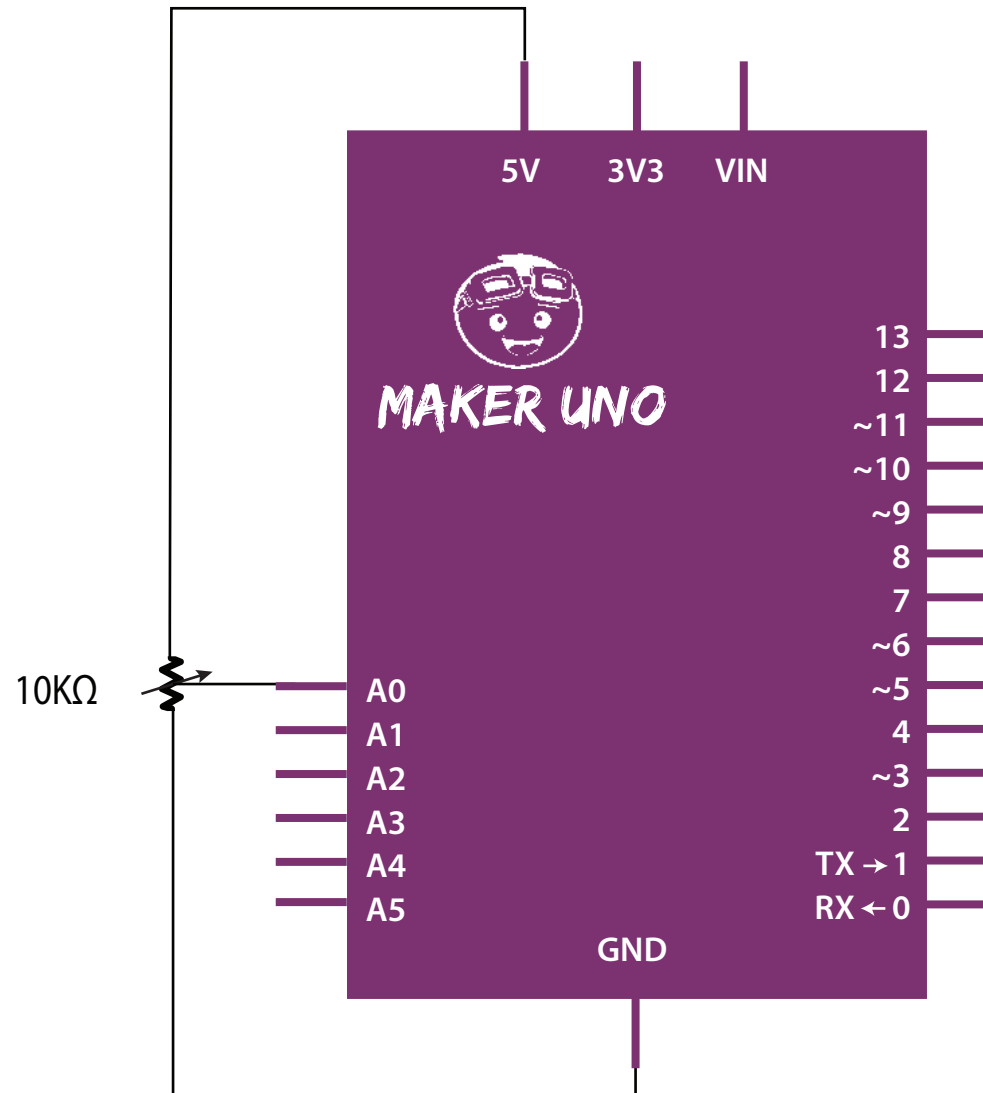


LESSON 8 : POTENTIOMETER AS ANALOG INPUT





LESSON 8 : SCHEMATIC DIAGRAM





LESSON 8 : POTENTIOMETER ANALOG INPUT (IDE)

```
File Edit Sketch Tools Help
Lesson_8
1 int sensorPin = A0;
2 int ledPin = 13;
3 int sensorValue = 0;
4
5 void setup()
6 {
7   pinMode(ledPin,OUTPUT);
8 }
9
10 void loop()
11 {
12   sensorValue = analogRead(sensorPin);
13   digitalWrite(ledPin, HIGH);
14   delay(sensorValue);
15   digitalWrite(ledPin, LOW);
16   delay(sensorValue);
17 }
18
Done uploading.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
18 Maker UNO on COM4
```

1

Open new sketch on Arduino IDE.

