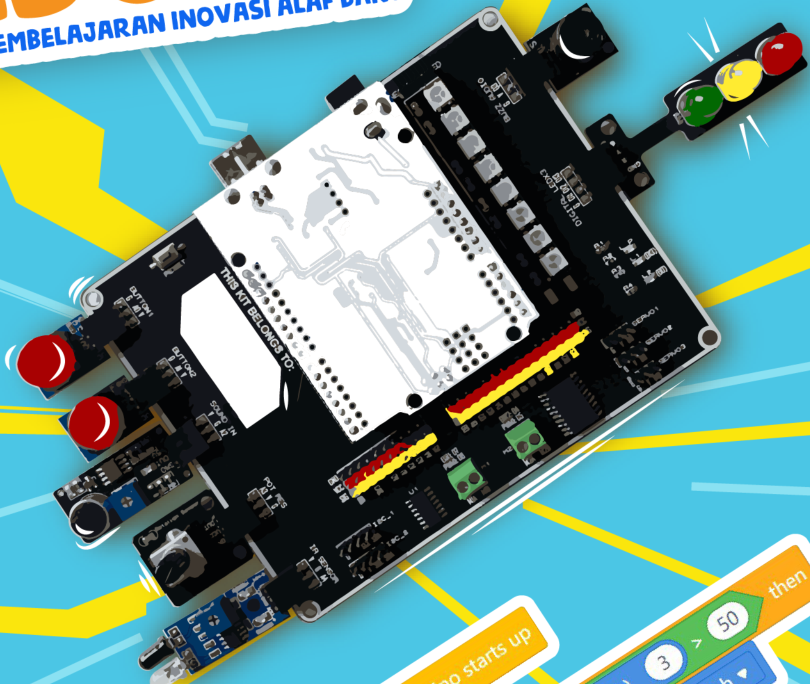
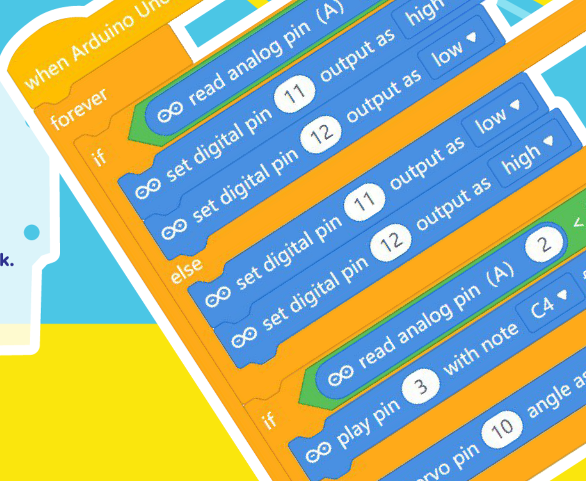
MCT
MICRO CONCEPT TECH

EKSPLORASI ILMU

REKA EDUKIT

KIT PEMBELAJARAN INOVASI ALAF BARU





```
when Arduino Uno starts up
forever
  if < read analog pin (A) 3 > 50 > then
    set digital pin 11 output as high
    set digital pin 12 output as low
  else
    set digital pin 11 output as low
    set digital pin 12 output as high
  if < read analog pin (A) 2 <
    play pin 3 with note C4
    rvo pin 10 angle a
```

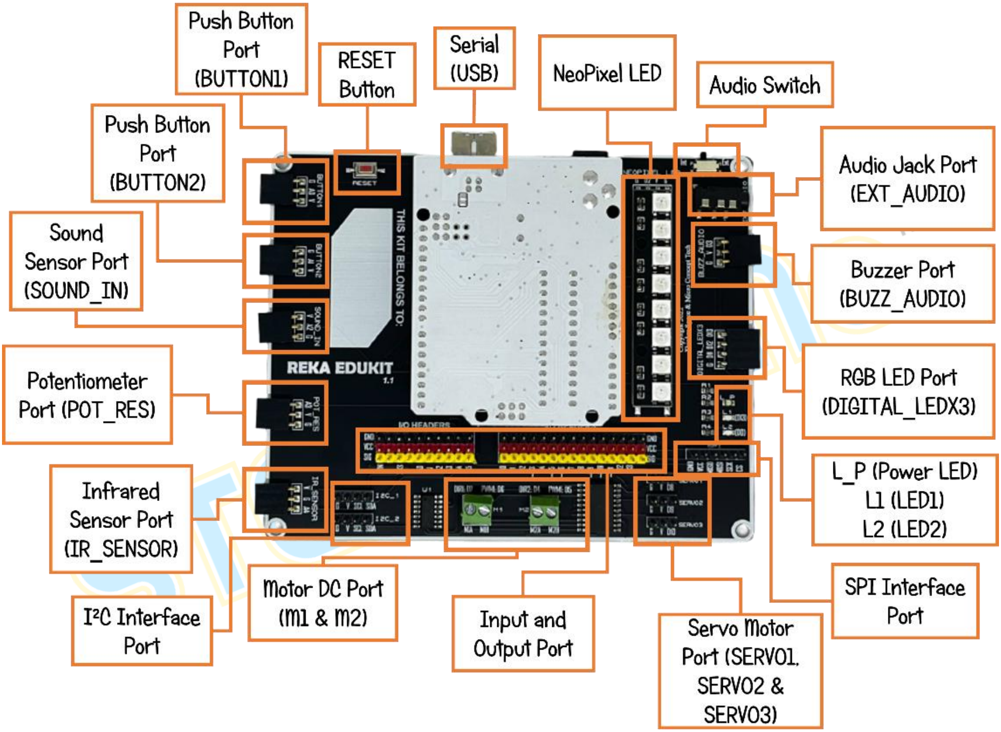
- Reka dan bangunkan pelbagai inovasi berteknologi.
- Papan mikropengawal mudah dinaik taraf.
- Menggunakan perisian sumber terbuka ("open-source")
- Pengekodan grafik yang mudah dan seronok.

UMUR 7+

CONTENTS

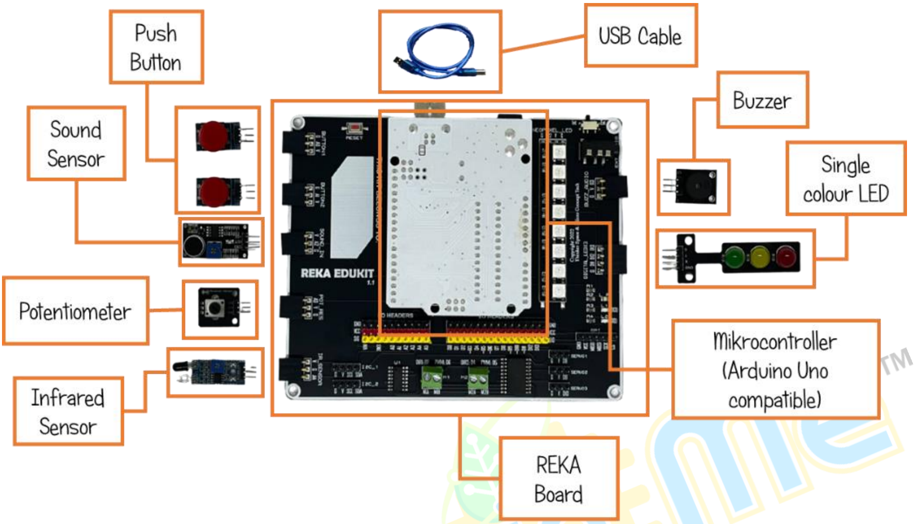
REKA EDUKIT ANATOMY

Devices on circuit board.

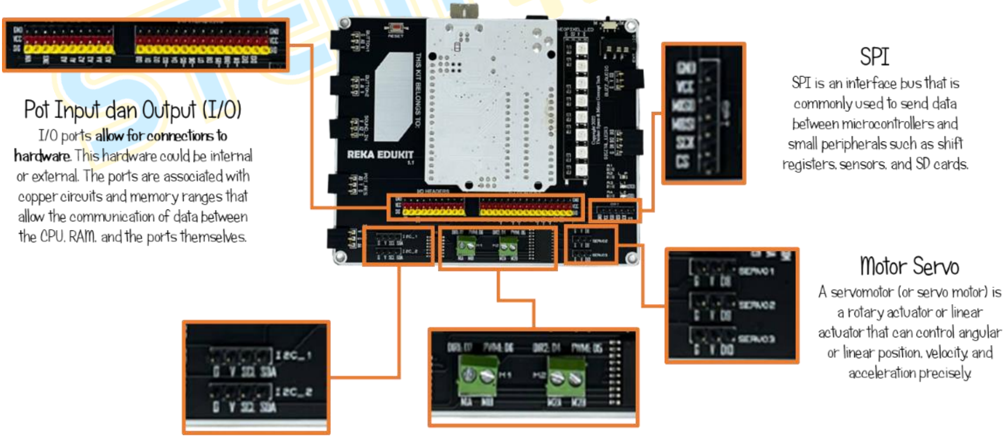

Push Button Port (BUTTON1)
RESET Button
Serial (USB)
NeoPixel LED
Audio Switch
Push Button Port (BUTTON2)
Audio Jack Port (EXT_AUDIO)
Sound Sensor Port (SOUND_IN)
Buzzer Port (BUZZ_AUDIO)
Potentiometer Port (POT_RES)
RGB LED Port (DIGITAL_LEDX3)
REKA EDUKIT
THIS KIT BELONGS TO:
L_P (Power LED)
L1 (LED1)
L2 (LED2)
Infrared Sensor Port (IR_SENSOR)
SPI Interface Port
I²C Interface Port
Motor DC Port (M1 & M2)
Input and Output Port
Servo Motor Port (SERVO1, SERVO2 & SERVO3)

REKA EDUKIT ANATOMY

Reka Edukit Peripherals.



Interface device.



Pot Input dan Output (I/O)

I/O ports **allow for connections to hardware**. This hardware could be internal or external. The ports are associated with copper circuits and memory ranges that allow the communication of data between the CPU, RAM, and the ports themselves.

SPI

SPI is an interface bus that is commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards.

Motor Servo

A servomotor (or servo motor) is a rotary actuator or linear actuator that can control angular or linear position, velocity and acceleration precisely.

I²C

I²C is an abbreviation for Inter-Integrated Circuit. It is a bus interface connection protocol used in serial communication devices.

DC Motor

A direct current (DC) motor is a type of rotary electrical motor that converts direct current (DC) electrical energy into mechanical energy.

REKA EDUKIT ANATOMY

Devices and interfaces pin number details on Arduino Uno

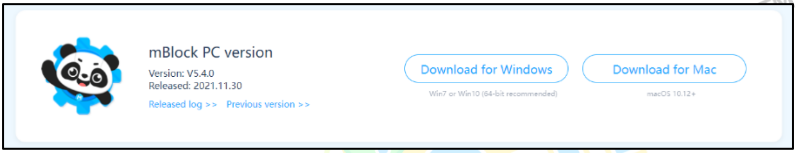
PERIPHERALS PORT	PIN ARDUINO UNO	PERPHERALS
BUTTON1	A0	Push Button 1
BUTTON2	A1	Push Button 2
SOUND_IN	A2	Sound Sensor
POT_RES	A3	Potentiometer
IR_SENSOR	A4	Infrared Sensor
BUZZ_AUDIO	D3	Buzzer
DIGITAL_LEDX3 (RED)	D11	Single Colour LED (Red)
DIGITAL_LEDX3 (YELLOW)	D12	Single Colour LED (Yellow)
DIGITAL_LEDX3 (GREEN)	D13	Single Colour LED (Green)
NEOPIXEL_LED	D2	NeoPixel LED
SERVO1	D8	Motor Servo 1
SERVO2	D9	Motor Servo 2
SERVO3	D10	Motor Servo 3
MOTOR1 DIRECTION	D7	Motor DC 1 (Rotation Direction)
MOTOR1 PWM (SPEED)	D6	Motor DC 1 (PWM / Motor Speed)
MOTOR2 DIRECTION	D4	Motor DC 2 (Rotation Direction)
MOTOR2 PWM (SPEED)	D5	Motor DC 2 (PWM / Motor Speed)
I2C – SCL	A5	I2C – SCL
I2C – SDA	A4	I2C – SDA
SPI – MISO	D12	SPI – MISO
SPI – MOSI	D11	SPI – MOSI
SPI – SCK	D13	SPI – SCK
SPI - CS	D10	SPI – CS
LED1	D13	Led 1
LED2	D0	Led 2

MBLOCK V5 SOFTWARE INSTALLATION

Step 1 MBlock v5 software can be obtained from

Link: https://mblock.makeblock.com/en-us/download/ @

Step 2 Download the latest version of MBlock v5 based on the computer's operating system.

