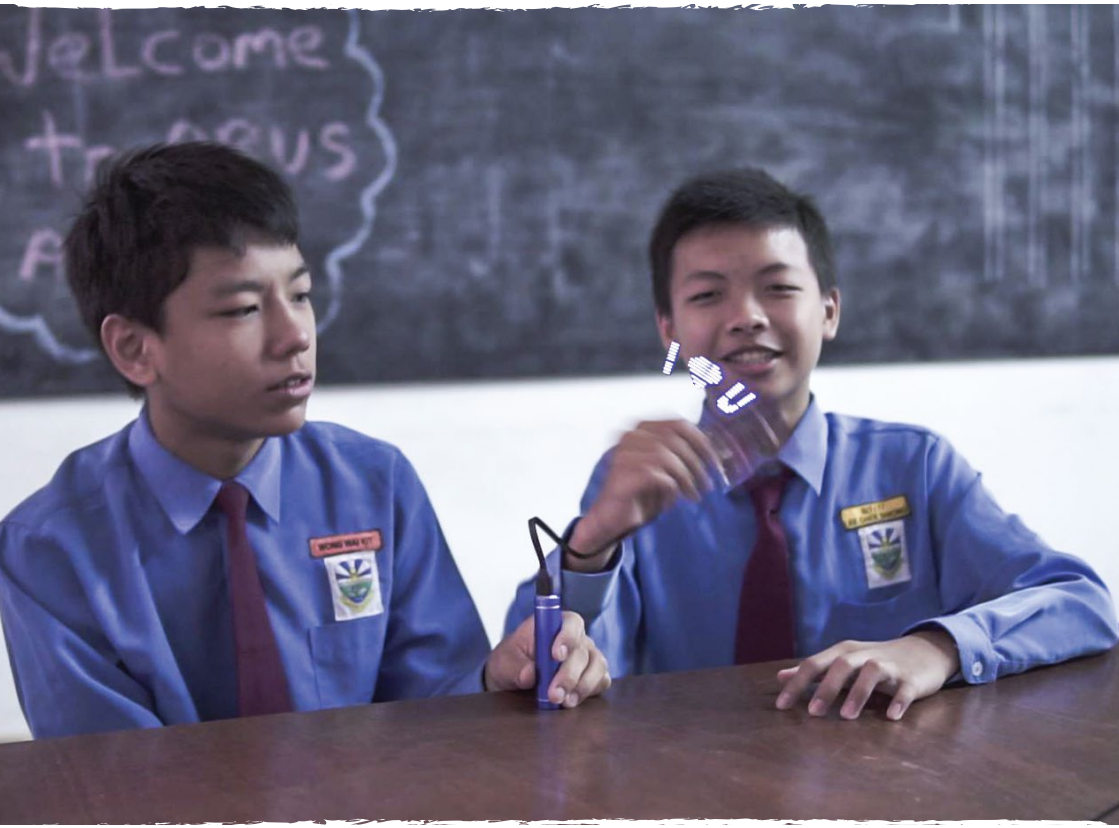
Update: Ver 1.3 Dec 2018



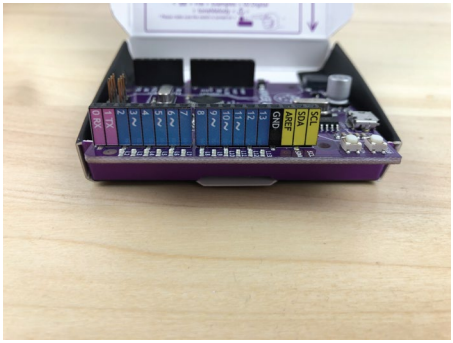
Arduino Learning Guide For Beginner Using MAKER UNO

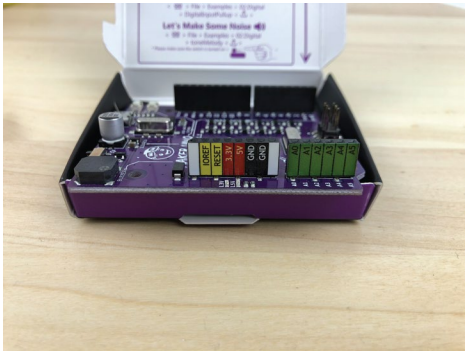
LESSON 0
SETTING UP HARDWARE & SOFTWARE

Part 1: Put Up Label Stickers for Pin Headers

1. Remove the Maker UNO from the packaging and you will find a set of label sticker.

2. Peel off the stickers and attach them on the pin headers accordingly.



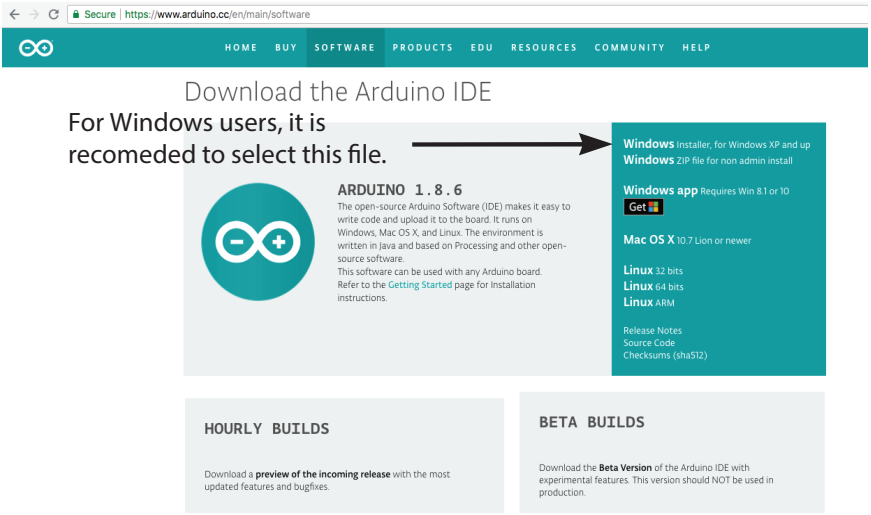


3. Connect Maker UNO to your PC with a micro USB cable.

Part 2: Download & Install Arduino IDE

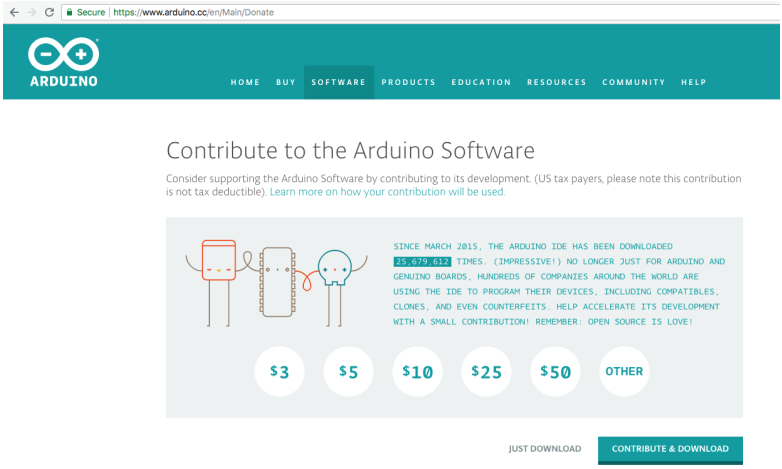
1. Log on to https://www.arduino.cc/en/main/software.

2. Choose your OS to proceed.



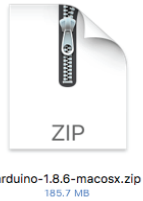
3. Arduino IDE is an open source software that allows you to download and use it for free.

   However, you are encouraged to make a monetary contribution to help them to continue to fund their development. Anyway, you are free to click "JUST DOWNLOAD".



4. Double click on the downloaded file to proceed.

For Window



For Mac

5. Once installation is completed, the Arduino's icon will appear. Double click the icon to launch the Arduino IDE.

Part 3: Download & Install Driver

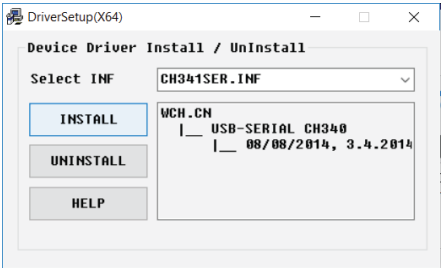
For Window Users:

1. Download the driver here:
   https://cdn.cytron.io/makeruno/CH341SER.EXE

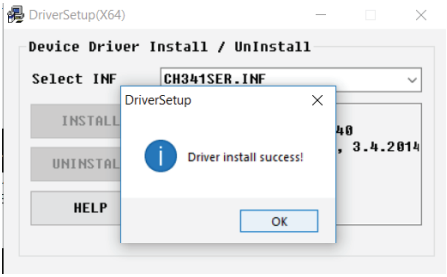
2. Double click the “CH341SER” file to begin installation.

CH341SER

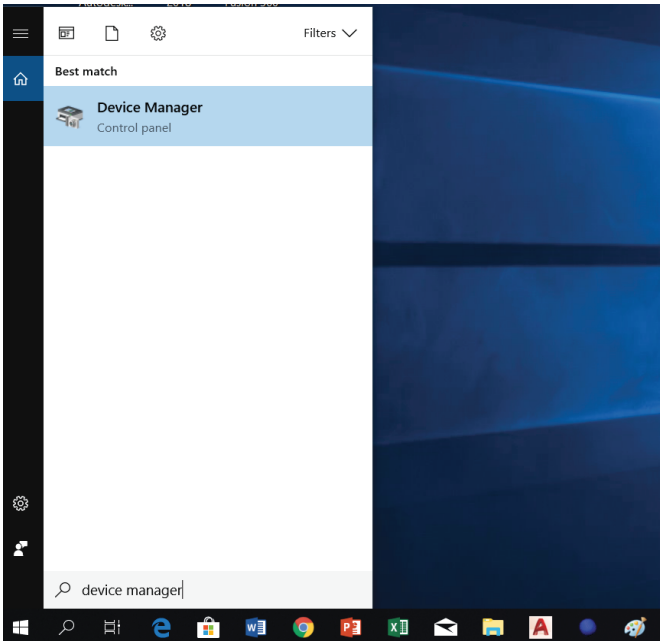
3. Click “INSTALL”.



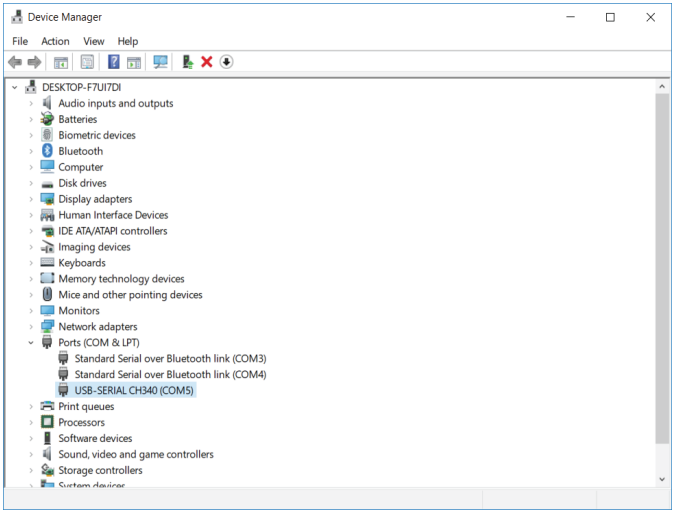
4. Click “OK”.



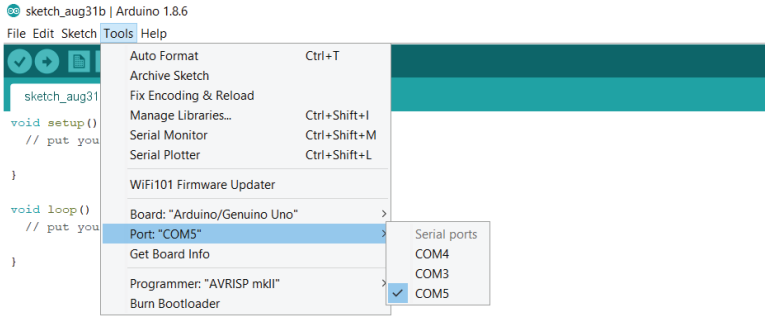
5. Go to window search, seaarch for “device manager”.



6. Click › to expand “Ports (COM & LPT)”. Check which port the CH340 driver is being assigned to. Remember the com number. (For this example, the com number is com 5)



7. Launch Arduino IDE. Select the right com port. Tools >Ports >COM X (Make sure your Maker UNO is connected to your PC)



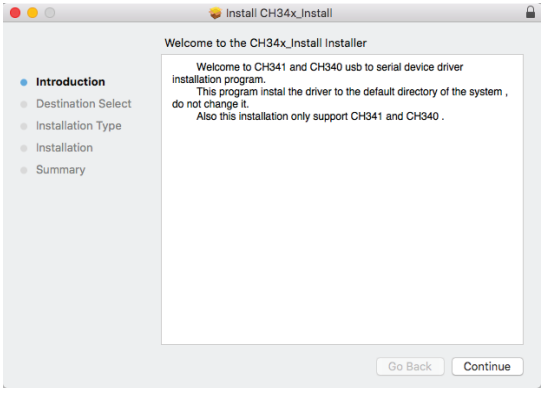
For Mac users:

1. Download the driver here:
   https://cdn.cytron.io/makeruno/CH341SER_MAC.ZIP

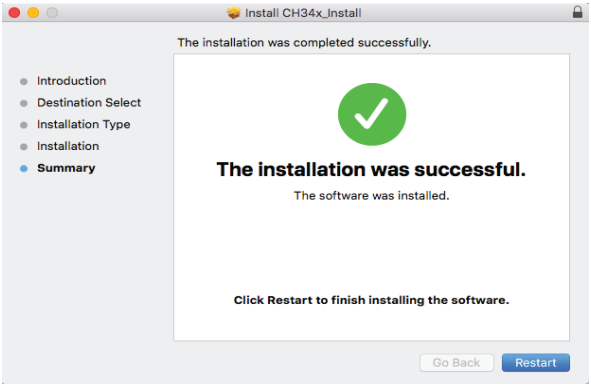
2. Double click the zip file, open the unzip folder then double click the pkg file.