2

Write this code to your sketch :

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

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

void loop()
{
  sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
}
```

```
Lesson_8
1 int sensorPin = A0;
2 int ledPin = 13;
3 int sensorValue = 0;
4
5 void setup()
6 {
7   pinMode(ledPin, OUTPUT);
8 }
9
10 void loop()
11 {
12   sensorValue = analogRead(sensorPin);
13   digitalWrite(ledPin, HIGH);
14   delay(sensorValue);
15   digitalWrite(ledPin, LOW);
16   delay(sensorValue);
17 }
18
```

Done uploading.

Invalid library found in C:\Users\Cytron\Documents\Arduino\librar...

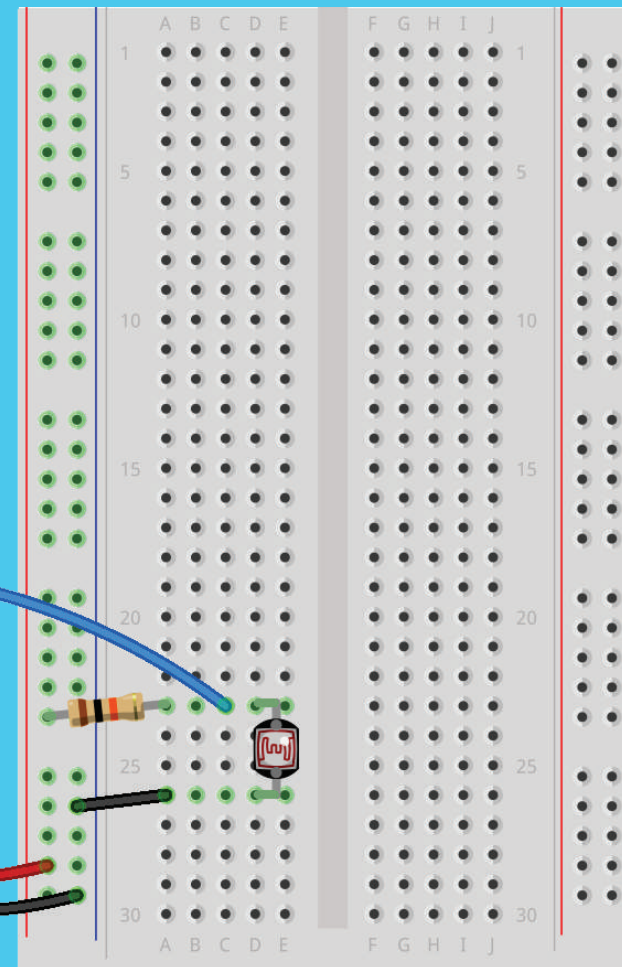
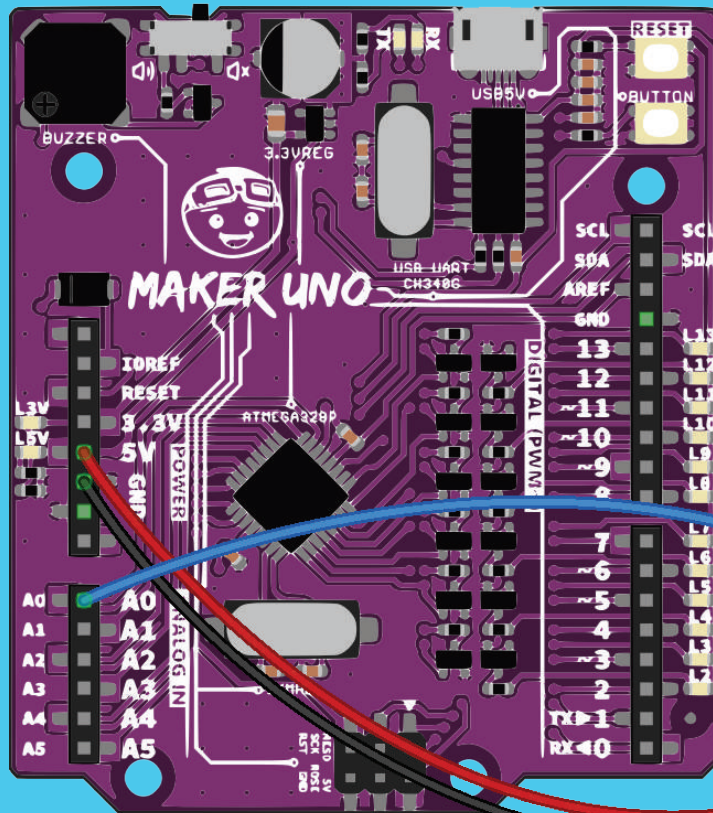
Invalid library found in C:\Users\Cytron\Documents\Arduino\librar...

18 Maker UNO on COM4

- 3 Compile the file.
- 4 Upload the sketch.
- 5 Turn the potentiometer and you will see the blinking speed change.

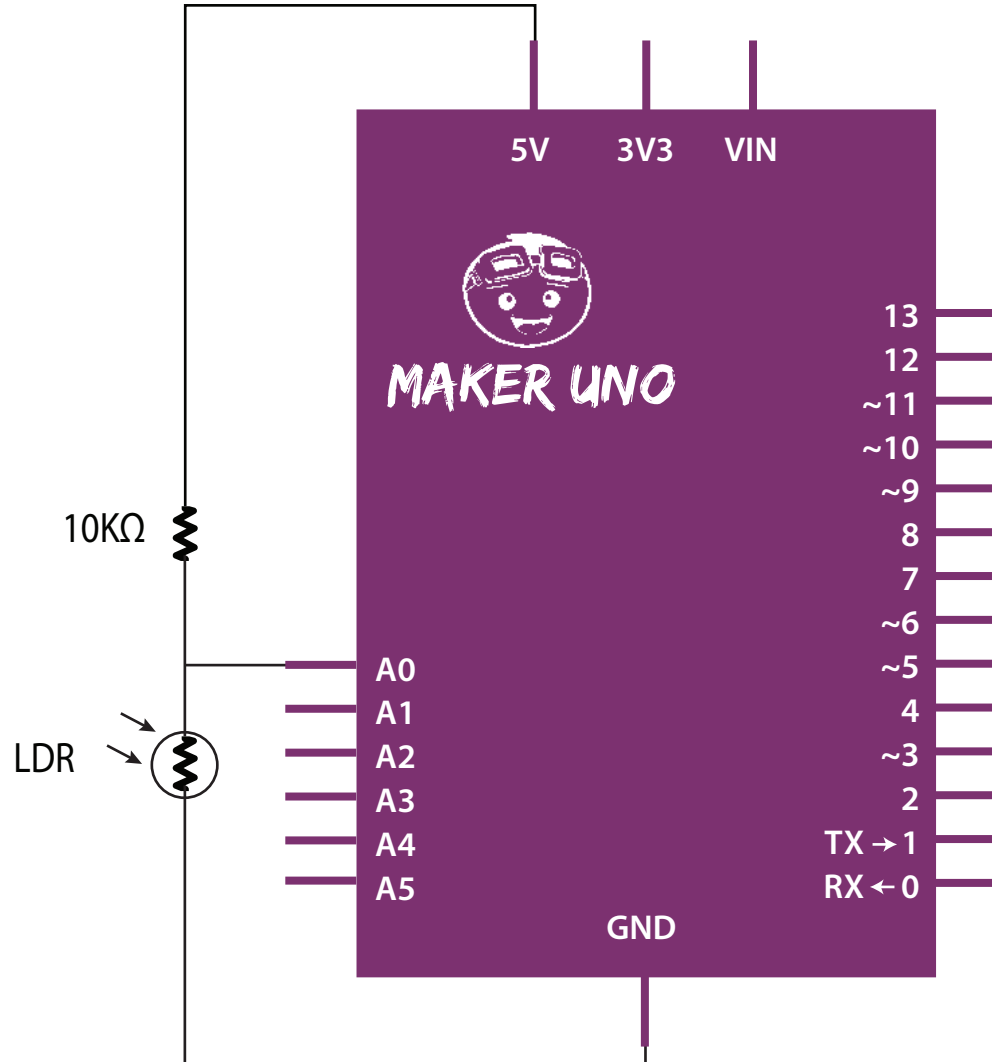


LESSON 9 : LDR AS ANALOG INPUT





LESSON 9 : SCHEMATIC DIAGRAM





LESSON 9 : LDR ANALOG INPUT (IDE)