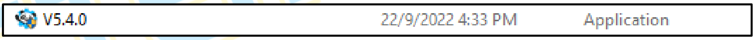


mBlock PC version

Version: V5.4.0

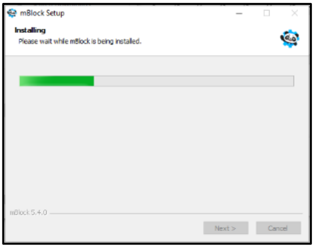
Released: 2021.11.30

Step 3 Click mBlock v5 on your download location.



V5.4.0 22/9/2022 4:33 PM Application

Step 4 Wait until mBlock v5 installation is complete.



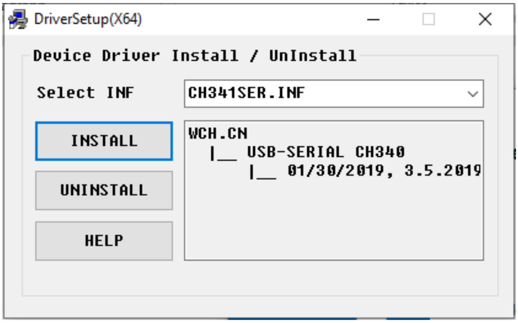
mBlock Setup

Installing

Please wait while mBlock is being installed.

mBlock 5.4.0

Step 5 ClicK "INSTALL"

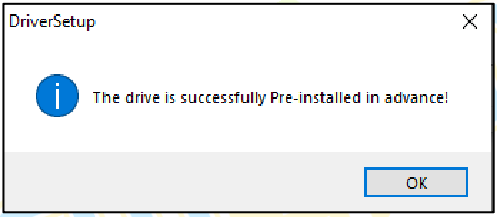

DriverSetup(X64)

Device Driver Install / UnInstall

Select INF: CH341SER.INF

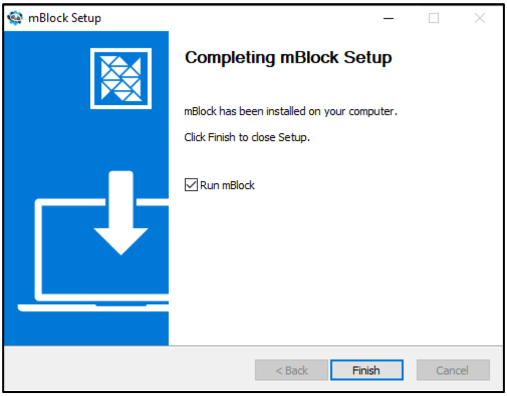
```
WCH.CN
 |__ USB-SERIAL CH340
      |__ 01/30/2019, 3.5.2019
```

Step 6 Click OK and exit.


DriverSetup

The drive is successfully Pre-installed in advance!

Step 7 Tandakan "Run mBlock". Klik "Finish".


mBlock Setup

Completing mBlock Setup

mBlock has been installed on your computer.

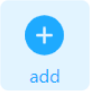
Click Finish to close Setup.

☑ Run mBlock

HOW TO ADD REKAEDUKIT

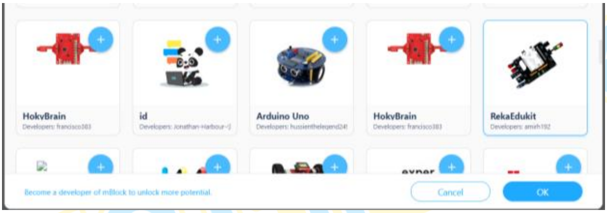
Step 1 Open mBlock v5

Step 2 Press the add button



.

Step 3 Select the RekaEduKit device and press ok.



HokyBrain
Developers: francisco383

id
Developers: Jonathan-Harbour-J

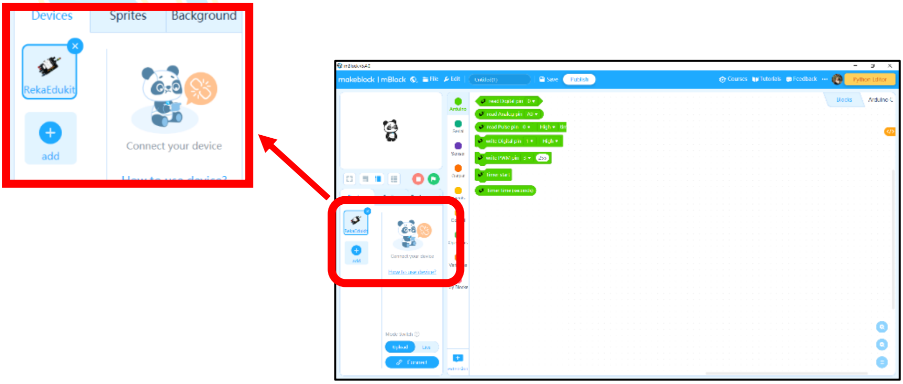
Arduino Uno
Developers: hussienthelegend24l

HokyBrain
Developers: francisco383

RekaEdukit
Developers: amirh192

Become a developer of mBlock to unlock more potential.

Step 4 Now you can enjoy using mBlock v5!



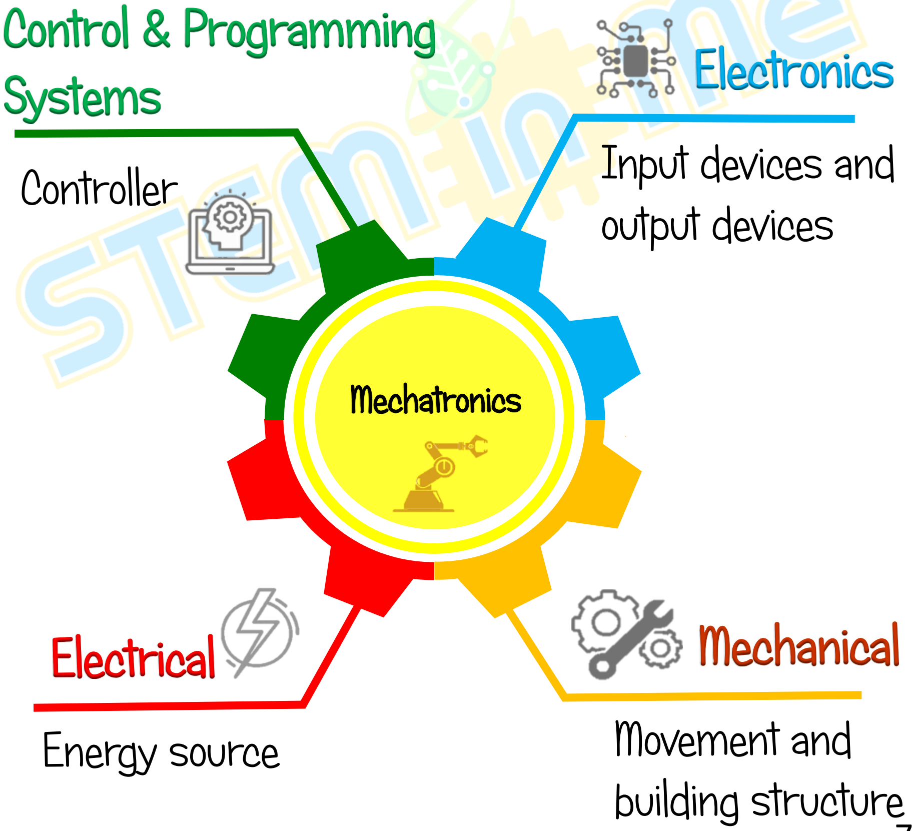
INTRODUCTION TO MECHATRONICS

Definition of Mechatronics

- ✓ Application of electronics and computer technology to control the movement of a product's mechanical system

Mechanism Electronics

Mechatronics



Control & Programming Systems

Controller

Electronics

Input devices and output devices

Mechatronics

Electrical

Energy source

Mechanical

Movement and building structure

1: COMMUNICATION CONTROL PUSH BUTTON

In this project we will display words on a serial monitor.

INTRODUCTION TO PUSH BUTTON

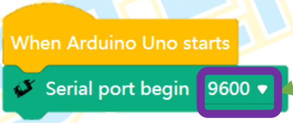
A push button is a mechanical device used to control an electrical circuit. Used to actuate the internal switching mechanism.

DEVICE USED :

1. RekaEdukit Circuit Board
2. Push Button

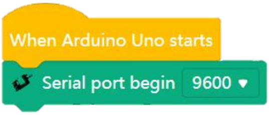
TUTORIAL

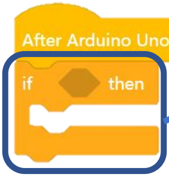
1. Drag the block When Arduino Uno starts along with block Serial port begin.



Set baud rate value to 9600.

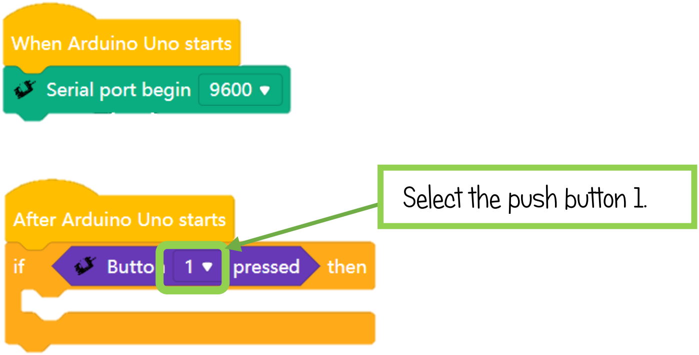
2. Drag the block After Arduino Uno starts along with block if, then.





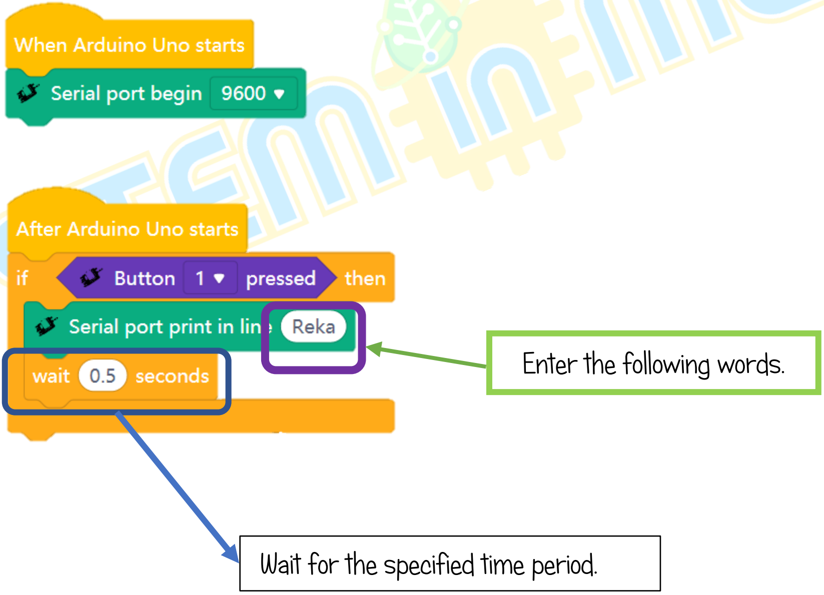
This block is executed if the statement that has been set is true.

3. Next, add block **Button pressed** into the hexagonal space inside the block **if, then**.



Select the push button 1.

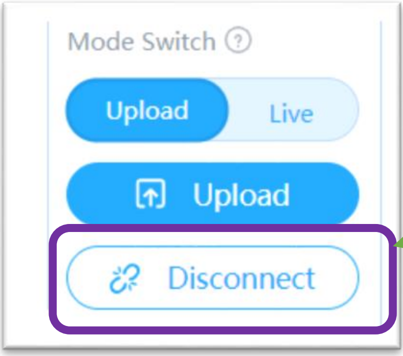
4. Add block **Serial port print in line Reka** in the block **if, then** followed by a block **delay**.



Enter the following words.