CH34x_Install_V1.
4.pkg

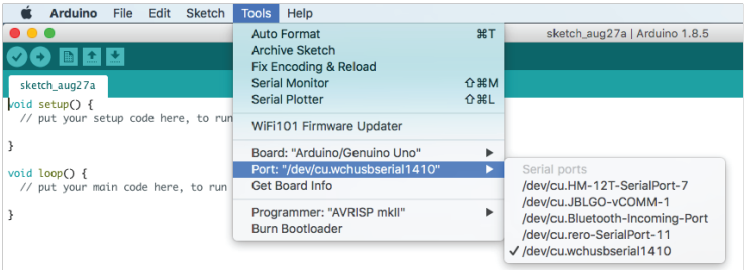
3. Click “Continue” to begin installation.



4. Once done, click “Restart” to restart your Mac.



5. After you have restarted your Mac, launch Arduino IDE again. Choose the driver in Tools > Port > /dev/cu.wchusbserial1410



Note: if you are unable to locate /dev/cu.wchusbseial1410, please follow the troubleshooting steps:
https://cdn.cytron.io/makeruno/Troubleshooting_CH431_Driver_For_Mac.pdf.

Congratulation! You have successfully setup Maker UNO and we will start making something in the next lesson.

LESSON 1
DIGITAL OUTPUT

Project 1: Turn On An LED

1. Connect your Maker UNO with your PC with a USB cable.

2. Launch Arduino IDE.

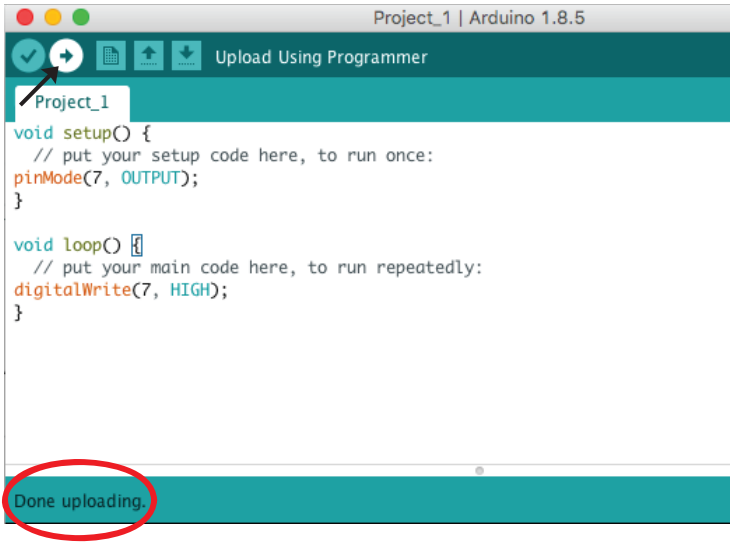
3. Write these code into your sketch.

4. Click compile , wait for a few seconds until you see “Done Compiling” appear at the bottom of the sketch.

Troubleshooting

1. If error occurs, go through your code line by line to make sure they are correct and then compile again.
2. Don’t mixed up the normal backet ( ) and the curly bracket { }.

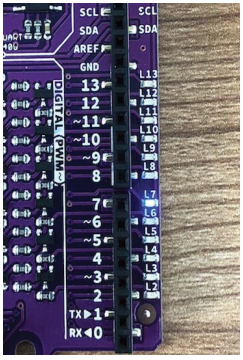
5. Then click upload  . Wait for a few seconds and you will see "Done Uploading" appear at the bottom of the sketch.



Troubleshooting

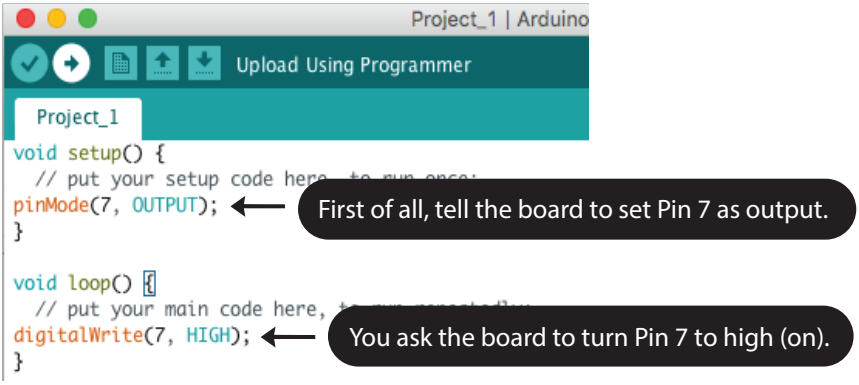
1. If error occurs, please ensure that your Maker UNO is connected to your PC and select the correct port at Arduino IDE again. "Tools > Port"

6. Check your result.



You will see LED 7 is on while the rest of the LEDs are off.

How it works



Good To Know

Basic Electronics

Input Devices: Switch, sensor, microphone, potentiometer, etc.

Output devices: LED, LCD screen, buzzer, speaker, motor, etc.

Arduino Function

1. To set a pin as input or output:

   pinMode(pin, mode);

   pin: the number of the pin whose mode you wish to set
   mode: INPUT, OUTPUT or INPUT_PULLUP

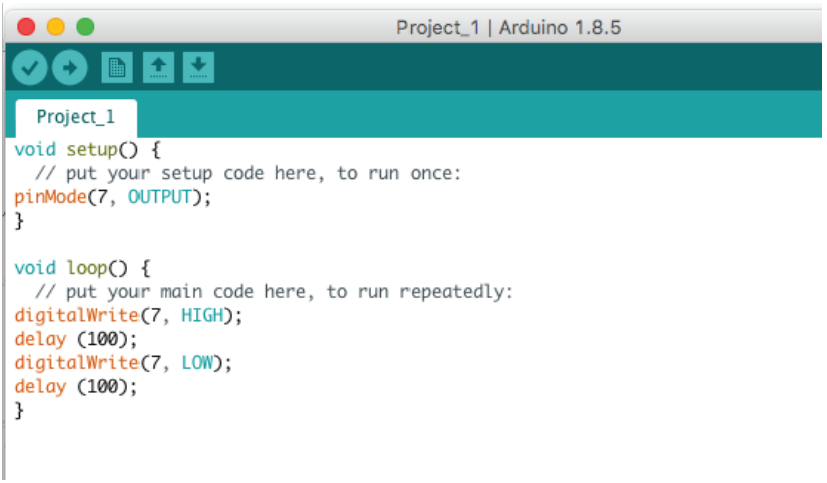
2. To turn a digital pin HIGH (on) or LOW (off):

   digitalWrite(pin, value);

   pin: the pin number
   value : HIGH or LOW

Project 2: Blink An LED

1. Modify your previous codes into this.



2. Click compile  , wait for a few seconds until you see “Done Compiling” appear at the bottom of the sketch.

3. Then click upload  . Wait for a few seconds and you will see “Done uploading” appear at the bottom of the sketch.

4. Check your result.

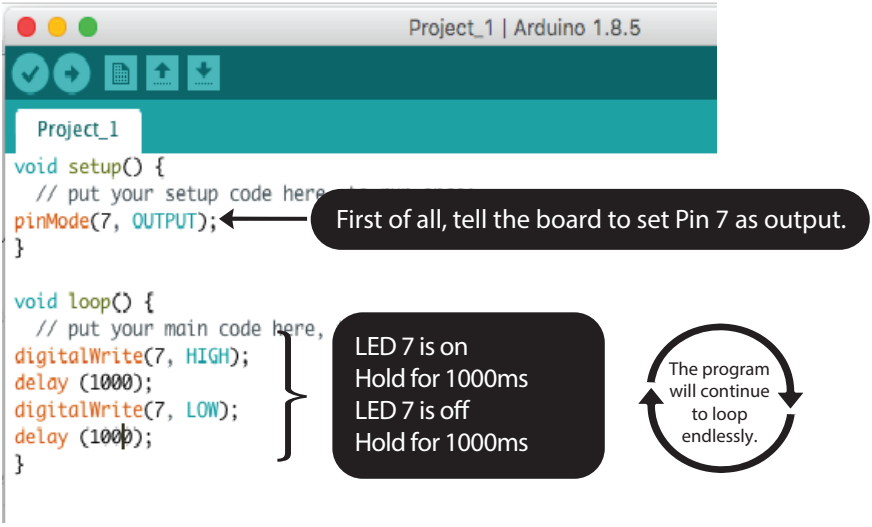
Click here or scan QR code to watch the demo video

LED 7 will be blinking while the rest of the LEDs are off.

5. Change the delay value to 1000 then upload it to your board again.

6. Observe the result and compare with the previous program. Can you tell the difference?

How it works



Good To Know

Arduino Function

1. To pause or hold the program:

   delay(ms);

   ms: the number of miliseconds to pause

Project 3: Blink the LEDs in Sequence

In this project, we want to blink LED 2, LED 3, LED 4, LED 5, LED 6 and LED 7 in sequence.

1. Modify your previous code into this.


```
void setup() {
  // put your setup code here, to run once:
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
delay(1000);
digitalWrite(2,LOW);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
delay(1000);
}
```

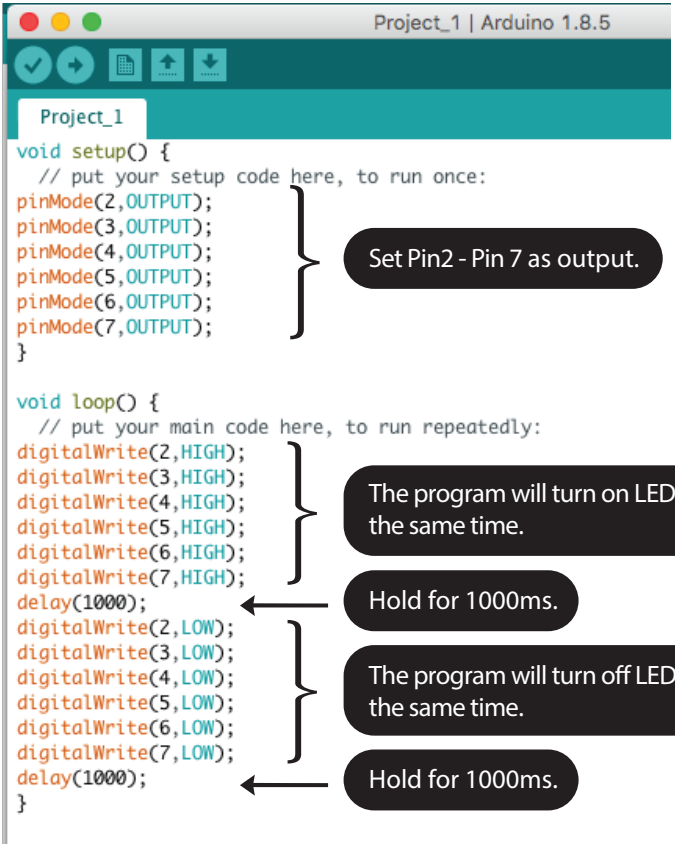
2. Click compile and then upload .

3. Check your result.

Click here or scan QR code to watch the demo video

LED 2- LED 7 go on and off together which is not the consequence that we want. Why?

How it works



4. Modify your previous code into this.

Project_1

```
void setup() {
  // put your setup code here, to run once:
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(2,HIGH);
delay(1000);
digitalWrite(3,HIGH);
delay(1000);
digitalWrite(4,HIGH);
delay(1000);
digitalWrite(5,HIGH);
delay(1000);
digitalWrite(6,HIGH);
delay(1000);
digitalWrite(7,HIGH);
delay(1000);
digitalWrite(2,LOW);
delay(1000);
digitalWrite(3,LOW);
delay(1000);
digitalWrite(4,LOW);
delay(1000);
digitalWrite(5,LOW);
delay(1000);
digitalWrite(6,LOW);
delay(1000);
digitalWrite(7,LOW);
delay(1000);
}
```

5. Check your result.

Click here or scan QR code to watch the demo video

The LED will go on one by one then off one by one.

Good To Know

```
void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(2,HIGH);
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
delay(1000);
digitalWrite(2,LOW);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
delay(1000);
}
```

The program runs line by line super fast (in less than 1 micro second) until all lines are executed almost at the same time. That's why you will see all LEDs go on or off at the same time.

```
void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(2,HIGH);
delay(1000);
digitalWrite(3,HIGH);
delay(1000);
digitalWrite(4,HIGH);
delay(1000);
digitalWrite(5,HIGH);
delay(1000);
digitalWrite(6,HIGH);
delay(1000);
digitalWrite(7,HIGH);
delay(1000);
```

To fix that problem, we need to add a delay in between each LED.

Challenge

Q: Program LED 2 - LED 13 to execute the blinking pattern as shown at the demo video below.

Click here or scan QR code to watch the demo video

Congratulation! You have completed lesson 1and learnt the following:

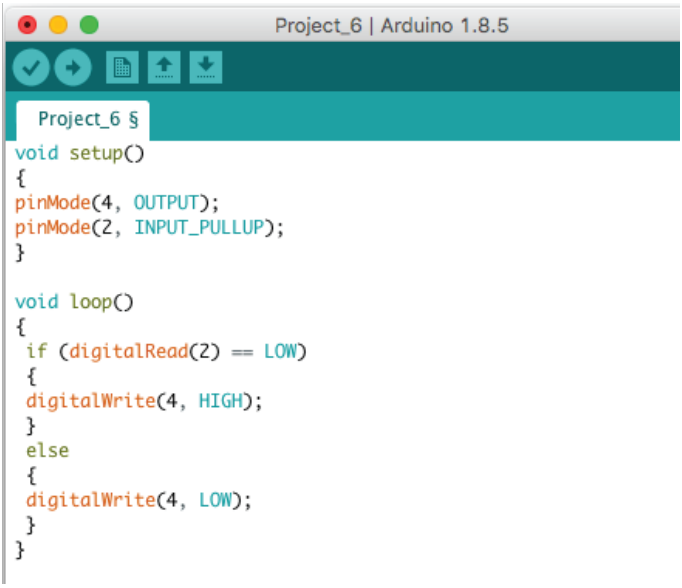
1. How to set pin as digital output.
2. How to on and off a digital output pin.
3. How to use delay function.

LESSON 2
DIGITAL INPUT

Project 4: On-Board Push Button Switch

In this project, we want to control an LED using the on-board push button switch.