```
File Edit Sketch Tools Help
Lesson_9
1 int LDR = A0;
2 int ledPin = 13;
3 int LDRvalue = 0;
4
5 void setup()
6 {
7   pinMode(ledPin, OUTPUT);
8 }
9
10 void loop()
11 {
12   LDRvalue = analogRead(LDR);
13   if(LDRvalue > 600)
14     digitalWrite(ledPin, HIGH);
15   else
16     digitalWrite(ledPin, LOW);
17 }
Done uploading.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
7 Maker UNO on COM4
```

1

Open new sketch on Arduino IDE.

2

Write this code to your sketch :

```
int LDR = A0;
int ledPin = 13;
int LDRvalue = 0;

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

void loop()
{
  LDRvalue = analogRead(LDR);
  if(LDRvalue > 600)
    digitalWrite(ledPin, HIGH);
  else
    digitalWrite(ledPin, LOW);
}
```

```
Lesson_9
1 int LDR = A0;
2 int ledPin = 13;
3 int LDRvalue = 0;
4
5 void setup()
6 {
7   pinMode(ledPin, OUTPUT);
8 }
9
10 void loop()
11 {
12   LDRvalue = analogRead(LDR);
13   if(LDRvalue > 600)
14     digitalWrite(ledPin, HIGH);
15   else
16     digitalWrite(ledPin, LOW);
17 }
```

Done uploading.

Invalid library found in C:\Users\Cytron\Documents\Arduino\librar...

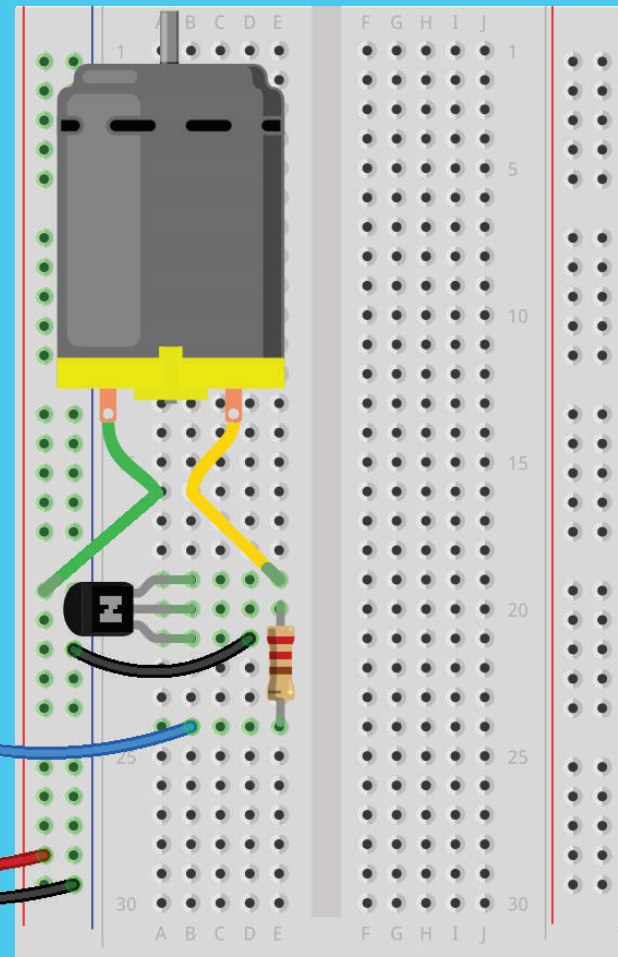
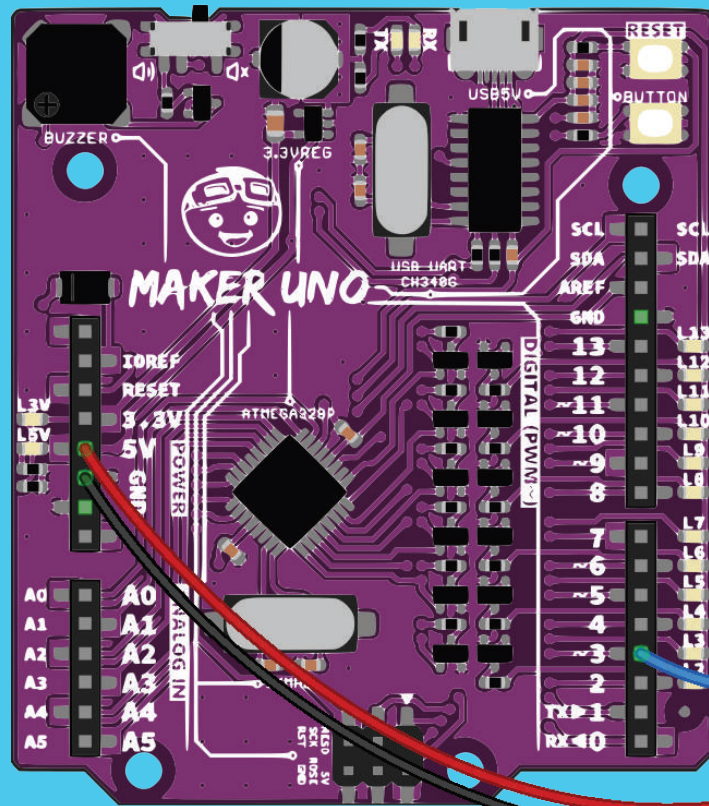
Invalid library found in C:\Users\Cytron\Documents\Arduino\librar...

7 Maker UNO on COM4

- 3 Compile the file.
- 4 Upload the sketch.
- 5 When it is dark, the LED on pin 13 will light up.

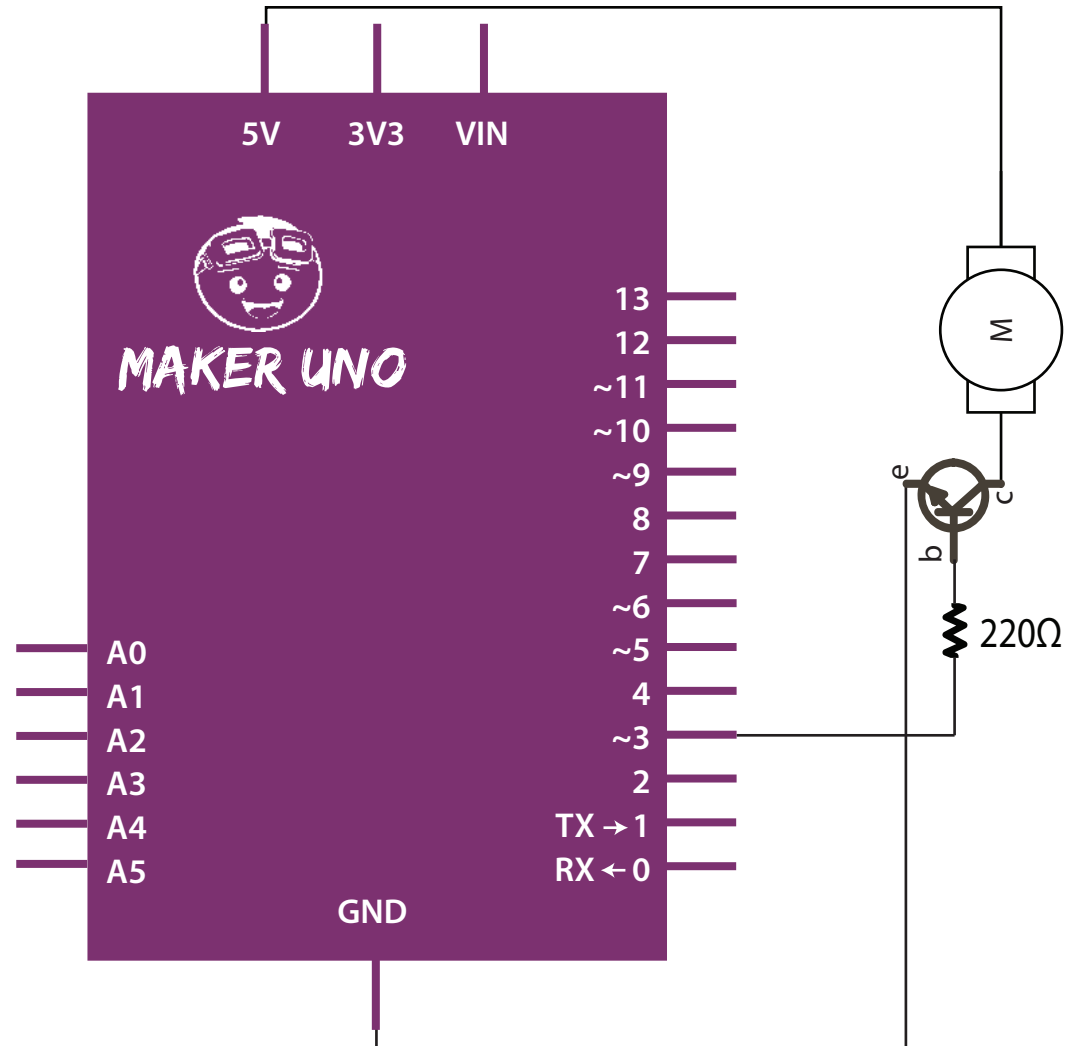


LESSON 10 : CONTROLLING MOTOR





LESSON 10 : SCHEMATIC DIAGRAM





LESSON 10 : CONTROLLING MOTOR (IDE)