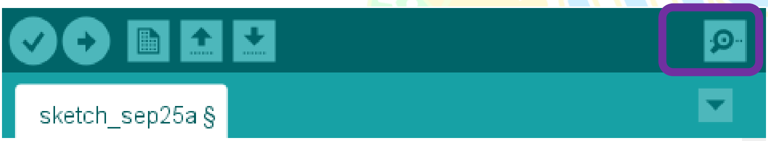
Wait for the specified time period.

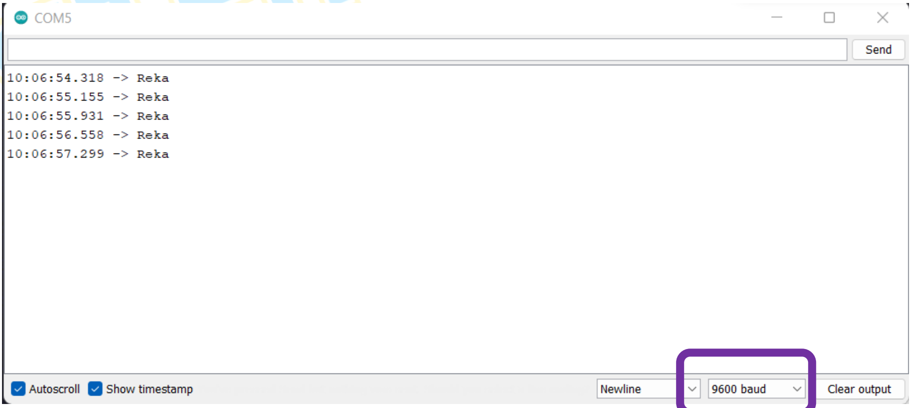
5. To see the output, disconnect the RekaEduKit board from MBlock.



Press the following icon.

6. Open the Arduino IDE software. Then, press the magnifying glass icon on the right. It will open the serial monitor and display the words.





```
10:06:54.318 -> Reka
10:06:55.155 -> Reka
10:06:55.931 -> Reka
10:06:56.558 -> Reka
10:06:57.299 -> Reka
```

Set the baud value to 9600 baud.

2: HOLIDAY LIGHTS

In this project we will light a single colour LED light (GYR LED) according to the sequence that has been set.

INTRODUCTION TO GYR LED

Single colour LED (GYR LED) is a mini traffic light display module that has high brightness, small size, and easy wiring. It can be connected to PWM to control the brightness of the LED.

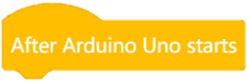
DEVICE USED:

1. RekaEdukit Circuit Board

2. Single colour LED (GYR LED)

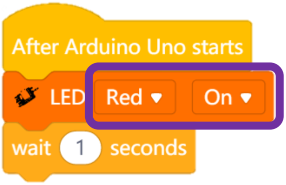
TUTORIAL

1. Drag the block After Arduino Uno starts .



```
After Arduino Uno starts
```

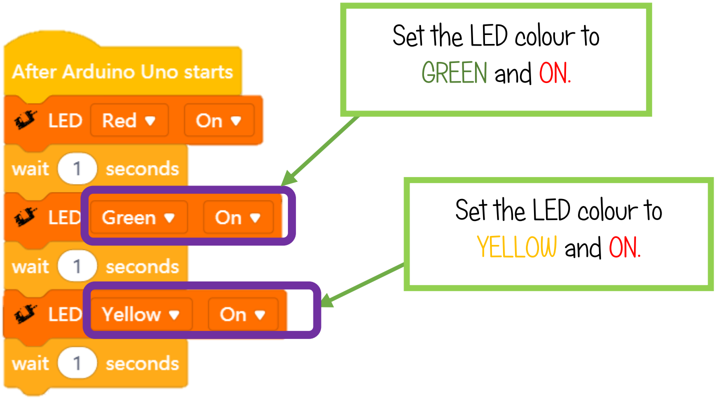
2. Insert the block LED along with block delay.



```
After Arduino Uno starts
LED Red ▾ On ▾
wait 1 seconds
```

Set the LED colour to RED and ON.

3. Repeat Step 2 for each LED colour.

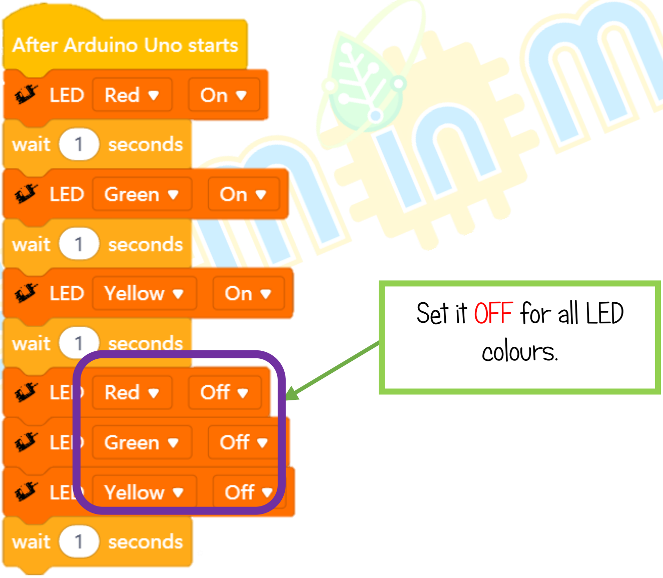


```
After Arduino Uno starts
LED Red On
wait 1 seconds
LED Green On
wait 1 seconds
LED Yellow On
wait 1 seconds
```

Set the LED colour to GREEN and ON.

Set the LED colour to YELLOW and ON.

4. Then, add 3 block LED consecutively and ending with a block delay.

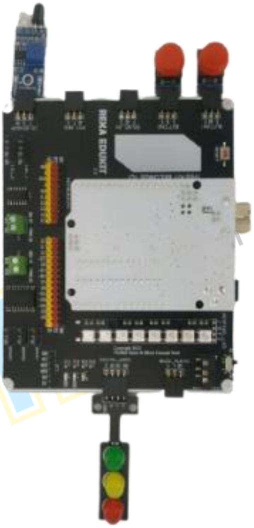


```
After Arduino Uno starts
LED Red On
wait 1 seconds
LED Green On
wait 1 seconds
LED Yellow On
wait 1 seconds
LED Red Off
LED Green Off
LED Yellow Off
wait 1 seconds
```

Set it OFF for all LED colours.

3: TRAFFIC SIGNAL LIGHTS

This project aims to practice the use of traffic lights. We can control the colour of single colour LED (GYR LED) one at a time. If the infrared sensor detects something, a green LED will light up. To change to yellow, we need to press push button 2. Finally, to change the LED to red, we need to press push button 1.

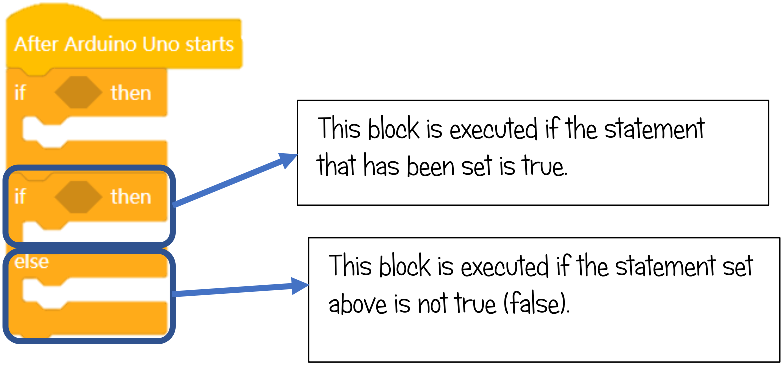


DEVICE USED:

1. RekaEdukit Circuit Board
2. Push Button
3. one colour LED (GYR LED)
4. Infrared Sensor

TUTORIAL

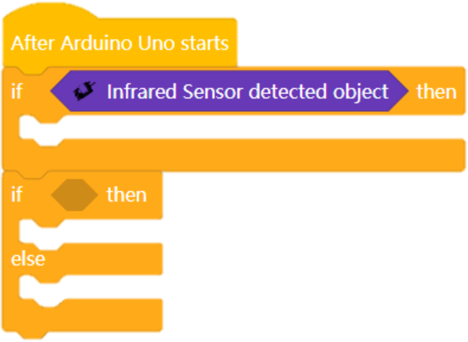
1. Drag After Arduino Uno starts along with block if, then and block if, then, else.



This block is executed if the statement that has been set is true.

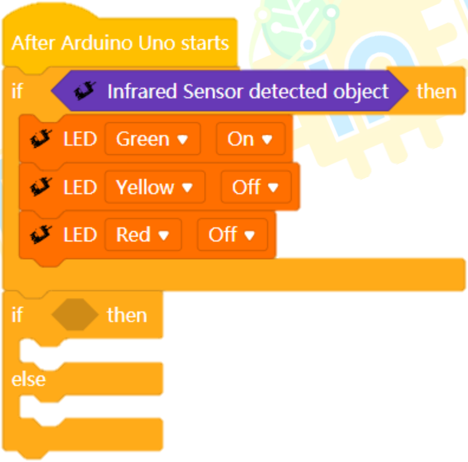
This block is executed if the statement set above is not true (false).

2. Insert the block Infrared Sensor detected object into the hexagonal space inside the block if, then.



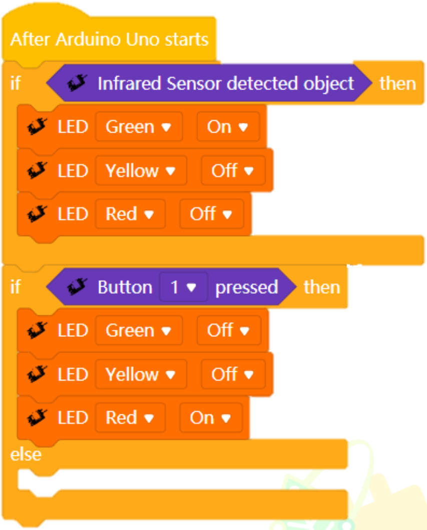
```
After Arduino Uno starts
if  Infrared Sensor detected object  then

if  then

else

```

3. Then, insert the block LED for each color Green, Yellow and Red.



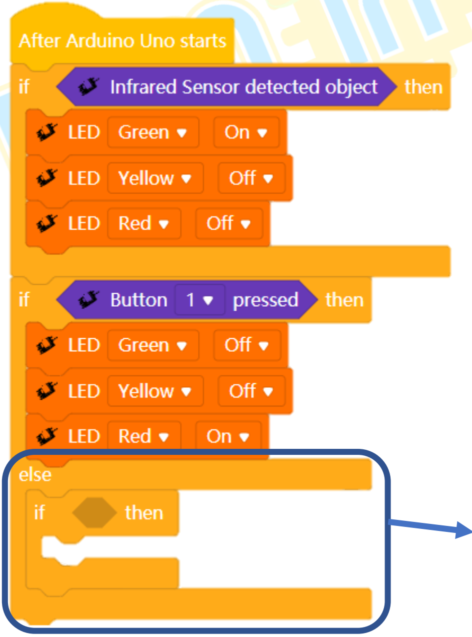
```
After Arduino Uno starts
if  Infrared Sensor detected object  then
    LED  Green ▾  On ▾
    LED  Yellow ▾  Off ▾
    LED  Red ▾  Off ▾
if  then

else

```

4. Drag and drop block button 1 pressed into the hexagonal space inside the block if, then, else.



```
After Arduino Uno starts
if  Infrared Sensor detected object  then
    LED  Green ▾  On ▾
    LED  Yellow ▾  Off ▾
    LED  Red ▾  Off ▾
if  Button  1 ▾  pressed  then
    LED  Green ▾  Off ▾
    LED  Yellow ▾  Off ▾
    LED  Red ▾  On ▾
else
```

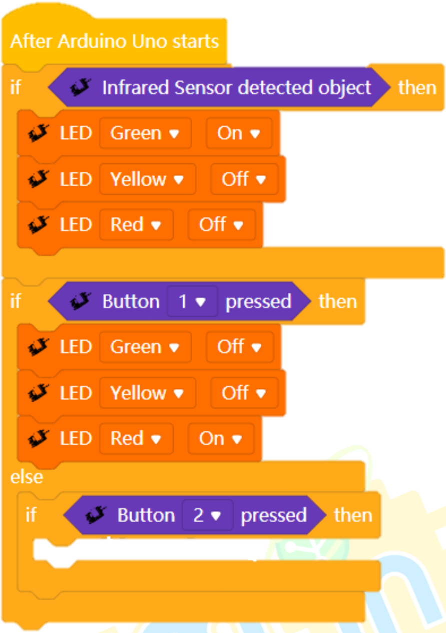
5. Drag and drop blocks if, then into the block else.



```
After Arduino Uno starts
if  Infrared Sensor detected object  then
    LED  Green ▾  On ▾
    LED  Yellow ▾  Off ▾
    LED  Red ▾  Off ▾
if  Button  1 ▾  pressed  then
    LED  Green ▾  Off ▾
    LED  Yellow ▾  Off ▾
    LED  Red ▾  On ▾
else
    if  then
```