1. Open a new sketch then write these codes into the sketch.




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

void loop()
{
 if (digitalRead(2) == LOW)
 {
 digitalWrite(4, HIGH);
 }
 else
 {
 digitalWrite(4, LOW);
 }
}
```

2. Compile and upload the program.

3. Check your result.

Click here or scan QR code to watch demo video

LED 4 will light up when the push button is pressed.

How it works

Project_6 | Arduino 1.8.5

Project_6 §

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

void loop()
{
 if (digitalRead(2) == LOW)
 {
 digitalWrite(4, HIGH);
 }
 else
 {
 digitalWrite(4, LOW);
 }
}
```

Set Pin 4 as output.
Set Pin 2 as on-board switch input.

Read Pin 2. If it is low (switch is pressed)
Set Pin 4 to high.

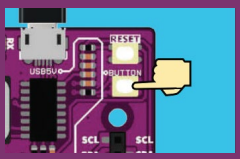
Else. (switch is not pressed)
Set Pin 4 to low.

Good To Know

Arduino Function

1. To use the on-board push button switch, we need to set it as "internal pullup input".

pinMode(2, INPUT_PULLUP);



2. The on-board LED at Pin 2 will act as an input indicator. It will turn off if the on-board switch is pressed.

3. The on-board switch is internally connected to Pin 2. Meaning it is occupied and cannot be connected to any other external components anymore if you would like to use it.

Coding Syntax

1. To use “if-else” statement.

```
if (condition 1)
{
// do thing A
}
else if (condition 2)
{
// do thing B
}
else
{
// do thing C
}
```

2. We can put a sign shown below, if you want to leave a reminder or comment for yourself while programming. Anything written behind this sign will be ignored by the program and will not be executed.

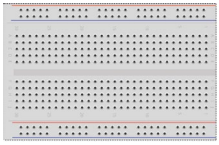
```
// your own comments here
```

Project 5: External Push Button Switch

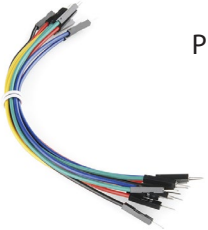
In this project, we want to construct a basic circuit of an external push button switch.

1. Get these components.

Push Button Switch



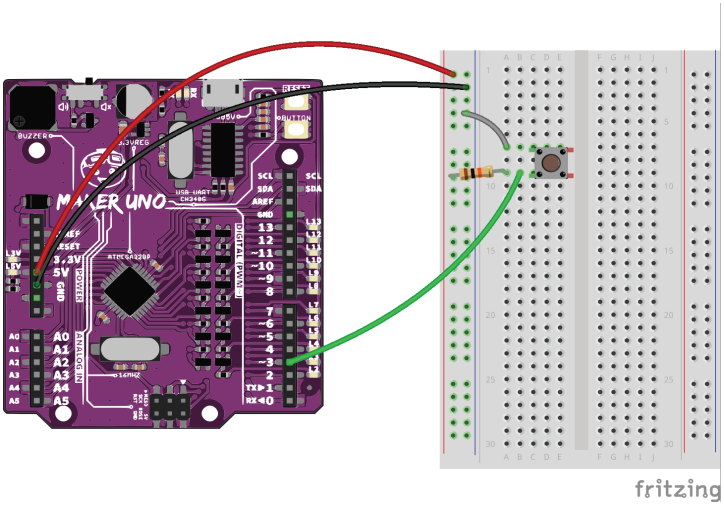
Breadboard



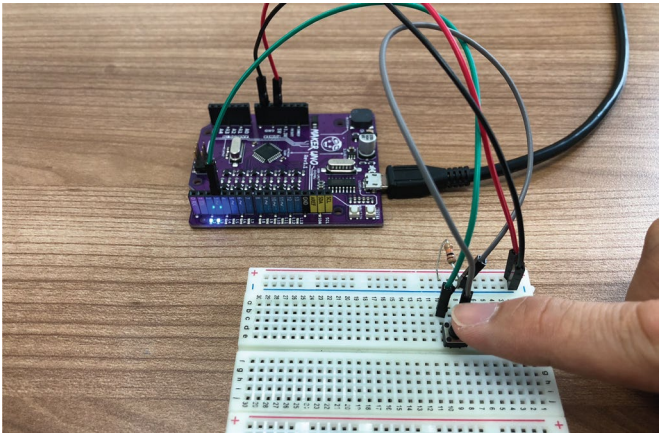
Jumper Wire

10K ohm resistor

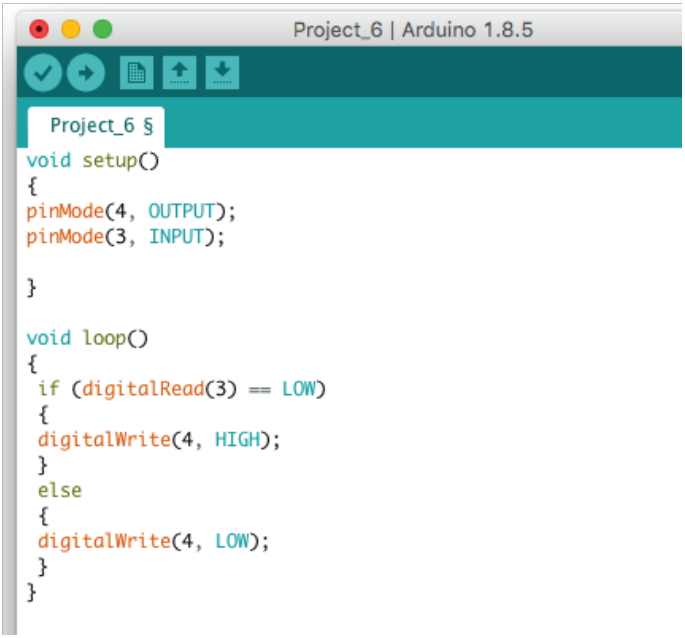
2. Construct the circuit as shown below.



3. Press the external push button switch. Observe LED 3, it is on before you press the switch and goes off when the switch is pressed. If it is not working, please fix your circuit before you proceed to the next step.



4. Modify your previous code into this, then upload to your board.

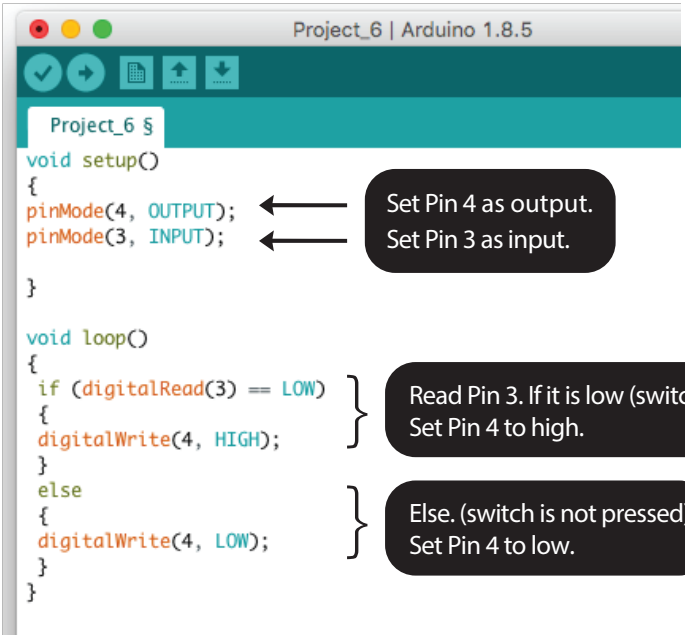


5. Check your result.

Click here or scan QR code to watch the demo video

When the external push button switch is pressed, LED 4 will be on.

How it works



Good To Know

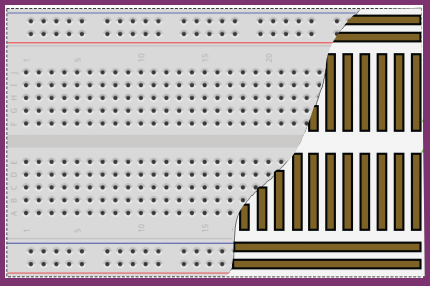
Arduino Function

1. Switch is an input, you need to set that pin as input before you can use it.

```
pinMode(pin, INPUT);
```

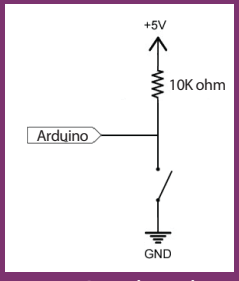
Basic Electronics

1. Breadboard internal connectivity.



2. For any digital input, you can make it as “pull-up” or “pull-down” circuit.

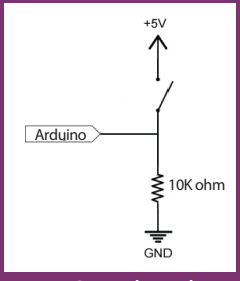
Pull-up resistor



It is HIGH when the switch is not pressed

VS

Pull-down resistor



It is LOW when the switch is not pressed

In this project, we are using “pull-up” circuit but both are OK to use.

Challenge

Q: Use both on-board and external switches. When both switches are not pressed, both LED 4 and LED 5 will be on. If the on-board switch is pressed, LED 4 goes off. If external switch is pressed, LED 5 would go off.

Click here or scan QR code to watch the demo video

Congratulation! You have completed lesson 2 and learnt the following:

1. How to read the digital input signal.
2. How to control an LED using a switch.
3. How to construct a simple pull up/ pull down circuit.
4. How to use if else statement.

LESSON 3
ANALOG OUTPUT

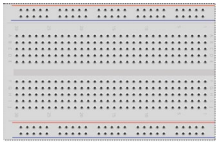
Project 6: Construct An LED Circuit

1. Get read these components.

LED

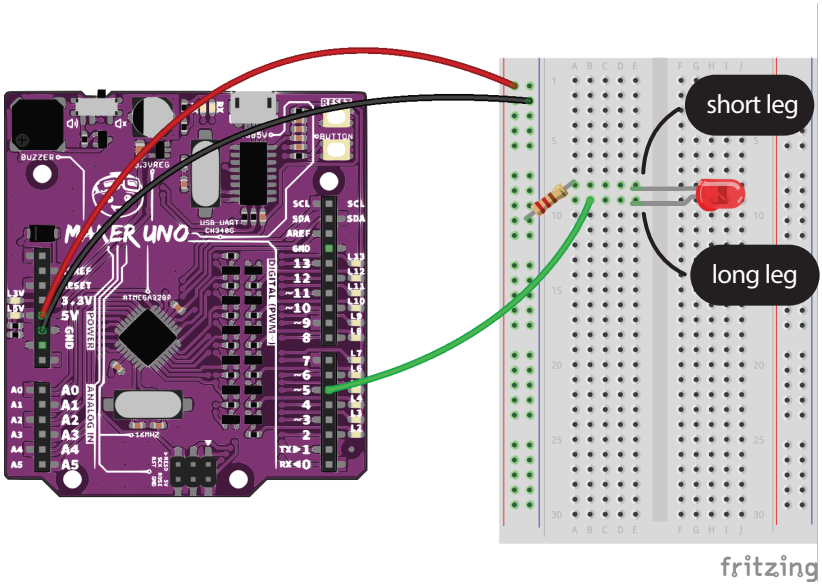
220 ohm resistor

Jumper Wire



Breadboard

2. Construct the circuit as shown below.



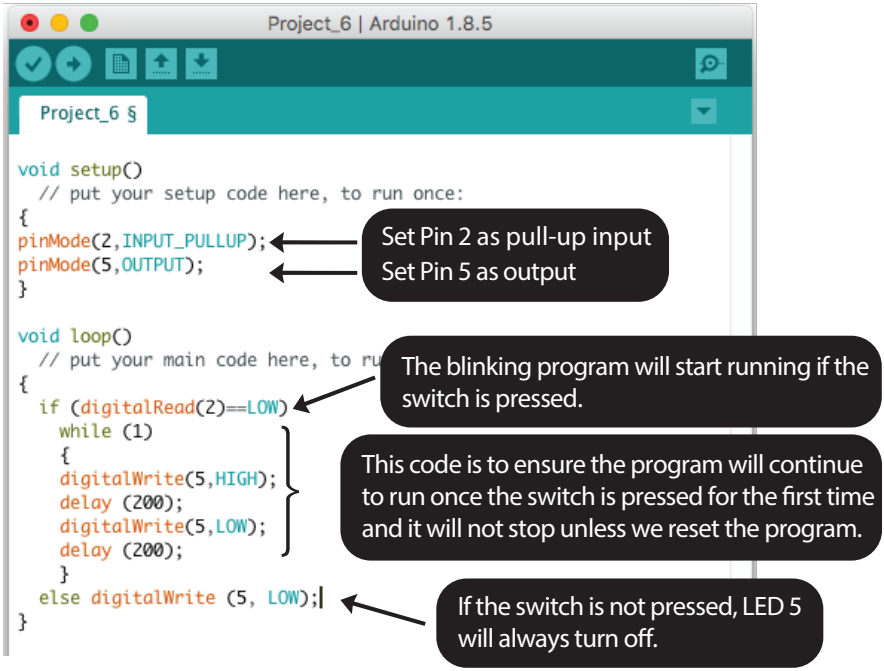
3. Upload these codes to your board.

5. Check your result.

Click here or scan QR code to watch the demo video

Both on-board LED and the external LED at Pin 5 will blink after the on-board switch is pressed. Nothing can stop the program unless the reset button is pressed.

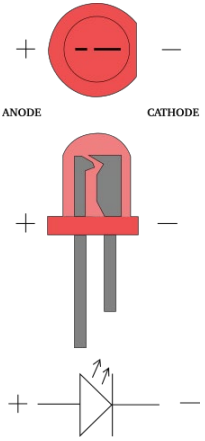
How it works



Good To Know

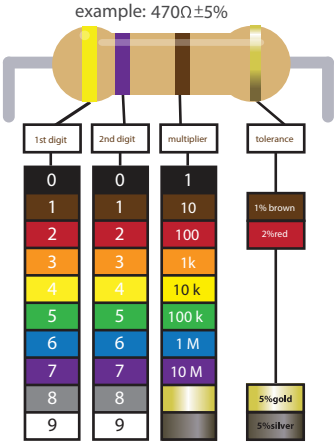
Basic Electronics

1. Light emitting diode (LED) has 2 pins, you need to connect it to the correct polarity, the long leg to the + and the short leg (flat edge at the body) to the - . Otherwise it will not work.