```
File Edit Sketch Tools Help
Lesson_10
1 void setup()
2 {
3   pinMode(6,OUTPUT);
4 }
5
6 void loop()
7 {
8   analogWrite(6,255); // same with HIGH
9   delay(1000);
10  analogWrite(6,123);
11  delay(1000);
12  analogWrite(6,50);
13  delay(1000);
14  analogWrite(6,LOW); // same with 0
15  delay(1000);
16 }
Done uploading.
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries
8 Maker UNO on COM4
```

1

Open new sketch on Arduino IDE.

2

Write this code to your sketch :

```
void setup()
{
  pinMode(6,OUTPUT);
}

void loop()
{
  analogWrite(6,255); //same with HIGH
  delay(1000);
  analogWrite(6,123);
  delay(1000);
  analogWrite(6,50);
  delay(1000);
  analogWrite(6, LOW); //same with 0
  delay(1000);
}
```

```
1 void setup()
2 {
3   pinMode(6, OUTPUT);
4 }
5
6 void loop()
7 {
8   analogWrite(6, 255); // same with HIGH
9   delay(1000);
10  analogWrite(6, 123);
11  delay(1000);
12  analogWrite(6, 50);
13  delay(1000);
14  analogWrite(6, LOW); // same with 0
15  delay(1000);
16 }
```

Done uploading.

Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries

Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries

8 Maker UNO on COM4

- 3 Compile the file.
- 4 Upload the sketch.
- 5 The motor will rotate with 4 different speed.



PROJECT

PROJECT 1

INTERACTIVE TRAFFIC LIGHT

INTRODUCTION

Interactive Traffic Light is a combination of standard traffic light for vehicles and traffic light for pedestrian.

This project applies knowledge outcome from:

Lesson 1: Light Up LED

Lesson 4: Push Button as Digital Input

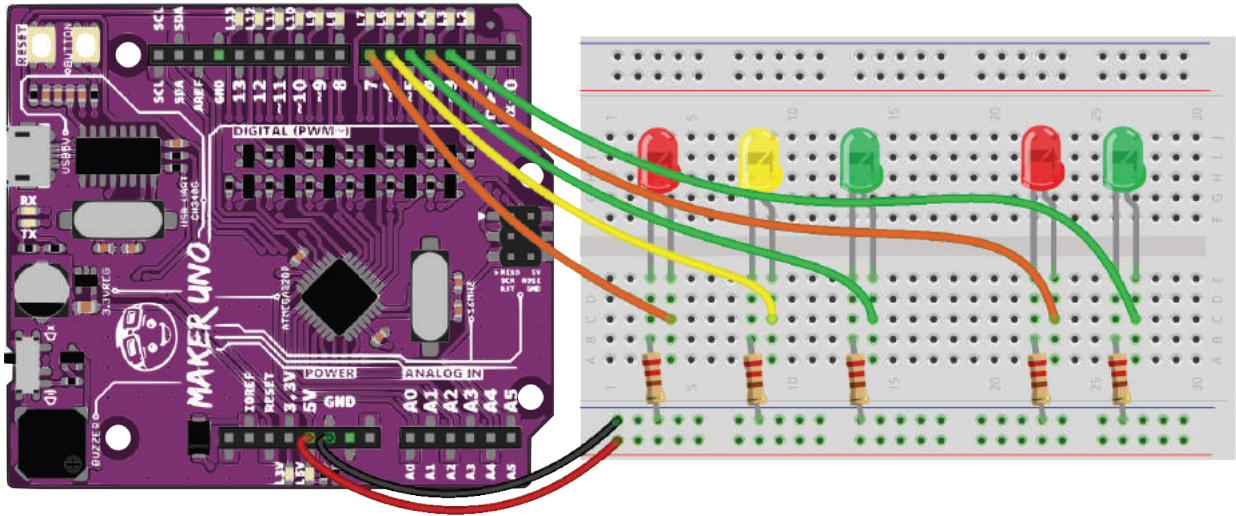
INGREDIENTS

- a. Maker UNO - 1x
- b. Breadbord - 1x
- c. Red LED - 2x
- d. Green LED - 2x
- e. Yellow LED - 1x
- f. Resistor 220Ω - 5x
- g. Jumper wires

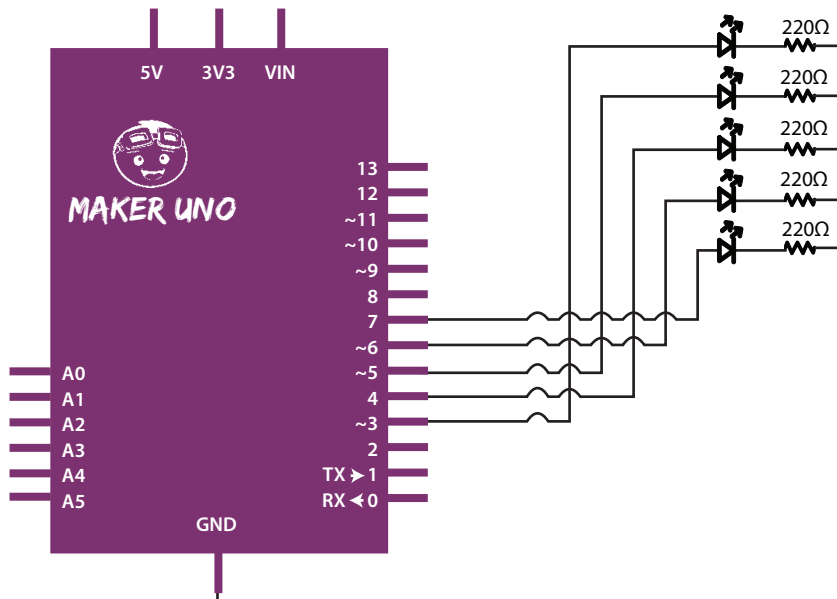
INSTRUCTION

By using all the parts above, create a simple traffic light system for a pedestrian crossing. Normally, the traffic light is green. But when the push button is pressed, the light will switch to yellow for two seconds, then to red. After 1 more second, the green pedestrian light will light up for 5 seconds, then turns back to red. After 1 more second, the traffic light turns green again.

HARDWARE CONNECTION



SCHEMATIC DIAGRAM



ARDUINO CODE