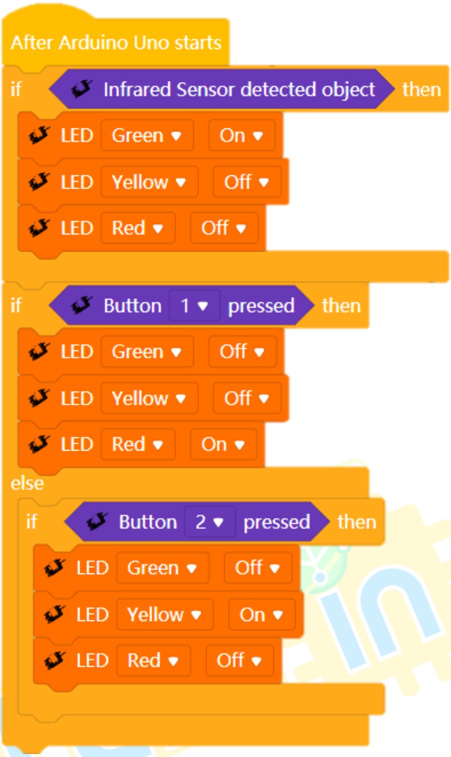
This block is executed if the statement set above is not true (false) and if the statement set here is true (true).

6. Insert the block button 2 pressed into the hexagonal space inside the block if, then.



```
After Arduino Uno starts
if  Infrared Sensor detected object  then
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾
if  Button 1 ▾ pressed  then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
else
    if  Button 2 ▾ pressed  then
```

7. Add block LED for each colour Green, Yellow and Red.



```
After Arduino Uno starts
if  Infrared Sensor detected object  then
    LED  Green ▾  On ▾
    LED  Yellow ▾  Off ▾
    LED  Red ▾  Off ▾
if  Button  1 ▾  pressed  then
    LED  Green ▾  Off ▾
    LED  Yellow ▾  Off ▾
    LED  Red ▾  On ▾
else
    if  Button  2 ▾  pressed  then
        LED  Green ▾  Off ▾
        LED  Yellow ▾  On ▾
        LED  Red ▾  Off ▾
```

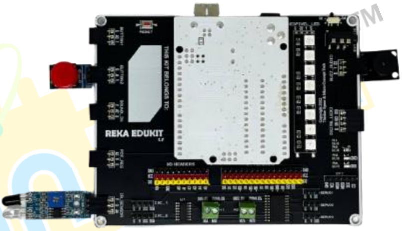
4: ALARM SYSTEM

In this project we will build an alarm system. The alarm will sound if the infrared sensor (IR sensor) detects an object in front. Next, to stop the alarm sound, we need to press push button 1.

INTRODUCTION TO INFRARED SENSORS

An infrared sensor (IR Sensor) is used to detect objects in front. The distance of the detected object is adjustable.

DEVICE USED :



1. RekaEdukit Circuit Board

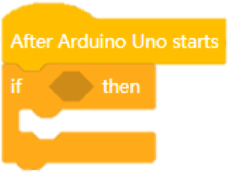
2. Push Button

3. Infrared Sensor

4. Buzzer

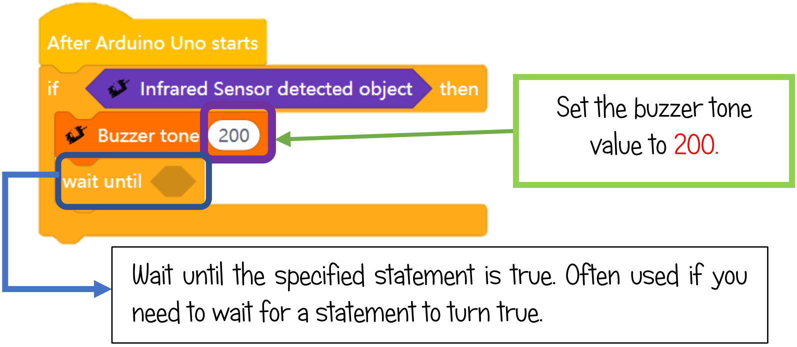
TUTORIAL

1. Drag After Arduino Uno starts along with block forever and block if, then.



After Arduino Uno starts
if then

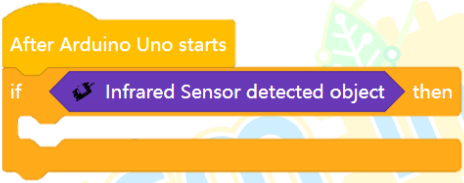
2. Next, insert the block Infrared sensor detect object into the hexagonal space inside the block if, then.


After Arduino Uno starts
if Infrared Sensor detected object then
Buzzer tone 200
wait until

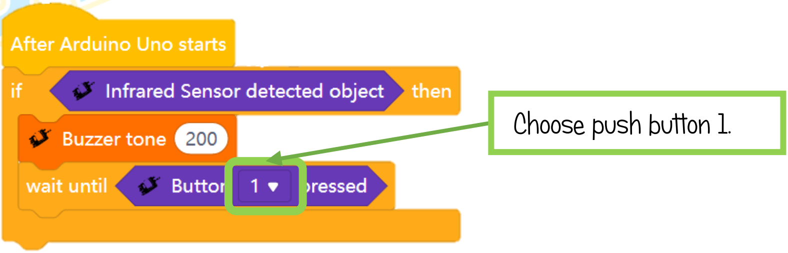
Set the buzzer tone value to 200.

Wait until the specified statement is true. Often used if you need to wait for a statement to turn true.

3. 3. Then, add buzzer tone along with block wait until in the block if, then.

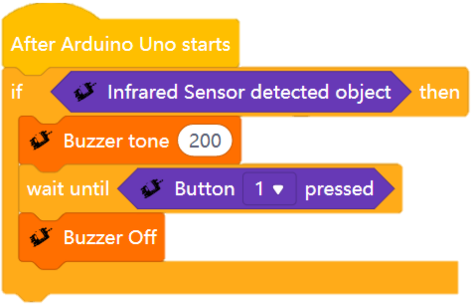

After Arduino Uno starts
if Infrared Sensor detected object then

4. Drag the block button pressed into the hexagonal space inside the block wait until.


After Arduino Uno starts
if Infrared Sensor detected object then
Buzzer tone 200
wait until Button 1 ▾ pressed

Choose push button 1.

5. Add the block Buzzer Off under the block wait until.


After Arduino Uno starts
if Infrared Sensor detected object then
Buzzer tone 200
wait until Button 1 ▾ pressed
Buzzer Off

5: FESTIVAL OF LIGHTS

In this project, we will use a potentiometer to turn on a multi-coloured LED (neopixel LED).

INTRODUCTION TO POTENTIOMETER

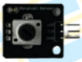
The potentiometer acts as an adjustable voltage divider. Potentiometers are changed manually to control the flow of electric current.

INTRODUCTION OF MULTI-COLOR LED (NEOPIXEL LED)

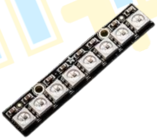
Each multi-color LED (Neopixel LED) is controlled by an integrated circuit that processes information and converts it into data to control the light.

DEVICE USED :

1. RekaEdukit Circuit Board
2. Potentiometer

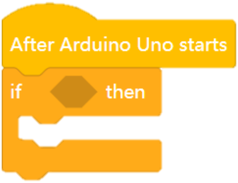


3. Multi-coloured LED (Neopixel LED)

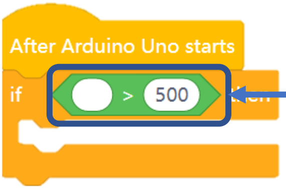


TUTORIAL

1. Drag the block After Arduino Uno starts along with block if, then.



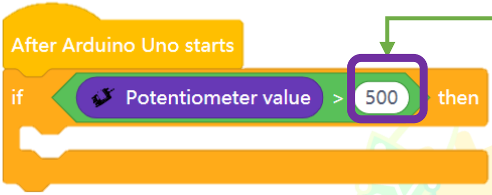
2. Add block greater than into the hexagonal space inside the block if, then.



```
After Arduino Uno starts
if ( ) > 500 then
```

This block is used to compare the left side value and the right side value. If the left value is greater than the right value, this statement evaluates to true.

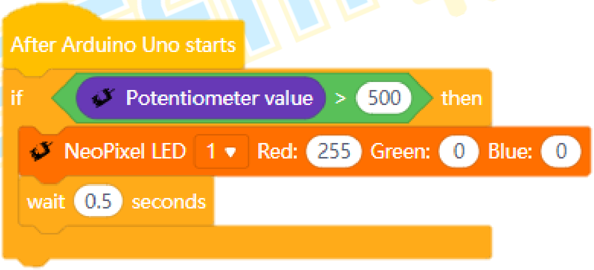
3. Then, add block potential value into the block greater than.



```
After Arduino Uno starts
if Potentiometer value > 500 then
```

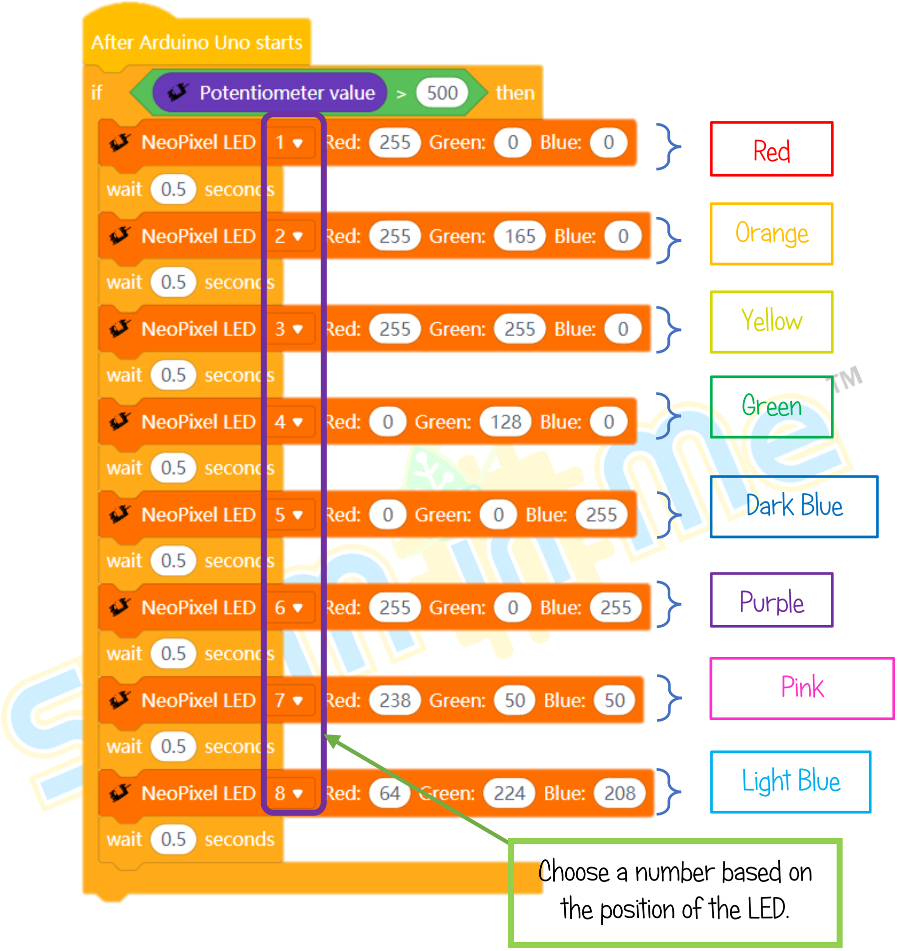
Tetapkan nilai kepada 500.