2. Resistor doesn't has + and - terminal, you can connect the pins anyway you like..

3. The color bands on a resistor can tell about its value. Please refer to the chart beside to undertand how to read them.



Coding Syntax

A while loop will loop continuously, and infinitely, until the condition inside the parenthesis, ( ) becomes false.

```
while (condition)
{
// statements
}
```

Example:
1. Do something for 200 times

```
var = 0;
while (var < 200)
{
  //do something
  var ++;
 }
```

2. Do something endlessly

```
while (1)
{
// do something
}
```

Project 7: Fade An LED

1. Remain the same circuit used in Project 6, upload these codes to your board.

Project_9

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

void loop()
{
  analogWrite(5, 60);
  delay (200);
  analogWrite(5, 55);
  delay (200);
  analogWrite(5, 50);
  delay (200);
  analogWrite(5, 45);
  delay (200);
  analogWrite(5, 40);
  delay (200);
  analogWrite(5, 35);
  delay (200);
  analogWrite(5, 30);
  delay (200);
  analogWrite(5, 25);
  delay (200);
  analogWrite(5, 20);
  delay (200);
  analogWrite(5, 15);
  delay (200);
  analogWrite(5, 10);
  delay (200);
  analogWrite(5, 5);
  delay (200);
  analogWrite(5, 0);
  delay (1000);

}
```

2. Check your result.

Click here or scan QR code to watch the demo video

You will notice that the LED's brightness is reduced gradually until it is completely off.

How it works

Project_9

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

void loop()
{
  analogWrite(5, 60);
  delay (200);
  analogWrite(5, 55);
  delay (200);
  analogWrite(5, 50);
  delay (200);
  analogWrite(5, 45);
  delay (200);
  analogWrite(5, 40);
  delay (200);
  analogWrite(5, 35);
  delay (200);
  analogWrite(5, 30);
  delay (200);
  analogWrite(5, 25);
  delay (200);
  analogWrite(5, 20);
  delay (200);
  analogWrite(5, 15);
  delay (200);
  analogWrite(5, 10);
  delay (200);
  analogWrite(5, 5);
  delay (200);
  analogWrite(5, 0);
  delay (1000);

}
```

Set Pin 5 as output

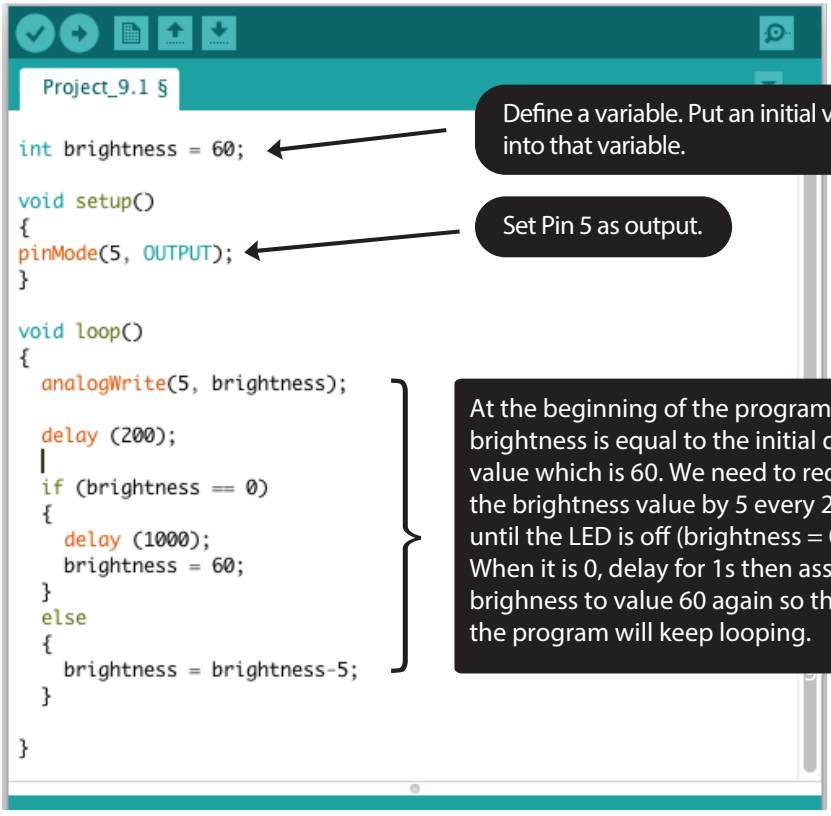
Set LED 5's brightness to 60.
hold for 200ms.
Set LED 5's brightness to 55.
hold for 200ms.
Set LED 5's brightness to 50.
hold for 200ms.
Set LED 5's brightness to 45.
hold for 200ms.
Set LED 5's brightness to 40.
hold for 200ms.
Set LED 5's brightness to 35.
hold for 200ms.
Set LED 5's brightness to 30.
hold for 200ms.
Set LED 5's brightness to 25.
hold for 200ms.
Set LED 5's brightness to 20.
hold for 200ms.
Set LED 5's brightness to 15.
hold for 200ms.
Set LED 5's brightness to 10.
hold for 200ms.
Set LED 5's brightness to 5.
hold for 200ms.
Set LED 5's brightness to 0 (off the LED).
hold for 1000ms.

3. We have an easier way to simplify the codes above. Modify your code into this then upload to your board:

4. Check your result. It should be exactly the same as the earlier result.

Click here or scan QR code to watch the demo video

How it works



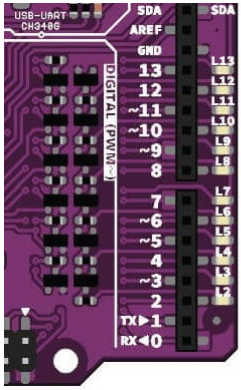
Good To Know

Arduino Function

1. There are 2 types of outputs, digital and analog. Digital means 0 (LOW) or 1 (HIGH). Analog means variable from 0 to 255. You can use the following function to control the analog output.

   analogWrite(pin, value);

2. However, not every pin on the board has analog output. Only Pin 3, 5, 6, 9, 10 and 11 has analog output.
(the pin with ~ sign)



Coding Syntax

1. The beauty of coding is allowing the programmer to simplify a very long instruction into a few lines of codes that executes same task.
2. Reducing the number of lines also helps to reduce the processing time, thus it is very important to always optimize your code.
3. In this project, we need to reduce the LED brightness every 200ms. So we define it as an integer variable at the beginning of our program. "brightness" is just a variable name, you can name it as A or B if you want.
4. These are the comparison commands you can use in coding.

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)

Challenge

Q: Fade the LED 5 only when the on-board switch is pressed .

Click here or scan QR code to watch the demo video

Congratulation! You have completed lesson 3 and learnt the following:

1. Construct a basic LED circuit.
2. How to use while loop.
3. The difference between digital output and analog output.
4. The importance of reducing the number of lines of codes.

LESSON 4
MELODY TONE

Project 8: Compose Basic Tones

1. Open a new sketch, write these codes into the sketch then upload to your board.

2. Check your result.

Click here or scan QR code to watch demo video

You should be able to hear the basic tone of “Do Re Mi Fa So La Ti”.

How it works

Project_40

```
void setup() {
  // put your setup code here, to run once:
pinMode (8,OUTPUT);
}

void loop() {
  // put your main code her
tone (8, 262, 250);
delay (325);
tone (8, 294, 250);
delay (325);
tone (8, 330, 250);
delay (325);
tone (8, 349, 250);
delay (325);
tone (8, 392, 250);
delay (325);
tone (8, 440, 250);
delay (325);
tone (8, 494, 250);
delay (1000);

}
```

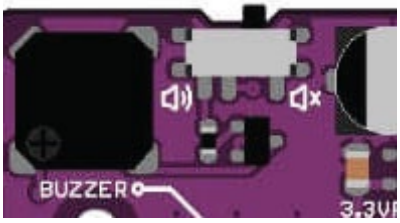
Set Pin 8 as output

Get piezo buzzer to play "Do" (C4) for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play "Re" (D4) for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play "Me" (E4) for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play "Fa" (F4) for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play "So" (G4) for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play "La" (A4) for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play "Ti" (B4) for 1 beat.
Delay for 1s.

Good To Know

Maker UNO's Feature

1. The on-board piezo buzzer is connected to Pin 8. You need to switch it to the sign if you want use the piezo buzzer. When you connect it with other I/O, you need to switch it to



Arduino Function

1. To play a tone from the piezo buzzer.

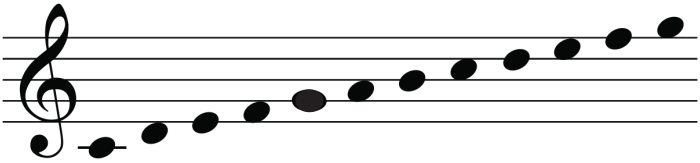
   tone(pin, frequency, duration);
   pin: The pin that connected with a piezo buzzer
   frequency: the frequency of the tone in hertz
   duration: the duration of the tone in miliseconds (whether it is 1 beat, 2 beats or etc.)

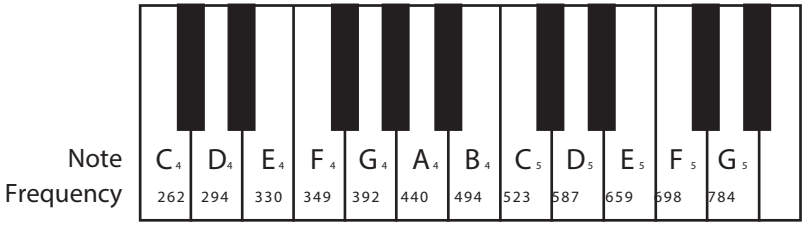
2. There is always a delay after tone. The delay has to drag 30% longer than the tone duration to ensure the tone is completed.

   e.g.: The tone duration for 1 beat is 250ms, thus we need to allow a 325ms delay.

Music Sheet

1. The position of a music note on the staff (i.e. the five horizontal lines) determines its pitch (the frequency of the piezo buzzer). The higher the note sits on the staff, the higher the pitch of the sound and vice versa.



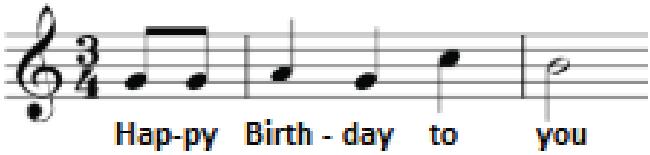


2. The different musical notations used to tell us the duration a note is to be played. In Arduino, the duration to play quarter note (1 beat) is 250ms.

Sign	Name	Relative Length	Duration	
	Semibreve	Whole Note	4 beats	1000ms
	Minim	Half Note	2 beats	500ms
	Crotchet	Quarter Note	1 beat	250ms
	Quaver	Eight Note	1/2 beat	125ms
	Semiquaver	Sixteenth Note	1/4 beat	63ms

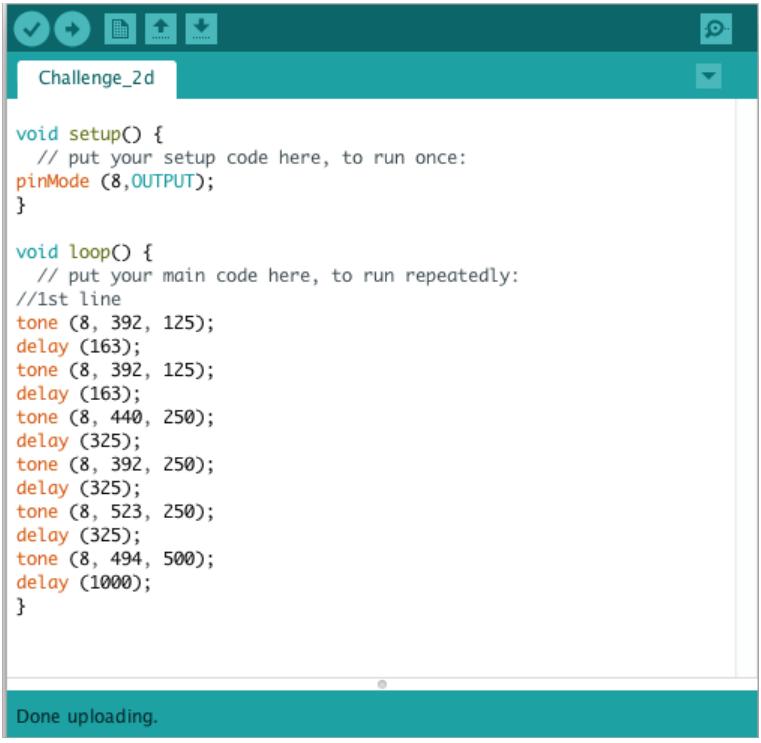
Project 9: Compose “Happy Birthday” Melody

Compose the first line of the “Happy Birthday To You” song.



Note	G4	G4	A4	G4	C5	B4
Frequency	392	392	440	392	523	494
Duration	125ms	125ms	250ms	250ms	250ms	500ms

1. Modify your previous code into this then upload to your board.



2. Check your result.

Click here or scan QR code to watch the demo video

How it works

Challenge_2d

```
void setup() {
  // put your setup code here, to run once:
pinMode (8,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
//1st line
tone (8, 392, 125);
delay (163);
tone (8, 392, 125);
delay (163);
tone (8, 440, 250);
delay (325);
tone (8, 392, 250);
delay (325);
tone (8, 523, 250);
delay (325);
tone (8, 494, 500);
delay (1000);
}
```

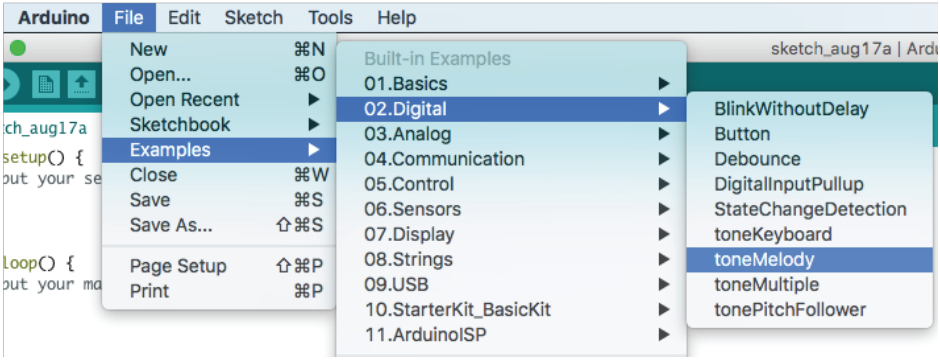
Set Pin 8 as output

Get piezo buzzer to play G_4 for 1/2 beat.
Add delay until the note is completed.
Get piezo buzzer to play G_4 for 1/2 beat.
Add delay until the note is completed.
Get piezo buzzer to play A_4 for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play G_4 for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play C_5 for 1 beat.
Add delay until the note is completed.
Get piezo buzzer to play B_4 for 2 beats.
Delay for 1s.