```
const int greenLedVehicle = 5;
const int yellowLedVehicle = 6;
const int redLedVehicle = 7;
const int greenLedPedestrian = 3;
const int redLedPedestrian = 4;
const int pushButton = 2;

void setup()
{
  pinMode(greenLedVehicle, OUTPUT);
  pinMode(yellowLedVehicle, OUTPUT);
  pinMode(redLedVehicle, OUTPUT);
  pinMode(greenLedPedestrian, OUTPUT);
  pinMode(redLedPedestrian, OUTPUT);
  pinMode(pushButton, INPUT_PULLUP);

  digitalWrite(greenLedVehicle, HIGH);
  digitalWrite(redLedPedestrian, HIGH);
}

void loop()
{
  if(digitalRead(pushButton) == LOW)
  {
    digitalWrite(greenLedVehicle, LOW);
    digitalWrite(yellowLedVehicle, HIGH);
    delay(2000);
    digitalWrite(yellowLedVehicle, LOW);
    digitalWrite(redLedVehicle, HIGH);
    delay(1000);
    digitalWrite(redLedPedestrian, LOW);
    digitalWrite(greenLedPedestrian, HIGH);
    delay(5000);
    digitalWrite(greenLedPedestrian, LOW);
    digitalWrite(redLedPedestrian, HIGH);
    delay(1000);
    digitalWrite(redLedVehicle, LOW);
    digitalWrite(greenLedVehicle, HIGH);
  }
}
```

PROJECT 2

LIGHT THEREMIN

INTRODUCTION

A theremin is an instrument that makes sounds based on the movements of a musician's hands around the instrument. This project will use LDR as an input where the amount of light intensity will determine the melody notes.

This project applies knowledge outcome from:

Lesson 3: Create Melody with Piezo

Lesson 8: Light Dependent Resistor

INGREDIENTS

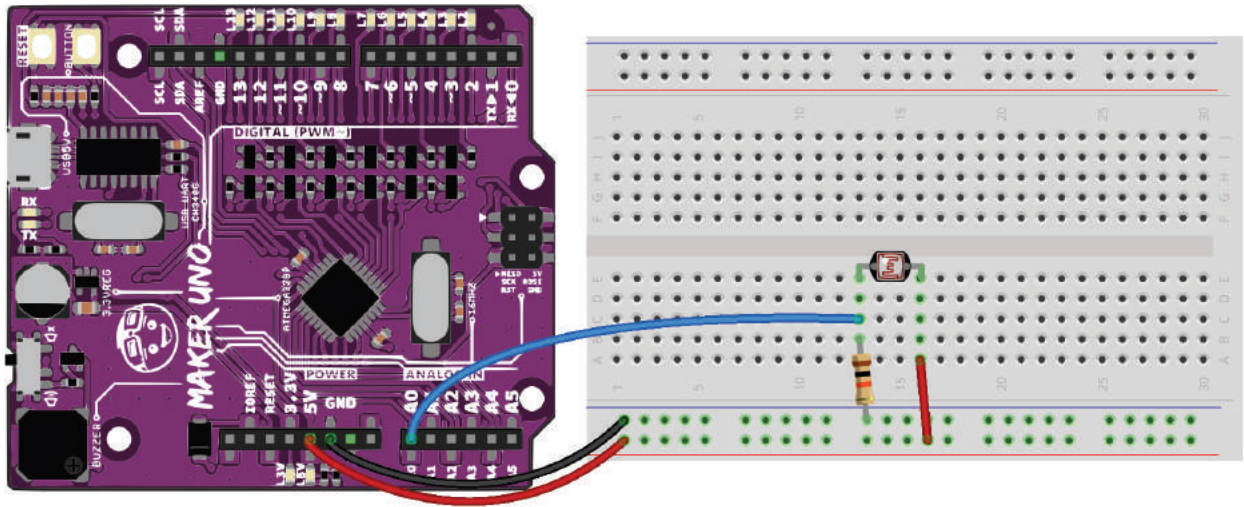
- a. Maker UNO - 1x
- b. Breadboard - 1x
- c. Resistor 10k Ω - 1x
- d. LDR - 1x
- e. Jumper wires

INSTRUCTION

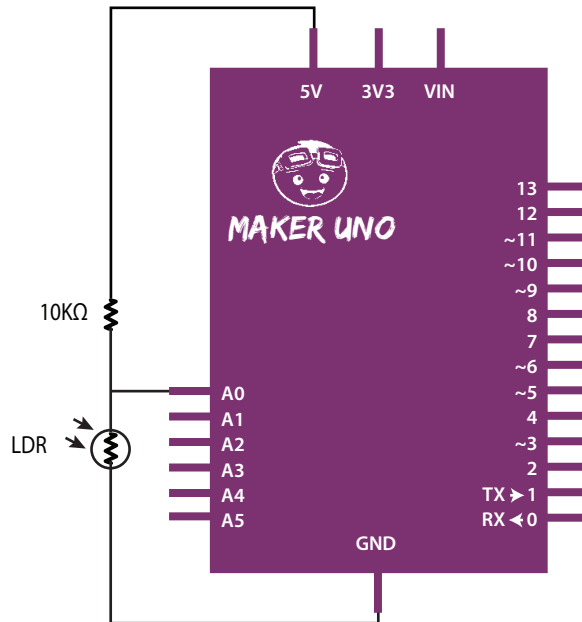