4. Next, drag and insert the block Neopixel LED together with block delay into the block if, then.



```
After Arduino Uno starts
if Potentiometer value > 500 then
    NeoPixel LED 1 ▾ Red: 255 Green: 0 Blue: 0
    wait 0.5 seconds
```

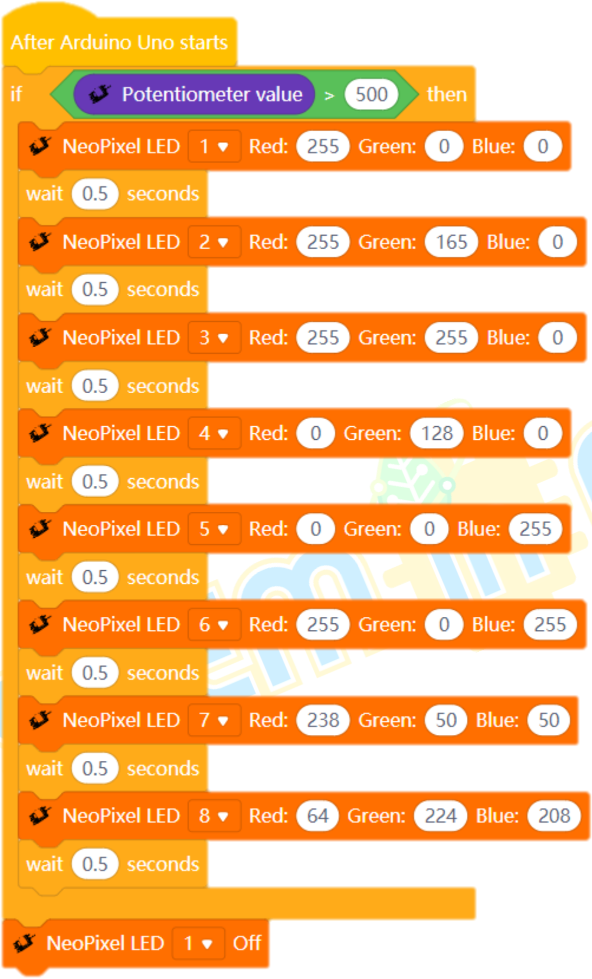
5. 5. Repeat Step 4 for each 8blok Neopixel LED.



```
After Arduino Uno starts
if  Potentiometer value > 500  then
    NeoPixel LED 1 ▾  Red: 255  Green: 0  Blue: 0
    wait 0.5 seconds
    NeoPixel LED 2 ▾  Red: 255  Green: 165  Blue: 0
    wait 0.5 seconds
    NeoPixel LED 3 ▾  Red: 255  Green: 255  Blue: 0
    wait 0.5 seconds
    NeoPixel LED 4 ▾  Red: 0  Green: 128  Blue: 0
    wait 0.5 seconds
    NeoPixel LED 5 ▾  Red: 0  Green: 0  Blue: 255
    wait 0.5 seconds
    NeoPixel LED 6 ▾  Red: 255  Green: 0  Blue: 255
    wait 0.5 seconds
    NeoPixel LED 7 ▾  Red: 238  Green: 50  Blue: 50
    wait 0.5 seconds
    NeoPixel LED 8 ▾  Red: 64  Green: 224  Blue: 208
    wait 0.5 seconds
```

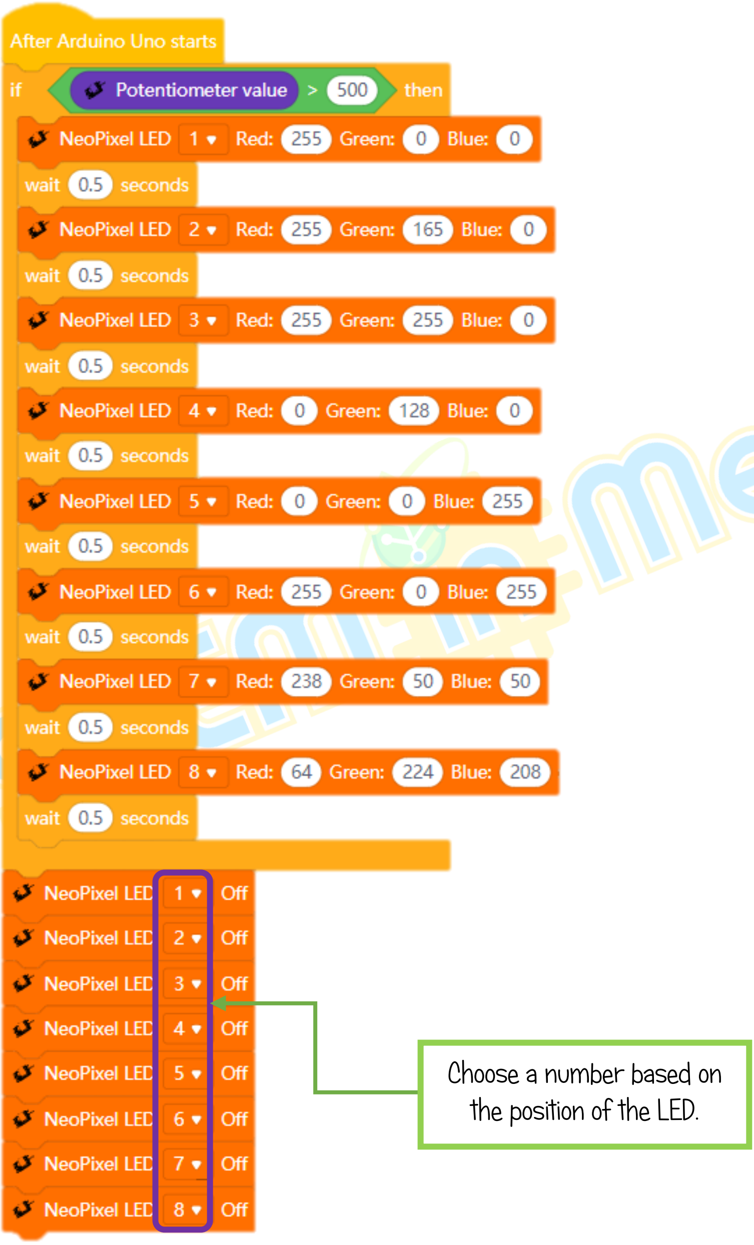
Choose a number based on the position of the LED.

6. Drag block Neopixel LED off under the block if, then.



```
After Arduino Uno starts
if  Potentiometer value > 500  then
    NeoPixel LED 1 ▾ Red: 255 Green: 0 Blue: 0
    wait 0.5 seconds
    NeoPixel LED 2 ▾ Red: 255 Green: 165 Blue: 0
    wait 0.5 seconds
    NeoPixel LED 3 ▾ Red: 255 Green: 255 Blue: 0
    wait 0.5 seconds
    NeoPixel LED 4 ▾ Red: 0 Green: 128 Blue: 0
    wait 0.5 seconds
    NeoPixel LED 5 ▾ Red: 0 Green: 0 Blue: 255
    wait 0.5 seconds
    NeoPixel LED 6 ▾ Red: 255 Green: 0 Blue: 255
    wait 0.5 seconds
    NeoPixel LED 7 ▾ Red: 238 Green: 50 Blue: 50
    wait 0.5 seconds
    NeoPixel LED 8 ▾ Red: 64 Green: 224 Blue: 208
    wait 0.5 seconds
NeoPixel LED 1 ▾ Off
```

7. Repeat Step 6 for each of the 8 Neopixel LEDs.



```
After Arduino Uno starts
if  Potentiometer value > 500  then
    NeoPixel LED 1 ▾ Red: 255 Green: 0 Blue: 0
    wait 0.5 seconds
    NeoPixel LED 2 ▾ Red: 255 Green: 165 Blue: 0
    wait 0.5 seconds
    NeoPixel LED 3 ▾ Red: 255 Green: 255 Blue: 0
    wait 0.5 seconds
    NeoPixel LED 4 ▾ Red: 0 Green: 128 Blue: 0
    wait 0.5 seconds
    NeoPixel LED 5 ▾ Red: 0 Green: 0 Blue: 255
    wait 0.5 seconds
    NeoPixel LED 6 ▾ Red: 255 Green: 0 Blue: 255
    wait 0.5 seconds
    NeoPixel LED 7 ▾ Red: 238 Green: 50 Blue: 50
    wait 0.5 seconds
    NeoPixel LED 8 ▾ Red: 64 Green: 224 Blue: 208
    wait 0.5 seconds
NeoPixel LED 1 ▾ Off
NeoPixel LED 2 ▾ Off
NeoPixel LED 3 ▾ Off
NeoPixel LED 4 ▾ Off
NeoPixel LED 5 ▾ Off
NeoPixel LED 6 ▾ Off
NeoPixel LED 7 ▾ Off
NeoPixel LED 8 ▾ Off
```

Choose a number based on the position of the LED.

6: PEDESTRIAN SYSTEM

We will build a pedestrian system using the Devices listed below. In this system, a single colour LED will light RED. When, the infrared sensor detects something and the user presses the push button at the same time, the buzzer will sound and the single colour LED will light GREEN.

INTRODUCTION TO BUZZER

A buzzer is a sound signalling device that can convert audio signals into sound signals.

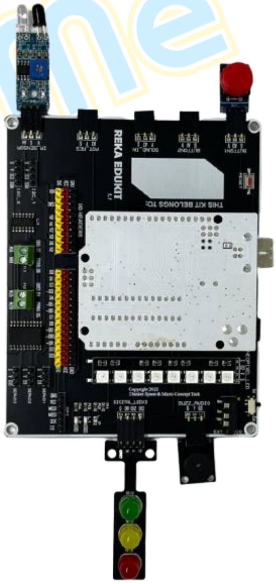
DEVICE USED :

1. RekaEdukit Circuit Board
2. Single color LED (GYR LED)

3. Push Button

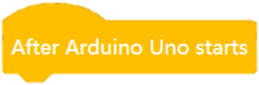
4. Infrared Sensor

4. Buzzer



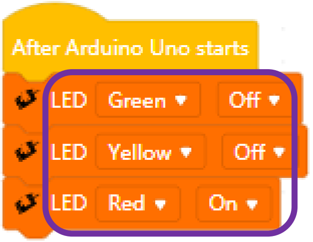
TUTORIAL

1. Drag the block After Arduino Uno starts.



After Arduino Uno starts

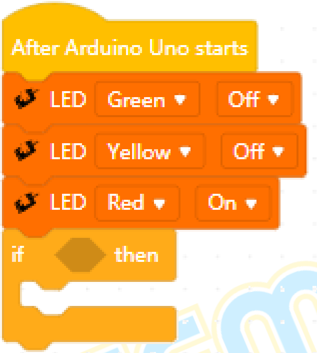
2. Insert 3 block LED for each LED colour.



After Arduino Uno starts
LED Green ▾ Off ▾
LED Yellow ▾ Off ▾
LED Red ▾ On ▾

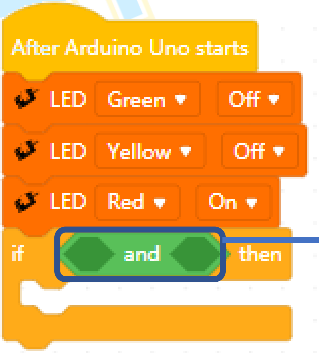
Select the LED colors in the following order. Then, choose ON for RED LED.

3. Next, drag and drop the block if, then under the block LED.



After Arduino Uno starts
LED Green ▾ Off ▾
LED Yellow ▾ Off ▾
LED Red ▾ On ▾
if then

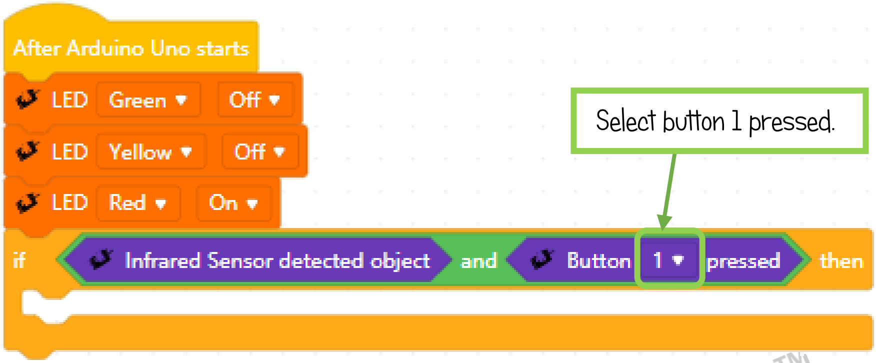
4. Insert the block and into the hexagonal space inside the block if, then.