Project 10: Optimize Your Code

Instead of manually key in the tones' frequency and duration line by line, we can optimize it using a given sample code to shorten our prvious code. This allows us to compose a long melody effortlessly.

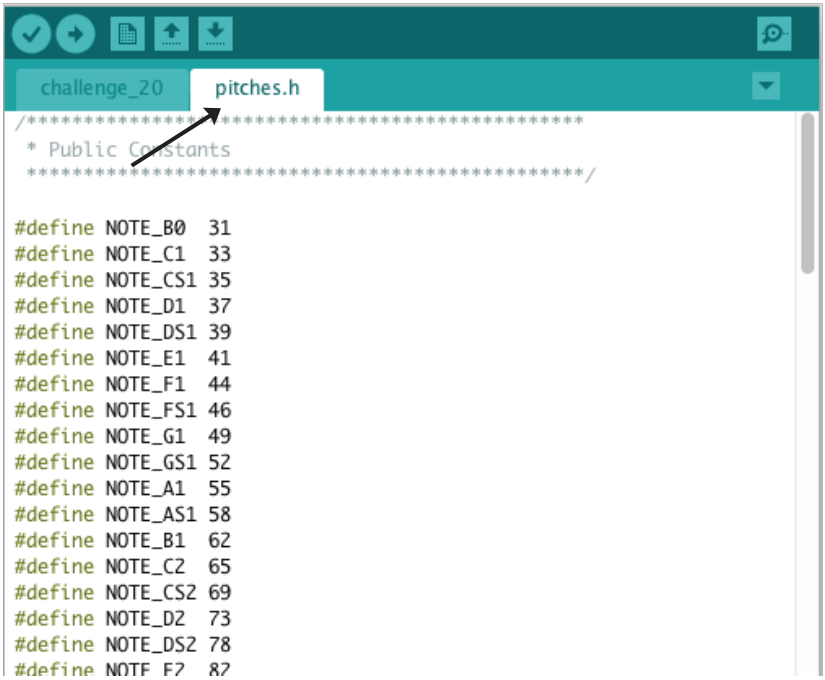
1. Open a sample code. File > Examples > 02 Digital > toneMelody



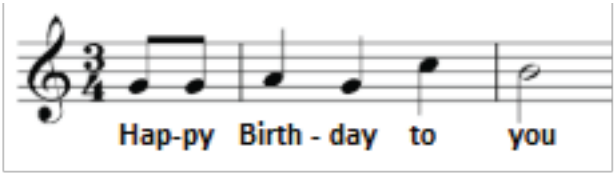
2. From the example given. Every note's frquency is predefined from note B0 until DS8. What we need to do is to key in the notes we want and the note duration in seqence.

```
#include "pitches.h"

// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};
```

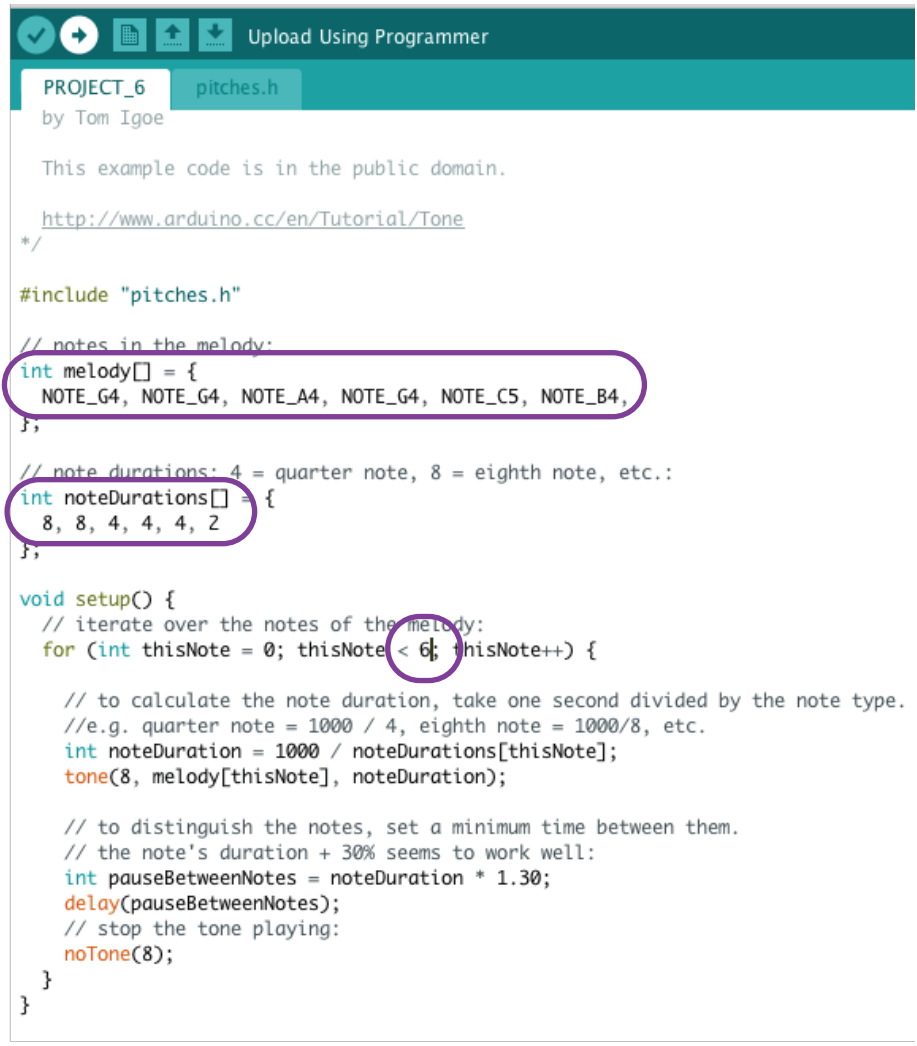
You can click the “pitches.h” tab to view the predefined notes’ frequency.



3. Refer to the same music sheet again then key in the note and note duration accordingly.



Note	G4	G4	A4	G4	C5	B4
Frequency	392	392	440	392	523	494
Duration	125ms Eighth note (8)	125ms Eighth note (8)	250ms Quarter note (4)	250ms Quarter note (4)	250ms Quarter note (4)	500ms Half note (2)



4. Check your result.

You are supposed to get the same result as Project 9's.

How it works


Upload Using Programmer
PROJECT_6
pitches.h
by Tom Igoe
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Tone
*/
#include "pitches.h"
// notes in the melody:
int melody[] = {
NOTE_G4, NOTE_G4, NOTE_A4, NOTE_G4, NOTE_C5, NOTE_B4,
};
// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
8, 8, 4, 4, 4, 2
};
void setup() {
// iterate over the notes of the melody:
for (int thisNote = 0; thisNote < 6; thisNote++) {
// to calculate the note duration, take one second
//e.g. quarter note = 1000 / 4, eighth note =
int noteDuration = 1000 / noteDurations[thisNote];
tone(8, melody[thisNote], noteDuration);
// to distinguish the notes, set a minimum
// the note's duration + 30% seems to work
int pauseBetweenNotes = noteDuration * 1.30;
delay(pauseBetweenNotes);
// stop the tone playing:
noTone(8);
}
}
Insert the notes to be played in sequence.
Insert the notes' duration in sequence.
We are using a for loop here. 6 means there are 6 notes to be played in this program.
1000 divided by noteDuration is to get duration in miliseconds. e.g. 1000/ 8 = 125ms
This is exactly what we wrote previously. tone (8, frequency, duration);
delay (duration + 30%);

Good To Know

Coding Syntax

1. Did you notice that the melody only play once and it doesn't repeat like the previous programs?

   This is because the entire code is written under void setup() instead of void loop (). You can only repeat it if you reset the program (by pressing the reset button) .

2. To use "for" statement.

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

Example:

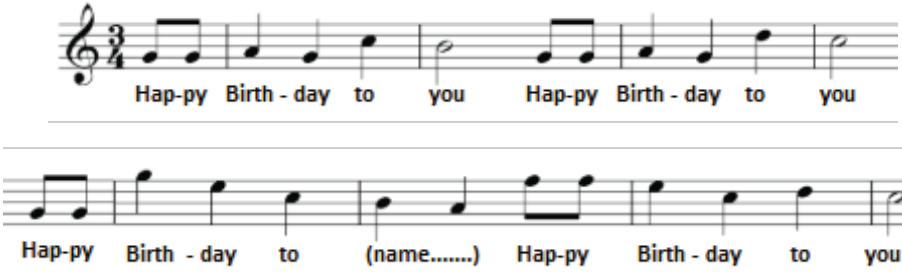
```
{
analogWrite (PWMpin, 1);
delay (10);
analogWrite (PWMpin, 2);
delay (10);
analogWrite (PWMpin, 3);
delay (10);
analogWrite (PWMpin, 4);
delay (10);
}
```

The above codes can be shorten into 3 lines using "for" statement shown below:

```
for (int i=0; i <= 4; i++)
{
    analogWrite (PWMpin, i);
    delay (10)
}
```

Challenge

Q: Program Maker UNO to play a complete “Happy Birthday to you” melody when the on-board switch is pressed.



Click here or scan QR code to watch the demo video

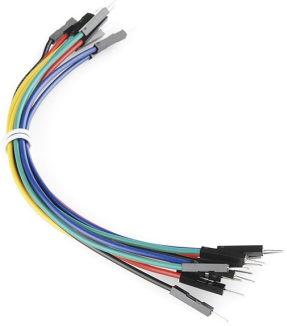
Congratulation! You have completed lesson 4 and learnt the following:

1. How to program a tone from Maker UNO.
2. How to compose a simple melody using Maker UNO.
3. How to load Arduino’s sample program.
4. How to use for statement.

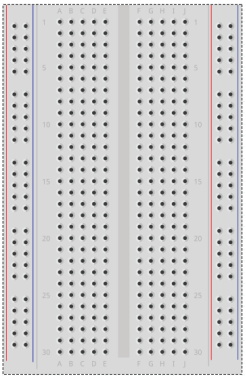
LESSON 5
ANALOG INPUT

Project 11: Display Analog Value on Serial Monitor

1. Get ready these components



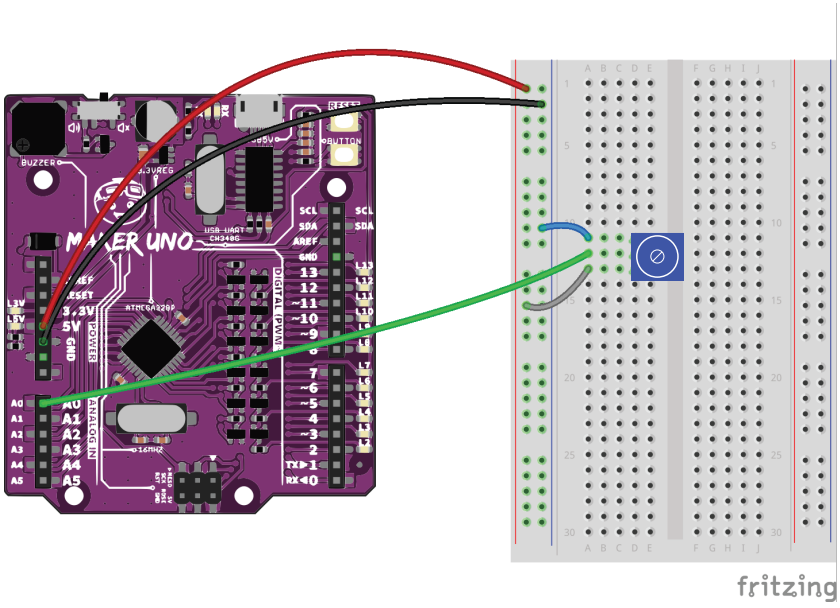
jumper wires



breadboard

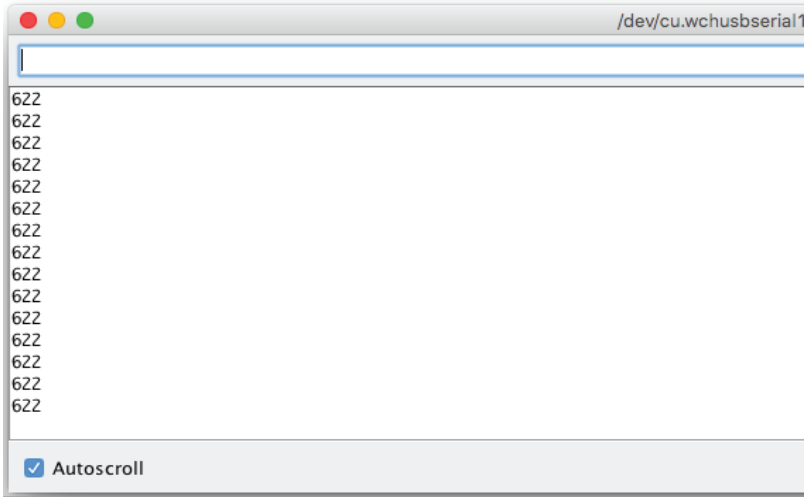
Potentiometer

2. Construct the circuit as shown below.

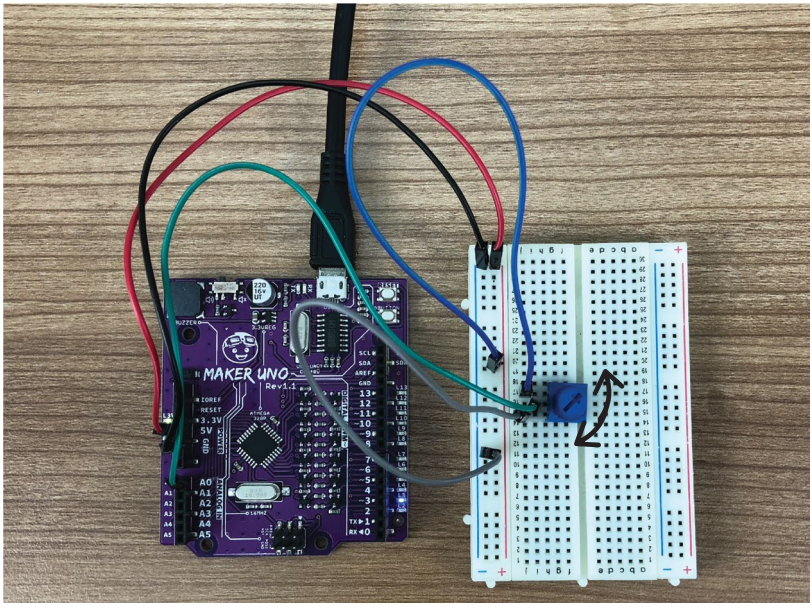


3. Write these codes then upload to your board.

4. Once done uploading, click (serial monitor) located at the right top bar. Immediately, you will see a pop up window appear.