Using all the parts above create an instrument that creates melody played by piezo depends on your hand position. The closer your hand is to the LDR, the higher the notes produced. When you withdraw your hand, no sound will be generated. So, enjoy the melody you create!

Note: To calibrate the sensor, move your hand up and down over the LDR for 5 seconds to change the amount of light that reaches it. The closer you replicate the motions you expect to use while playing the instrument, the better the calibration will be.

HARDWARE CONNECTION



SCHEMATIC DIAGRAM



ARDUINO CODE

```
#include "pitches.h"
int melody[49] = {
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  NOTE_C2, NOTE_D2, NOTE_E2, NOTE_F2, NOTE_G2, NOTE_A2, NOTE_B2,
  NOTE_C3, NOTE_D3, NOTE_E3, NOTE_F3, NOTE_G3, NOTE_A3, NOTE_B3,
  NOTE_C4, NOTE_D4, NOTE_E4, NOTE_F4, NOTE_G4, NOTE_A4, NOTE_B4,
  NOTE_C5, NOTE_D5, NOTE_E5, NOTE_F5, NOTE_G5, NOTE_A5, NOTE_B5,
  NOTE_C6, NOTE_D6, NOTE_E6, NOTE_F6, NOTE_G6, NOTE_A6, NOTE_B6
};
int sensorValue = 0;
int sensorLow = 1023;
int sensorHigh = 0;
const int ledPin = 13;

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

  // Calibrate for the first five seconds after program runs
  while(millis() < 5000)
  {
    sensorValue = analogRead(A0);
    if(sensorValue > sensorHigh)
      sensorHigh = sensorValue;
    if(sensorValue < sensorLow)
      sensorLow = sensorValue;
  }
  digitalWrite(ledPin, LOW);
}

void loop()
{
  sensorValue = analogRead(A0);
  int pitch = map(sensorValue, sensorLow, sensorHigh, 48, 0);
  tone(8, melody[pitch], 50);
  delay(50);
  noTone(8);
  delay(150);
}
```

Prepared by:
Cytron Technologies Sdn Bhd
www.cytron.io
No. 1, Lorong Industri Impian 1,
Taman Industri Impian,
14000 Bukit Mertajam,
Penang, Malaysia.

Tel: +604 - 548 0668
Fax: +604 - 548 0669

Email:
support@cytron.io
sales@cytron.io