After Arduino Uno starts
LED Green ▾ Off ▾
LED Yellow ▾ Off ▾
LED Red ▾ On ▾
if and then

This block is used if you want both statements set to be true.

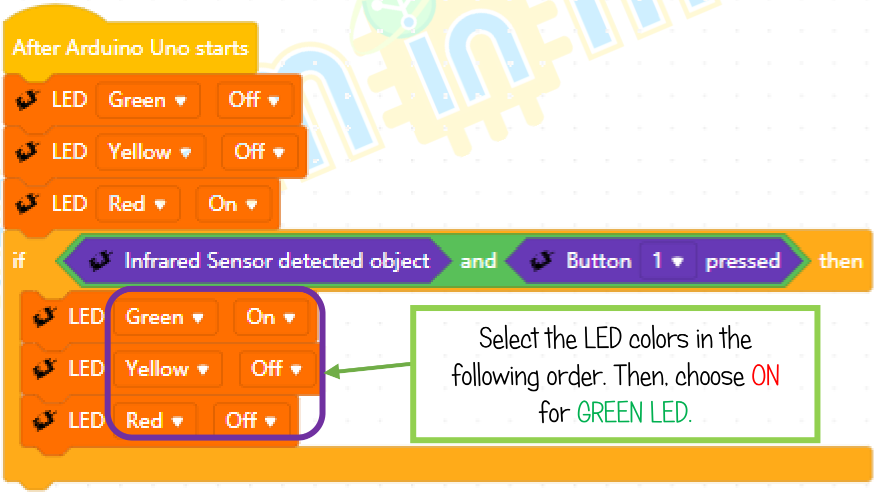
5. 5. Insert the block Infrared sensor detected object and block button pressed into the block and.



```
After Arduino Uno starts
LED Green ▾ Off ▾
LED Yellow ▾ Off ▾
LED Red ▾ On ▾
if Infrared Sensor detected object and Button 1 ▾ pressed then
```

Select button 1 pressed.

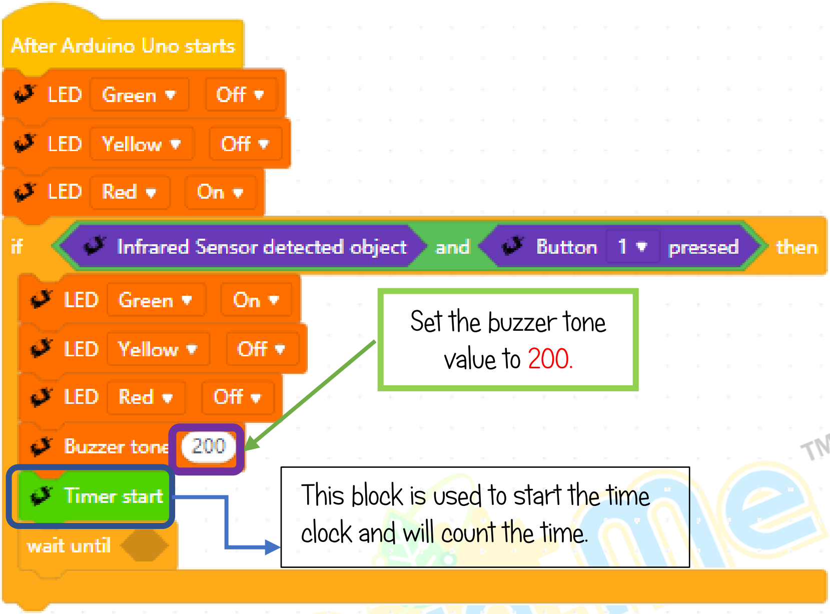
6. Then, insert the block LED for each colour in the block if, then.



```
After Arduino Uno starts
LED Green ▾ Off ▾
LED Yellow ▾ Off ▾
LED Red ▾ On ▾
if Infrared Sensor detected object and Button 1 ▾ pressed then
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾
```

Select the LED colors in the following order. Then, choose ON for GREEN LED.

7. Next, insert the block buzzer tone, timer start and wait until.

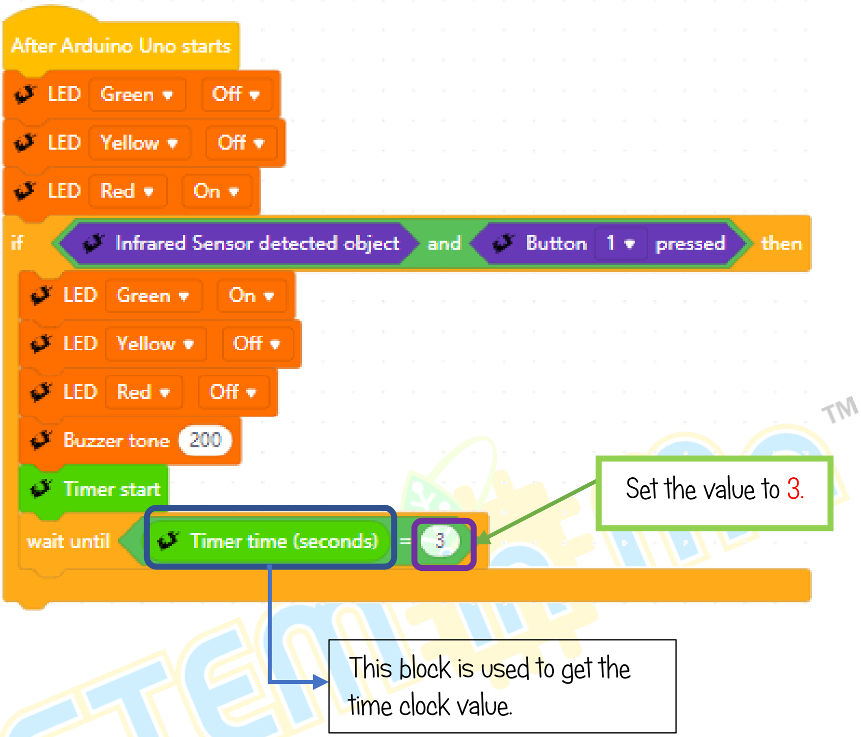


```
After Arduino Uno starts
  LED Green ▾ Off ▾
  LED Yellow ▾ Off ▾
  LED Red ▾ On ▾
  if Infrared Sensor detected object and Button 1 ▾ pressed then
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾
    Buzzer tone 200
    Timer start
    wait until
```

Set the buzzer tone value to 200.

This block is used to start the time clock and will count the time.

8. Insert block timer time inside the block equal then insert it into the hexagonal space in the block wait until.



After Arduino Uno starts
LED Green Off
LED Yellow Off
LED Red On
if Infrared Sensor detected object and Button 1 pressed then
  LED Green On
  LED Yellow Off
  LED Red Off
  Buzzer tone 200
  Timer start
  wait until Timer time (seconds) = 3

Set the value to 3.

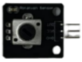
This block is used to get the time clock value.

7: THE NEW MILLENNIUM MUSIC INSTRUMENTS

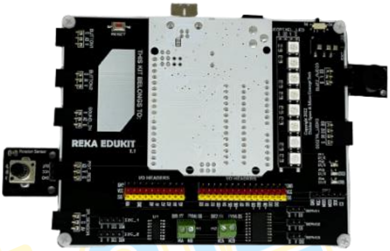
In this project, we will build a new millennium musical instrument. The buzzer will sound with a different tone when the power meter is turned to the set value.

DEVICE USED :

1. RekaEdukit Circuit Board

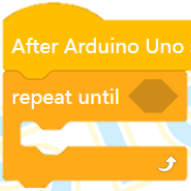
2. Potentiometer 

3. Buzzer

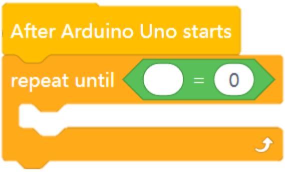


TUTORIAL

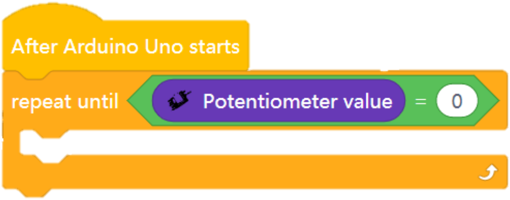
1. Drag After Arduino Uno starts followed by block repeat until.



2. Add block Equal into the hexagonal space inside the block repeat until.

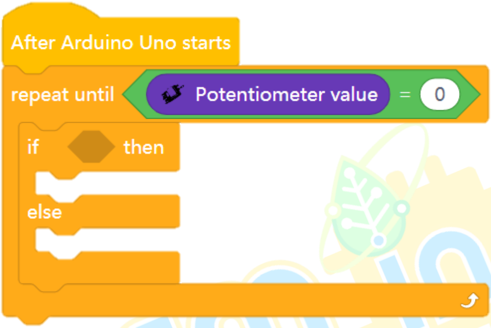


3. Insert block potentiometer value into the block equal.



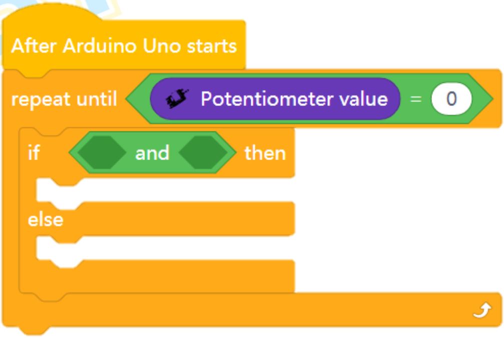
```
After Arduino Uno starts
repeat until < Potentiometer value = 0 >
```

4. Next, insert the block if, then, else into the block repeat until.



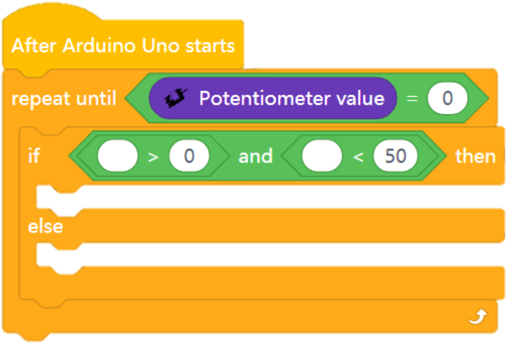
```
After Arduino Uno starts
repeat until < Potentiometer value = 0 >
    if < > then
    else
```

5. Insert the block and into the hexagonal space inside the block if, then.



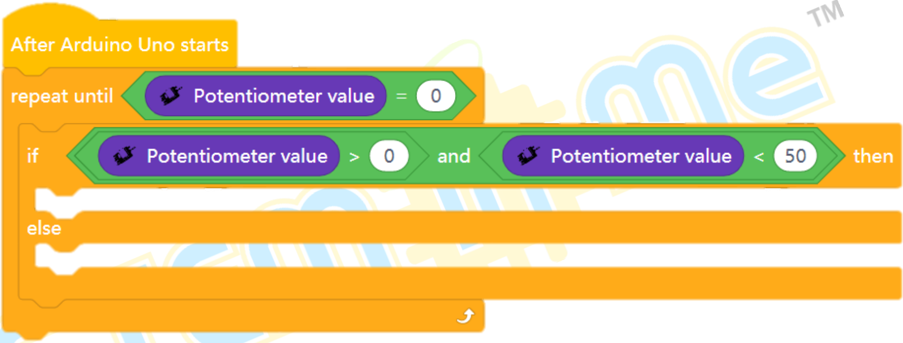
```
After Arduino Uno starts
repeat until < Potentiometer value = 0 >
    if < < > and < > > then
    else
```

6. Insert the block greater than dan less than into the block and.



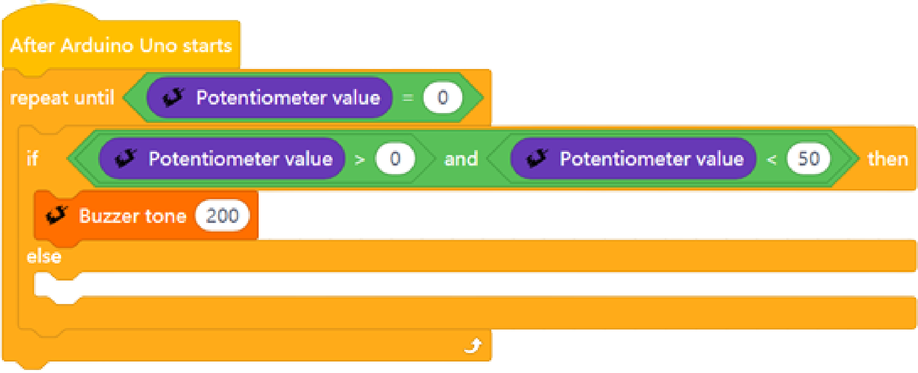
```
After Arduino Uno starts
repeat until Potentiometer value = 0
    if ( > 0) and ( < 50) then

    else