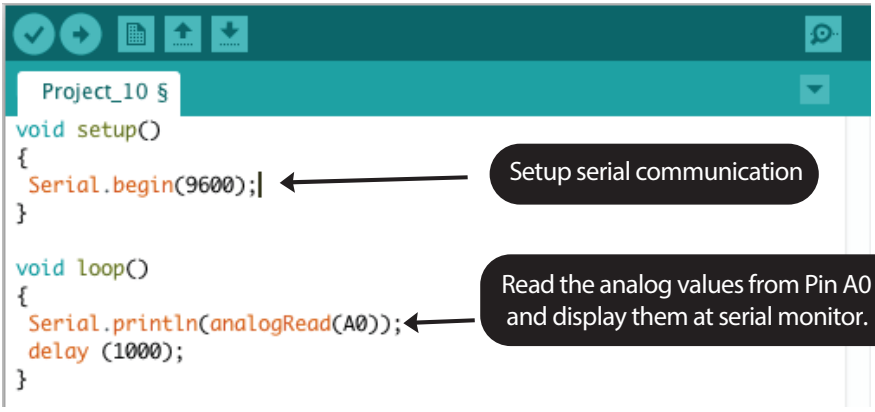
5. Turn the potentiometer clockwise and counter clockwise and you will find the value displayed on the serial monitor changes as you adjust the potentiometer. The display values are analog value provided by the potentiometer.



6. Check your result.

Click here or scan QR code to watch demo video

How it works



Good To Know

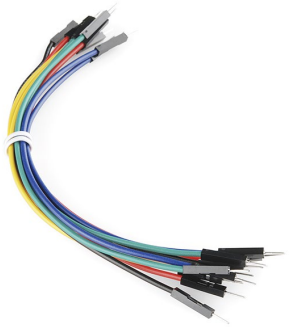
Arduino Function

1. Only pin A0, A1, A2, A3, A4 and A5 have analog input function. Hence we need to connect the analog sensors to these pins if we need analog input values.

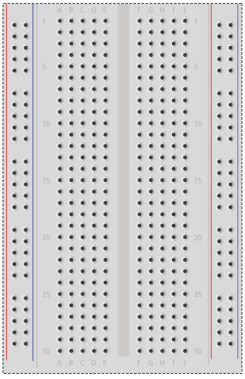
2. Digital input only has 2 values, either 0 (low) or 1 (high). However in analog input we can have variable from 0 to 1023.

Project 12: Read Analog IR Sensor

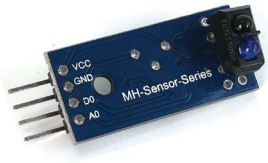
1. Get ready these components.



Jumper wires

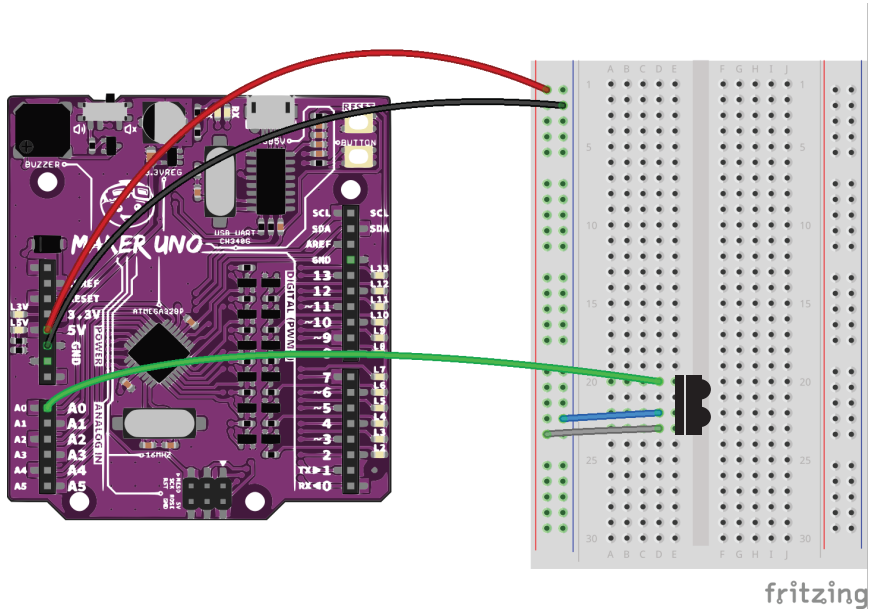


Breadboard



IR sensor

2. Construct the circuit as shown below.

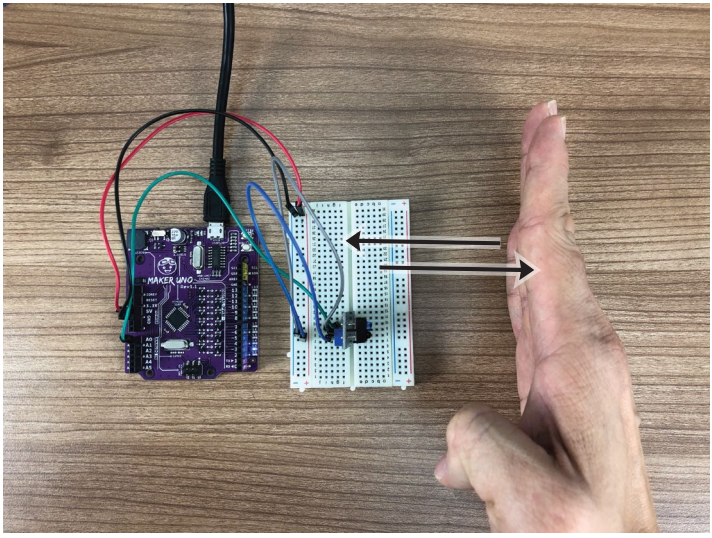


3. Upload the same program used in Project 11 to your board.

Project_10 §

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

void loop()
{
 Serial.println(analogRead(A0));
 delay (1000);
}
```

4. Click the serial monitor to observe the analog input's value. Move your palm closer to the IR sensor to see the changes of the analog value.

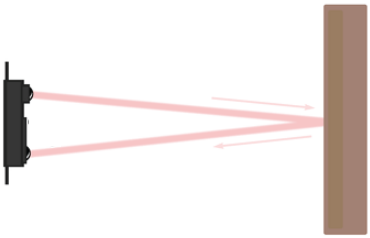


5. Check your result.

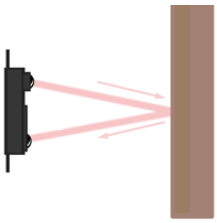
Click here or scan QR code to watch demo video

How it works

Infrared (IR) sensor consisted of an IR transmitter and IR receiver. The transmitter will transmit infrared light when power up. If there is an object placed infront of the sensor, the IR light will be reflected and received by the IR receiver. The distance between the object and IR sensor will change the intensity of the IR light received by the receiver. The intensity will then be converted into analog value that you will be observed in the serial monitor. When the intensity is high, the analog value that we receive will be low.



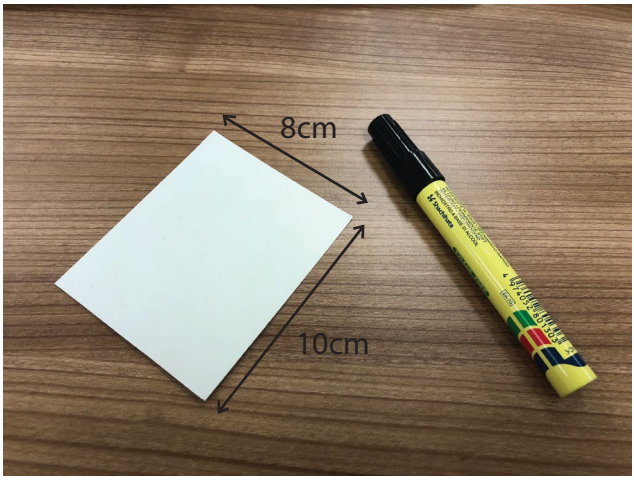
If the distance of the object is far, the intensity of the IR is low, the analog value is high.



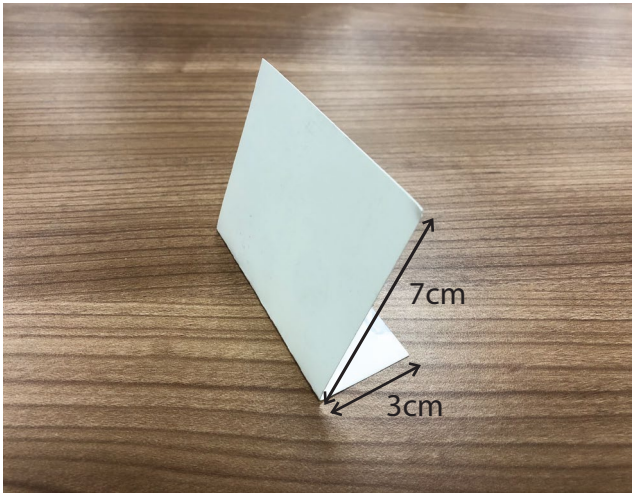
If the distance of the object is close, the intensity of the IR is high, the analog value is low.

Project 13: IR Sensor To Detect Black Line

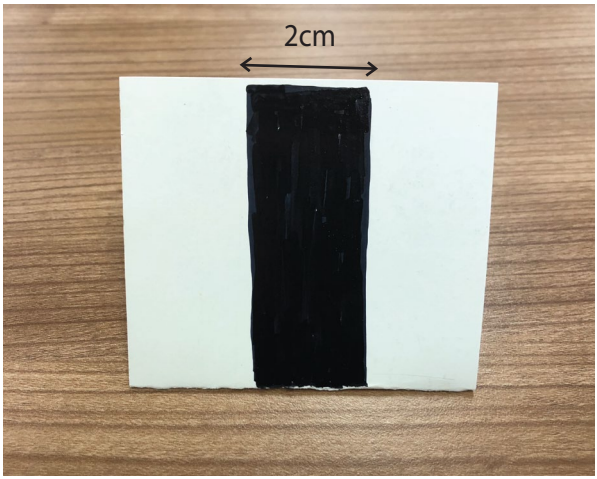
1. Get ready a piece of white cardboard and black marker.



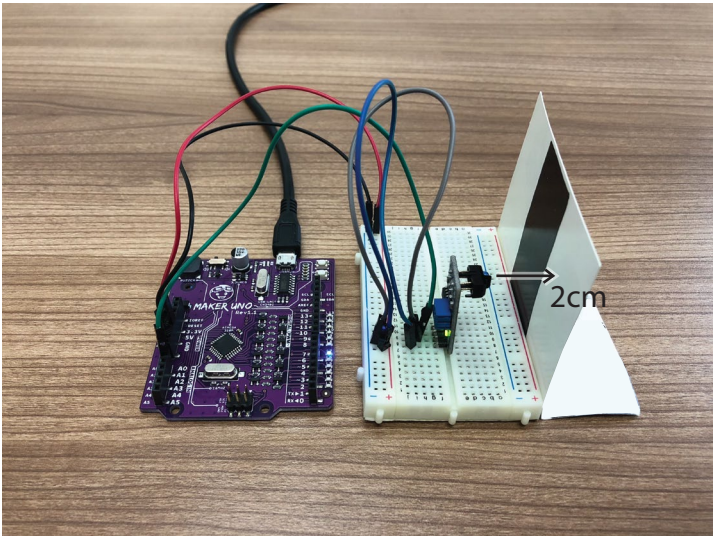
2. Fold the cardboard as shown below:



3. Draw a black line with 2cm thickness in the middle of the cardboard using the black marker.



4. Fixed the distance between the cardboad and the IR sensor at approximately 2cm. Move the cardboard horizontally from the white surface to the black line.



5. Observe and record the analog values from the serial monitor when the IR sensor is facing the white surface and black line respectively.

	Analog Value
White surface	
Black line	

6. Write these codes into a new sketch and then upload to your board.

Project_11

```
int IRvalue;

void setup()
{
 Serial.begin(9600);
 pinMode (3, OUTPUT);
}

void loop()
{
 Serial.println(analogRead(A0));

 IRvalue = analogRead(A0);
 if(IRvalue > 500)
 {
  digitalWrite(3, HIGH);
  delay(200);
 }
 else
 {
  digitalWrite(3,LOW);
  delay (200);
 }
}
```

Note: IRvalue > 500 is the threshold for my case. You can change the value according to the analog values shown in your serial monitor. You can pick any value between the white surface and black line.

7. Check your result.

Click here or scan QR code to watch the demo video

LED 3 will light up when the black line is detected and it will go off when the white surface is detected.

How it works

Project_11

```
int IRvalue;

void setup()
{
 Serial.begin(9600);
 pinMode (3, OUTPUT);
}

void loop()
{
 Serial.println(analogRead(A0));