```

7. Next, insert the block potentiometer value into two blocks greater than and less than.



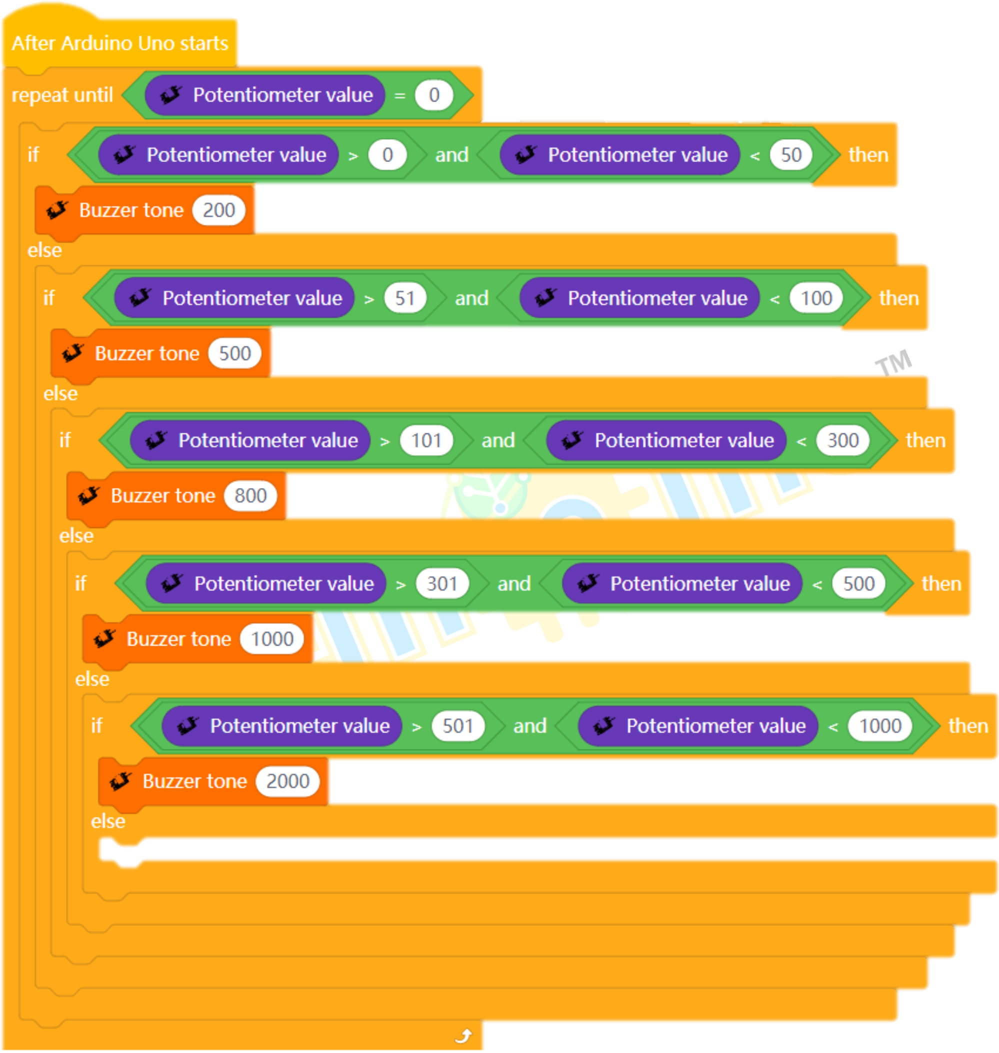
```
After Arduino Uno starts
repeat until Potentiometer value = 0
    if (Potentiometer value > 0) and (Potentiometer value < 50) then

    else

```

8. Drag and drop blocks buzzer tone into the block if, then.



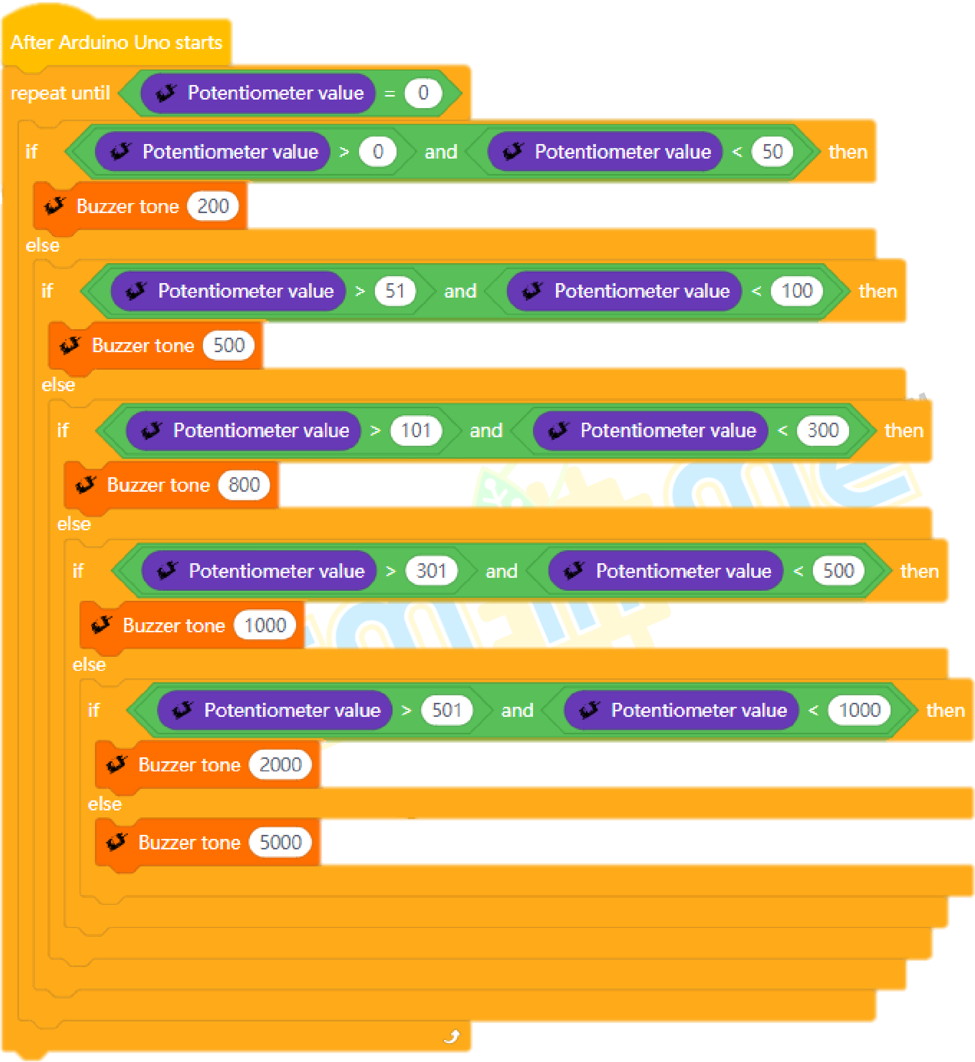
```
After Arduino Uno starts
repeat until Potentiometer value = 0
    if (Potentiometer value > 0) and (Potentiometer value < 50) then
        Buzzer tone 200
    else

```

9. Repeat Step 4 to Step 8 with a different power meter value and buzzer tone value, put in a block for each block else.



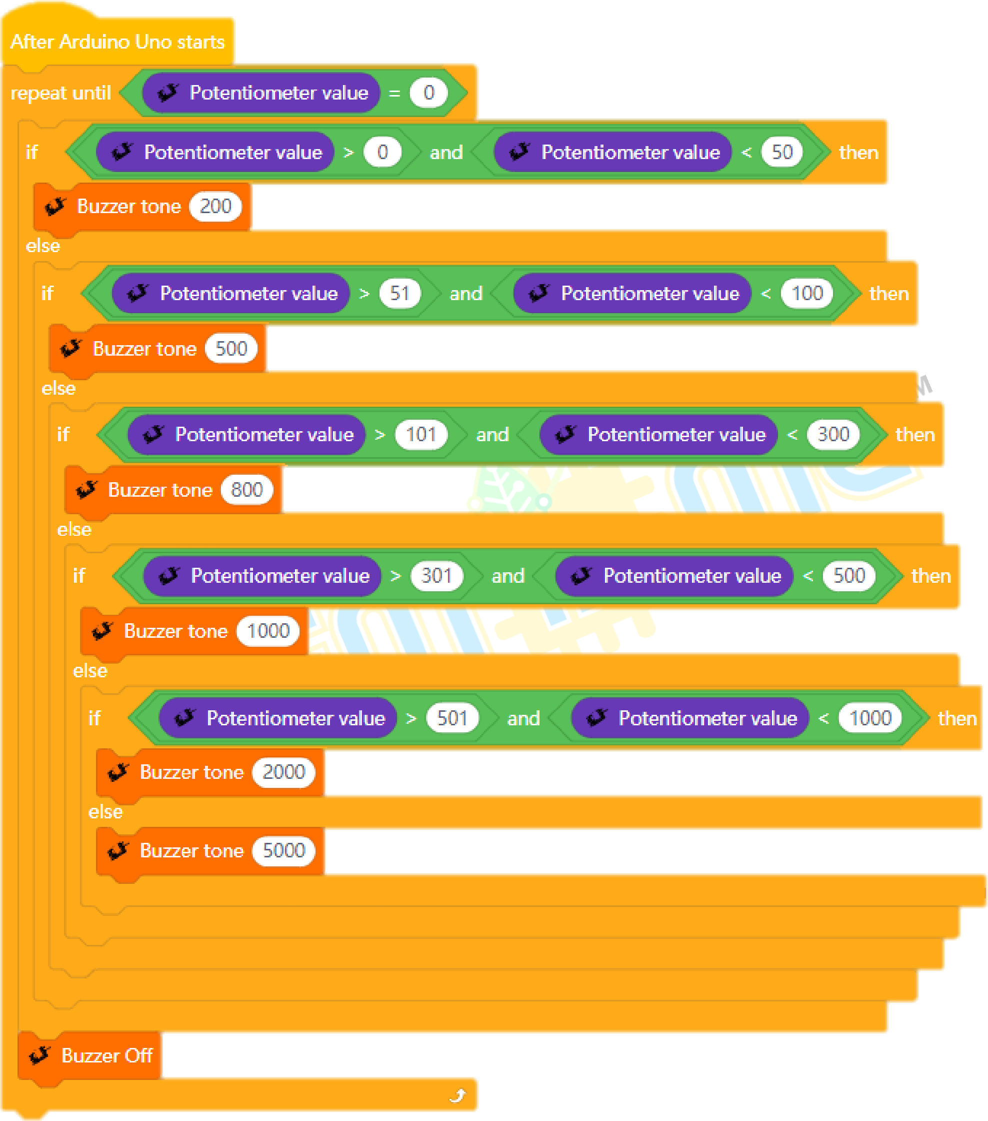
```
After Arduino Uno starts
repeat until Potentiometer value = 0
    if Potentiometer value > 0 and Potentiometer value < 50 then
        Buzzer tone 200
    else
        if Potentiometer value > 51 and Potentiometer value < 100 then
            Buzzer tone 500
        else
            if Potentiometer value > 101 and Potentiometer value < 300 then
                Buzzer tone 800
            else
                if Potentiometer value > 301 and Potentiometer value < 500 then
                    Buzzer tone 1000
                else
                    if Potentiometer value > 501 and Potentiometer value < 1000 then
                        Buzzer tone 2000
                    else
```

10. Insert the block buzzer tone into the last block else.



```
After Arduino Uno starts
repeat until  Potentiometer value = 0
  if  Potentiometer value > 0  and  Potentiometer value < 50  then
    Buzzer tone 200
  else
    if  Potentiometer value > 51  and  Potentiometer value < 100  then
      Buzzer tone 500
    else
      if  Potentiometer value > 101  and  Potentiometer value < 300  then
        Buzzer tone 800
      else
        if  Potentiometer value > 301  and  Potentiometer value < 500  then
          Buzzer tone 1000
        else
          if  Potentiometer value > 501  and  Potentiometer value < 1000  then
            Buzzer tone 2000
          else
            Buzzer tone 5000
```

11. Drag and drop block buzzer off into the block repeat until.



```
After Arduino Uno starts
repeat until  Potentiometer value = 0
  if  Potentiometer value > 0  and  Potentiometer value < 50  then
    Buzzer tone 200
  else
    if  Potentiometer value > 51  and  Potentiometer value < 100  then
      Buzzer tone 500
    else
      if  Potentiometer value > 101  and  Potentiometer value < 300  then
        Buzzer tone 800
      else
        if  Potentiometer value > 301  and  Potentiometer value < 500  then
          Buzzer tone 1000
        else
          if  Potentiometer value > 501  and  Potentiometer value < 1000  then
            Buzzer tone 2000
          else
            Buzzer tone 5000
  Buzzer Off
```

8: MULTI COLOR LED SYSTEM

We will build a project that can light multi-coloured LEDs (neopixel LEDs) using sound.