 IRvalue = analogRead(A0);
 if(IRvalue > 500)
 {
  digitalWrite(3, HIGH);
  delay(200);
 }
 else
 {
  digitalWrite(3,LOW);
  delay (200);
 }
}
```

Define a variable name as "IRvalue"

Setup serial communication

Set Pin 3 as output

Display the analog input for Pin A0 at serial monitor.

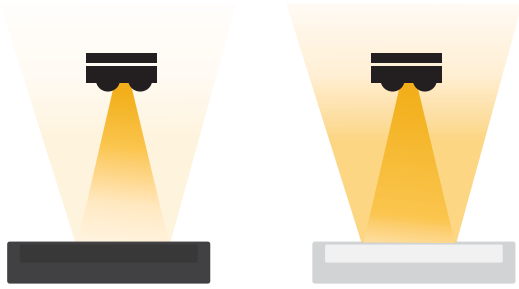
Assign the analog value at Pin A0 to IRvalue

If IRvalue > 500 (means black line is detected) on the LED 3

Else, off the LED 3

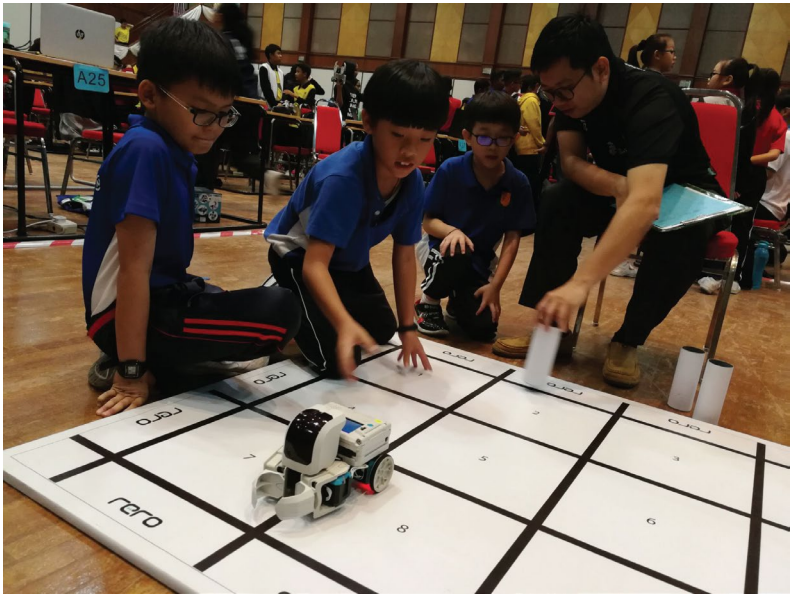
Good To Know

Besides detecting object and measuring distance, the IR sensor also can be used to differentiate black and white surfaces. This is because the light surface is able to reflect most of the IR light while dark surface tends to absorb the IR light. Hence the IR receiver will receive less IR light when it faces the dark surface.



dark surfaces absorb more
IR light than light surfaces

This concept allows the IR sensor to be used at the mobile robots to follow lines.



Coding Syntax

In this project, the analog value from the IR sensor is changed based on the color of the object. We need to store the variable number to a certain place and only summon it when needed. To do that, we can assign a variable.

int var = val; or int var;

var: your variable name (can be IRvalue or A or B or anything you like)

val: the value you assign to the variable (can be any number from -32,768 to 32,767)

Challenge

Q: Utilise the analog values from a potentiometer to change the blinking speed for LED 3.

Click here or scan QR code to watch demo video

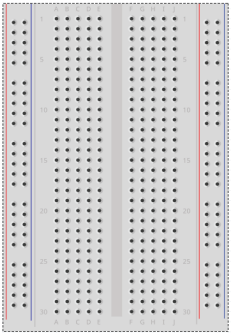
Congratulation! You have completed lesson 5 and learnt the following:

1. How to read analog input.
2. How to use serial monitor.
3. What is IR sensor and how it works.
4. How to assign variable numbers

LESSON 6
DC MOTOR

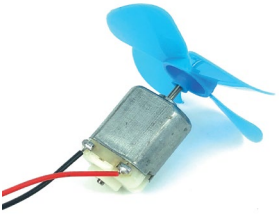
Project 14: Control A DC Motor

1. Get ready these components

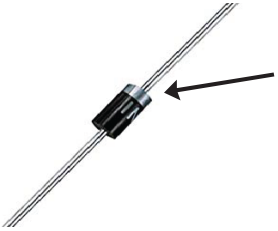


Breadboard

Jumper wires



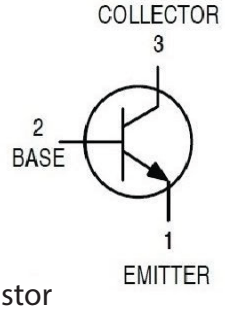
DC motor



The terminal with a white stripe should connect to the positive wire (red)

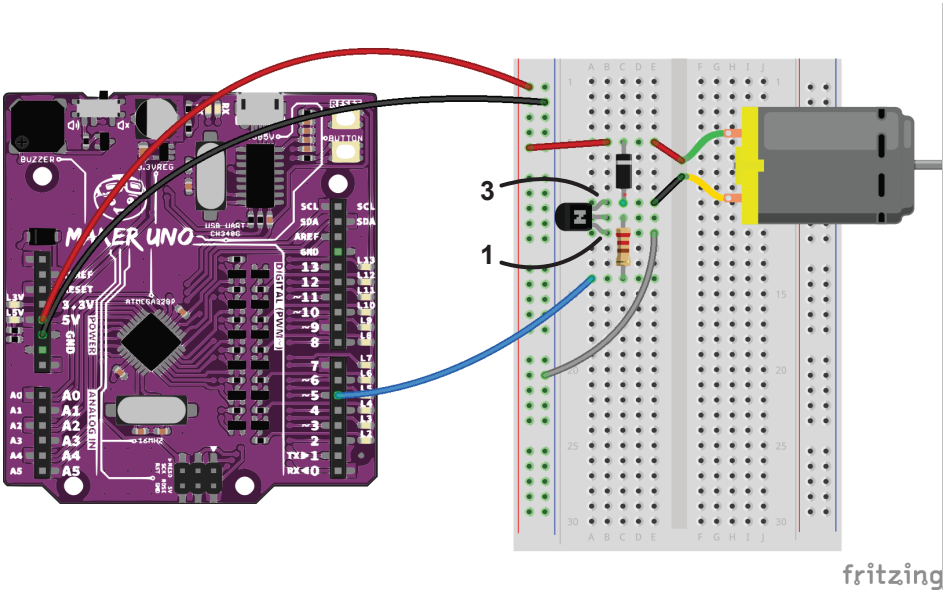
Diode

220 ohm resistor



2N2222 Transistor

2. Construct the circuit as shown below.



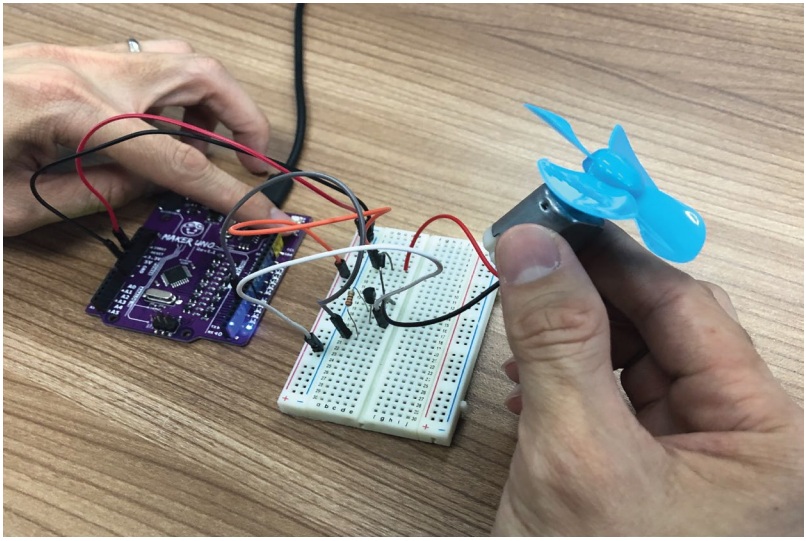
3. Write these codes then upload to your board.

Project_14 | Arduino 1.8.5

Project_14

```
void setup()
{
  pinMode(2, INPUT_PULLUP);
  pinMode(5,OUTPUT);
}

void loop()
{
 if (digitalRead(2)==LOW)
  {
    while (1)
      {
        analogWrite(5,255);
        delay(3000);
        analogWrite(5,80);
        delay(3000);
      }
  }
 else digitalWrite (5,LOW);
}
```

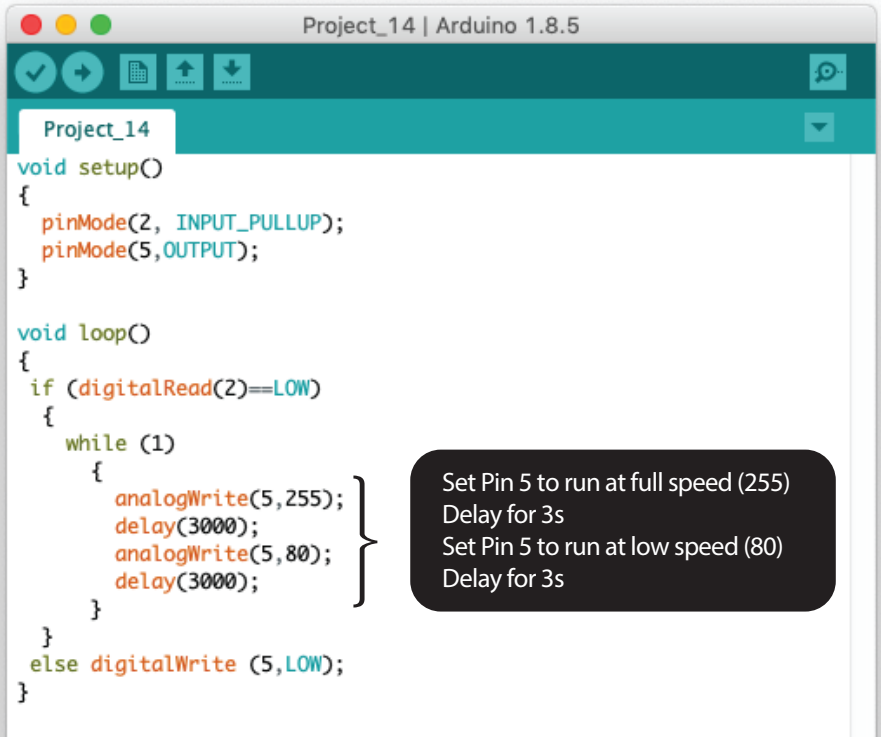
4. Hold the DC motor then press the on-board pushbutton switch.



5. Check your result.

Click here or scan QR code to watch the demo video

How it works



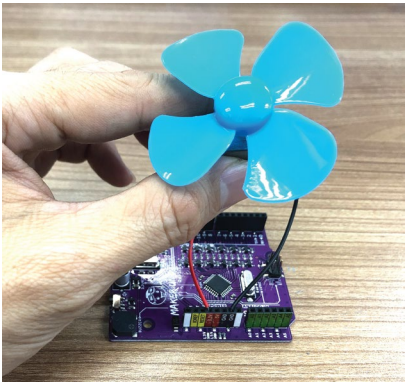
Troubleshooting

1. Make sure you are holding the DC motor, the motor may not have sufficuient torque to spin if the blade is attached on the table.
2. Also check the on-board LED at Pin 5. It should turn on in full brightness for 3s then dim for another 3s.
3. If everything is correct then you have to check your circuit board. Are all components connected correctly?

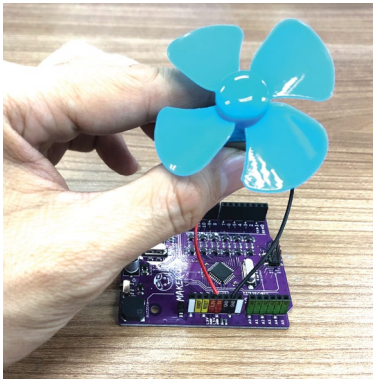
Good To Know

How To Control A Motor Speed?

To make a DC motor to spin, simply apply suitable input voltage on it. The input voltage is recommended by the motor manufacturer. The typical input voltage for DC motors are 3V, 6V and 12V. However, the motors still able to work even if you apply lesser or greater voltage than the recommended voltage. For the toy motor that we are using in this project, the recommded voltage is 3V, but we will try to apply 3.3V and 5V on it. Let's see what will happen.



Connect the red wire to 3.3V pin. The motor spins slower

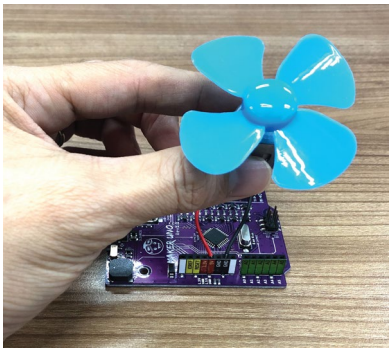


Connect the red wire to 5V pin. The motor spins faster

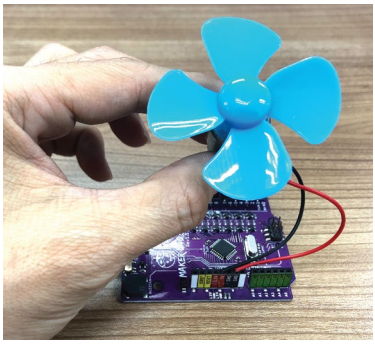
This test shows that the input voltage can change the speed of a DC motor. That's why in the earlier program, when we apply lower PWM value, the motor spins slower. But please take note that applying greater than the factory recommeded voltage to a motor for long run is not a good idea. It will shorten the motor's life cycle.

How To Control A Motor's Spinning Direction?

To change the spinning direction is easy. Simply change the positive and negative supply to the motor terminals.

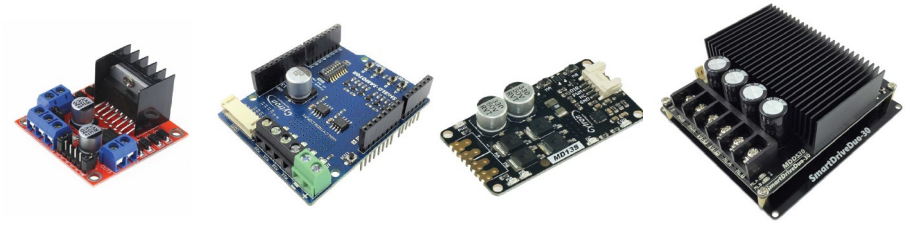


Red wire to 5V, black wire to the ground. The motor spins in 1 direction.



Black wire to 5V, red wire to the ground. The motor spins in another direction.

We are unable to change the motor spinning direction with the circuit we built earlier. The easier way to achieve that is to get a DC motor driver that able to control both speed and direction. You can either build it yourself or buy a ready made motor driver from the market. If you want to learn more about controlling a DC motor, you can refer to this tutorial: https://tutorial.cytron.io/2018/08/02/5-easiest-ways-to-control-a-dc-motor/



Sample of DC Motor drivers available in the market.

Project 15: Controlling Motor Speed Using A Pushbutton

1. Write these codes then upload to your board.

Project_15 §

```
int mode = 0;

void setup()
{
  pinMode(2, INPUT_PULLUP);
  pinMode(5,OUTPUT);
}

void loop()
{
  switch(mode)
    {
      case 0:
      analogWrite(5,0);
      break;

      case 1:
      analogWrite(5,100);
      break;

      case 2:
      analogWrite(5,255);
      break;

      default:
      mode = 0;
      break;
    }

  if (digitalRead(2)==LOW)
    {
      while(digitalRead(2)==LOW);
      mode++;
      if(mode == 3) mode = 0;
    }
}
```

2. Check your result.

Click here or scan QR code to watch the demo video

How it works

Project_15 §

```
int mode = 0;

void setup()
{
  pinMode(2, INPUT_PULLUP);
  pinMode(5,OUTPUT);
}

void loop()
{
  switch(mode)
    {
      case 0:
      analogWrite(5,0);
      break;

      case 1:
      analogWrite(5,100);
      break;

      case 2:
      analogWrite(5,255);
      break;

      default:
      mode = 0;
      break;
    }

  if (digitalRead(2)==LOW)
    {
      while(digitalRead(2)==LOW);
      mode++;
      if(mode == 3) mode = 0;
    }
}
```

Define a variable name "mode", set the initial value as 0.

if mode = 0, motor runs at speed 0 (stop)

if mode = 1, motor runs at speed 100 (slow)

if mode = 2, motor runs at speed 255 (full speed)

if mode is neither 0, 1 or 2, set mode to 0

if switch2 is pressed,
check switch2 again until it is released.
then mode value + 1
If mode value = 3, mode value set to 0

When the switch is pressed for the 1st time, mode = 1, motor will run at 100.
When the switch is pressed for the 2nd time, mode becomes 2, motor will run at full speed.
When the switch is pressed for the 3rd time, mode value will be reset to 0 again. The motor will stop.

Good To Know

Coding Syntax

1. Like if statemets, switch..case controls the flow of programs by allowing programmers to specify different code that should be executed in various conditions. This is useful when you want to use 1 switch to execute different set of program.

```
Example code
switch (var)
{
   case 1:
   // do something when var =1
   breaks;

   case 2:
   //do something when var = 2
   breaks;

   default:
   //if nothing else matches, do something
   breaks;
}
```

2. Did you notice we didn't put " ; " after while (1) at our previous projects but we put " ; " after while(digitalRead(2)==LOW) in this project?

without " ; "	with " ; "

while (condition)
{
// if the condition is true, program will run all codes inside { }
}
	while (condition); // if the condition is true, it will only run this line
	
	e.g.:
	while (digitalRead(2) == LOW);
	
	if switch 2 is pressed (condition is true), the program will stay there until switched 2 is released (condition false).

3. To stop the motor from spinning, you can write either of these:

digitalWrite (pin, LOW);
or
analogWrite (pin, 0);

Challenge

Q: The motor will run at a constant speed when you press and hold the pushbutton. It will stop when you release the button. When you press the pushbutton for the second time, the motor's speed will change.

Click here or scan QR code to watch the demo video

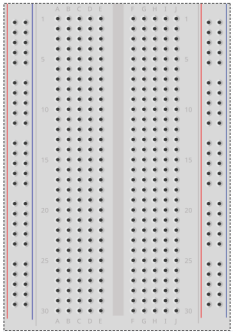
Congratulation! You have completed lesson 6 and learnt the following:

1. How to construct a circuit to run a DC motor.
2. How to control a DC motor speed.
3. How to use switch...case statement.

LESSON 7
ULTRASONIC SENSOR

Project 16: Setting Up Ultrasonic Sensor

1. Get ready these components

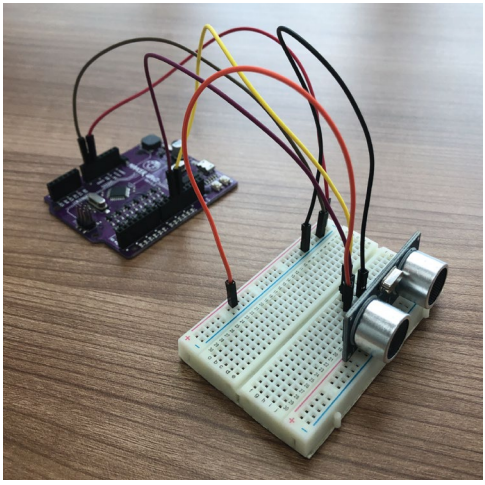


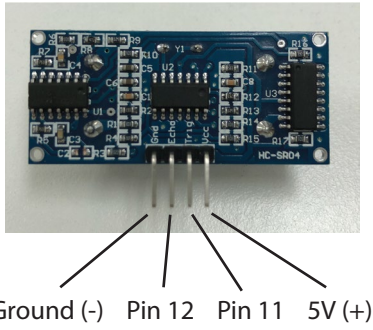
Breadboard

Jumper wires

Ultrasonic sensor

2. Construct the circuit as shown below.





Ground (-) Pin 12 Pin 11 5V (+)

Note: As shown in the figure. Please connect the sensor at the edge of the breadboard and ensure the jumpers are not blocking the sensor.

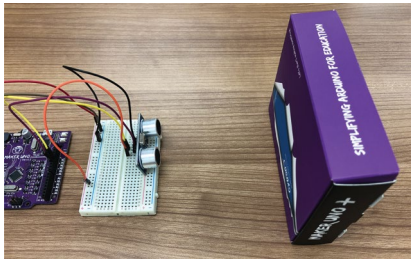
3. Write these codes then upload to your board.

Project_16 | Arduino 1.8.5

Project_16 §

```
long duration;
int distance;

void setup()
{
  pinMode (11, OUTPUT);
  pinMode (12, INPUT);
  Serial.begin (9600);
}

void loop()
{
  digitalWrite (11, LOW);
  delayMicroseconds (2);
  digitalWrite (11, HIGH);
  delayMicroseconds (10);
  digitalWrite (11, LOW);
  duration = pulseIn (12, HIGH);
  distance = duration*0.034/2;
  delay (50);

  Serial.print ("Distance = ");
  Serial.print (distance);
  Serial.println ("cm");
}
```

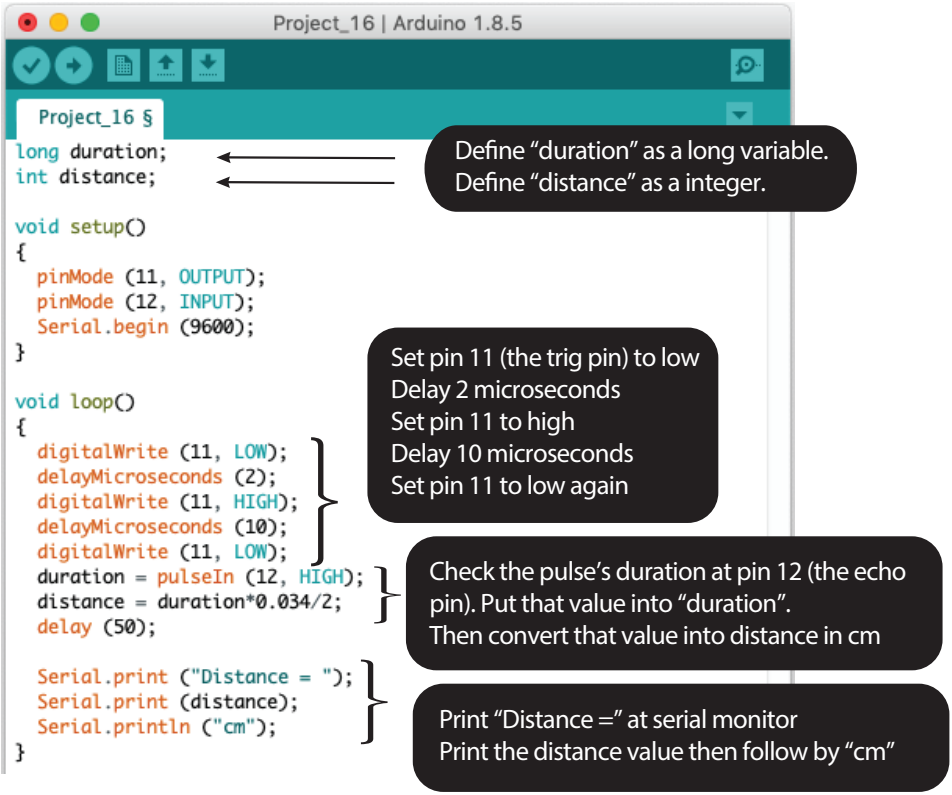
4. Once done uploading, click the serial monitor icon to view the result. You can place an object in front of the sensor. Try to adjust the distance between the object and the sensor to see the difference.



5. Check your result.

Click here or scan QR code to watch the demo video

How it works



6. We are going to group these codes into a self-declare function call “ultrasonic” because these are the same codes you need to write everytime you want to use the ultrasonic sensor. So that we can just call the function everytime we want to use it.

```
digitalWrite (11, LOW);
delayMicroseconds (2);
digitalWrite (11, HIGH);
delayMicroseconds (10);
digitalWrite (11, LOW);
duration = pulseIn (12, HIGH);
distance = duration*0.034/2;
delay (50);
```

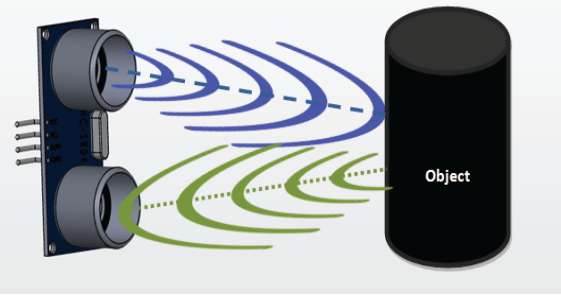
Modify your previous program into this then upload to your board.

7. Check your result again, you should be able to get the same result.

Good To Know

Ultrasonic Sensor

The baic idea of how an ultrasonic sensor works is simple. The sensor transmits ultrasound to the air and it will be bounced back if an object or obstacle is blocking its way. Measuring the duration needed for the ultrasound to transmit and receive, we are able to estimate the distance of the object from the sensor.



Ultrasonic sensor is widely used at our car's rear bumper to help the drivers to detect objects that cannot be seen. The advatage of an ultrasonic sensor over the infrared sensor is its insensitiveness to the surrounding lighting. it still can work under sunlight and in the darkness.

If you would like to learn more on how the distance is calculated, you can watch this video: https://www.youtube.com/watch?v=ZejQOX69K5M

Arduino Function

1. You can use this function to read the time cycle of a pulse.

   pulseIn (pin, value)

   pin: the number of the pin on which you want to read the pulse.
   value: type of pulse to read: either HIGH or LOW.

2. Data type

   You may ask why most of the time we are using “int” to define a variable but in this project we are using “long”. This is because we need to inform Arduino what kind of number we want to store at the variable, whether it is a small number, large number, a number with decimal points, and etc. In this project, the number we need to store for “duration” is large.

Variable	Number Range
char	-127 to 128
int	-32,768 to 32,767
long	-2,147,483,648 to 2,147,483,647
float	decimal numbers

Project 17: Build A Car's Rear Bumper Sensor

1. We are going to make the piezo buzzer to beep according to the distance sensed by the sensor. The buzzer beeps slower when the object is far from the sensor and it beeps faster when the object get closer just like our car's bumper sensor. Write these codes then upload to your board.

Project_17

```
long duration;
int distance;

void setup()
{
  pinMode (8, OUTPUT);
  pinMode (11, OUTPUT);
  pinMode (12, INPUT);
}

void loop()
{
  ultrasonic ();

  if (distance < 2)
  {
    tone (8, 349);
  }
  else
  {
    tone (8, 349);
    delay (50);
    noTone (8);
    delay (distance*10);
  }
}
```

```
void ultrasonic ()
{
  digitalWrite (11, LOW);
  delayMicroseconds (2);
  digitalWrite (11, HIGH);
  delayMicroseconds (10);
  digitalWrite (11, LOW);
  duration = pulseIn (12, HIGH);
  distance = duration*0.034/2;
  delay (50);
}
```

2. Check your result.

Click here or scan QR code to watch the demo video

How it works

Project_17

```
long duration;
int distance;

void setup()
{
  pinMode (8, OUTPUT);
  pinMode (11, OUTPUT);
  pinMode (12, INPUT);
}

void loop()
{
  ultrasonic ();

  if (distance < 2)
  {
    tone (8, 349);
  }
  else
  {
    tone (8, 349);
    delay (50);
    noTone (8);
    delay (distance*10);
  }
}
```

Set pin 8 (piezo buzzer) as output.

Call the ultrasonic function.

if distance is less than 2cm,
buzzer will beep continuously with note F_4

Else, buzzer will beep with note F_4
at the speed of "distance x 10".
eg: if distance = 5cm, beep speed 50ms
if distance = 10cm, beep speed 100ms

Challenge

Q: Build a Theremin using your Maker UNO and the ultrasonic sensor. Theremin is an electronic musical instrument that can be controlled without physical contact by the thereminist (the performer). With the ultrasonic sensor, we are able to divide the ultrasonic sensor's sensing range into a few zones that playing different tones.

Click here to watch the real Theremin in action

Click here or scan QR code to watch the demo video

Congratulation! You have completed lesson 7 and learnt the following:

1. How an ultrasonic sensor works.
2. How to create self-declare functions.