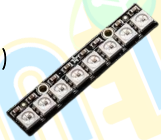
INTRODUCTION TO SOUND SENSOR

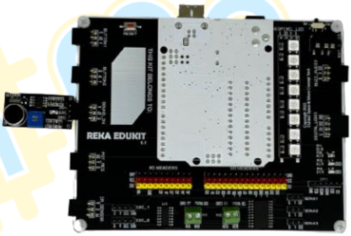
A sound sensor (sound sensor) converts vibrations into audio signals with the help of a microphone. It acts just like the human ear when sound is detected. The strength of the detected sound can be adjusted.

DEVICE USED :

1. Sound Sensor

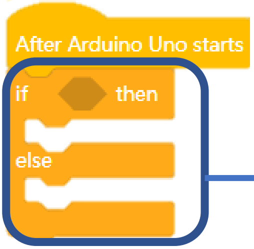
2. Multi-coloured LED (Neopixel LED)





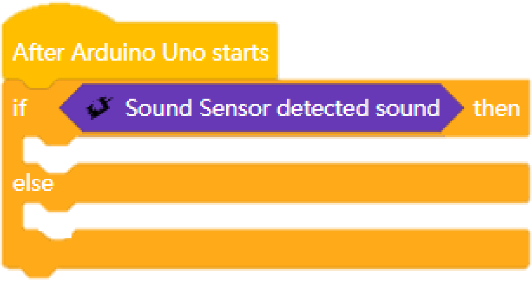
TUTORIAL

1. Drag After Arduino Uno starts along with block forever and block if, then, else.



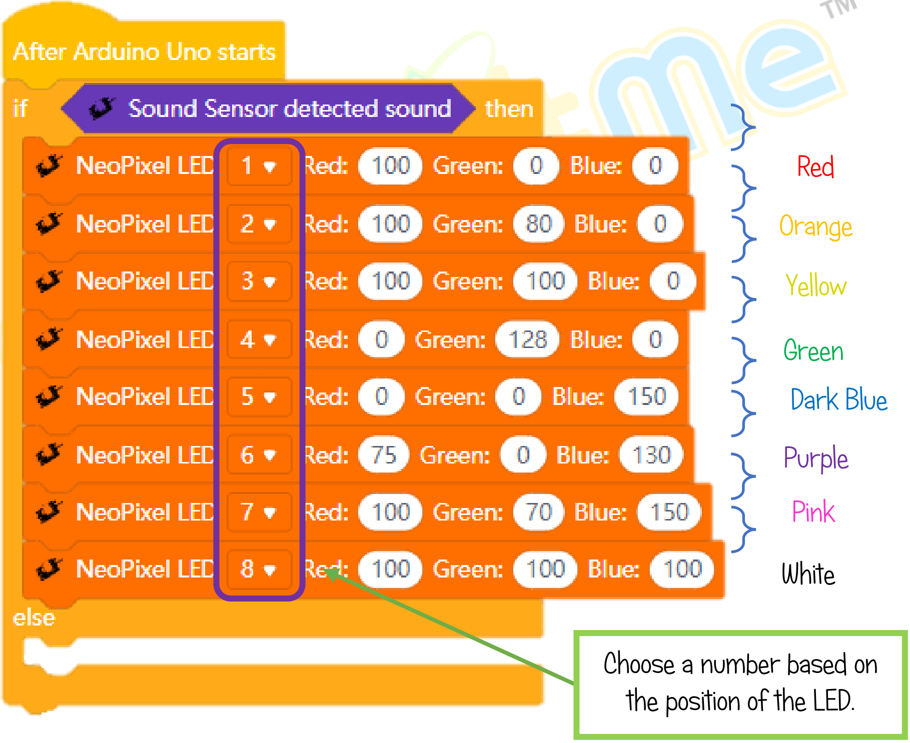
This block will check the statement inside the 'if' and will execute the statement block if the statement is true. But if the statement is not true (false) then this block will execute the 'else' block.

2. Insert the block sound sensor detect sound into the hexagonal space inside the block if, then.



```
After Arduino Uno starts
if  Sound Sensor detected sound  then

else

```

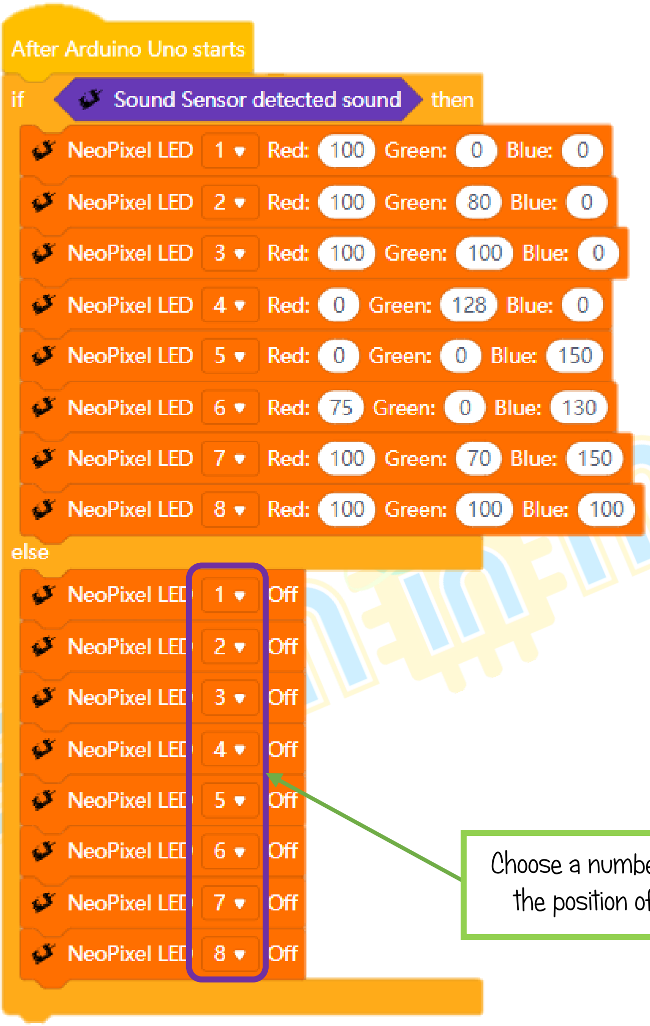
3. Add the block Neopixel LED to set 8 colors to all Neopixel LEDs under the block if, then.



```
After Arduino Uno starts
if  Sound Sensor detected sound  then
    NeoPixel LED 1 ▾ Red: 100 Green: 0 Blue: 0
    NeoPixel LED 2 ▾ Red: 100 Green: 80 Blue: 0
    NeoPixel LED 3 ▾ Red: 100 Green: 100 Blue: 0
    NeoPixel LED 4 ▾ Red: 0 Green: 128 Blue: 0
    NeoPixel LED 5 ▾ Red: 0 Green: 0 Blue: 150
    NeoPixel LED 6 ▾ Red: 75 Green: 0 Blue: 130
    NeoPixel LED 7 ▾ Red: 100 Green: 70 Blue: 150
    NeoPixel LED 8 ▾ Red: 100 Green: 100 Blue: 100
else

```

Choose a number based on the position of the LED.

4. Next, add the block Neopixel LED off to turn off all Neopixel LEDs.

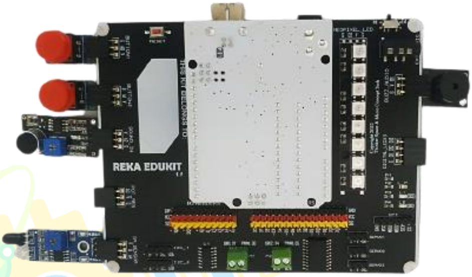


```
After Arduino Uno starts
if  Sound Sensor detected sound  then
    NeoPixel LED 1 ▾ Red: 100 Green: 0 Blue: 0
    NeoPixel LED 2 ▾ Red: 100 Green: 80 Blue: 0
    NeoPixel LED 3 ▾ Red: 100 Green: 100 Blue: 0
    NeoPixel LED 4 ▾ Red: 0 Green: 128 Blue: 0
    NeoPixel LED 5 ▾ Red: 0 Green: 0 Blue: 150
    NeoPixel LED 6 ▾ Red: 75 Green: 0 Blue: 130
    NeoPixel LED 7 ▾ Red: 100 Green: 70 Blue: 150
    NeoPixel LED 8 ▾ Red: 100 Green: 100 Blue: 100
else
    NeoPixel LED 1 ▾ Off
    NeoPixel LED 2 ▾ Off
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Off
```

Choose a number based on the position of the LED.

9: SECURITY SYSTEM

Ever known how security systems are built? Let's take a look at the project below. In this system, if the infrared sensor or the sound sensor detects a sound or object, the buzzer will sound and the multi-colored LEDs will light up one by one to warn that there is a threat of danger. To stop the buzzer and multi-colored LED, we need to press both push buttons at the same time.



DEVICE USED :

1. RekaEdukit Circuit Board

2. Push Button

3. Infrared Sensor

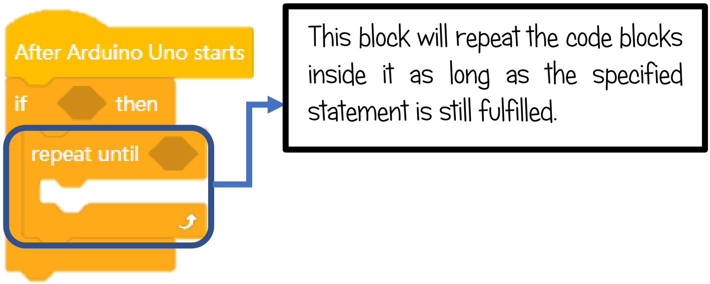
4. Sound Sensor

5. Multi-coloured LED (Neopixel LED)

6. Buzzer

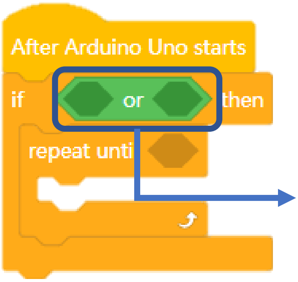
TUTORIAL

1. Drag After Arduino Uno starts along with block if, then and block repeat until.



This block will repeat the code blocks inside it as long as the specified statement is still fulfilled.

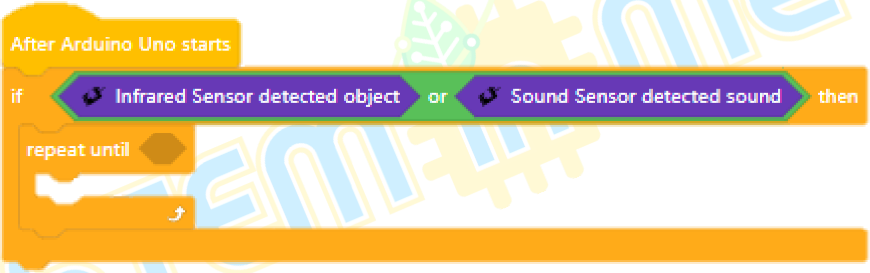
2. Enter the block or into the hexagonal space inside the block if, then.



```
After Arduino Uno starts
if < or > then
  repeat until < >
```

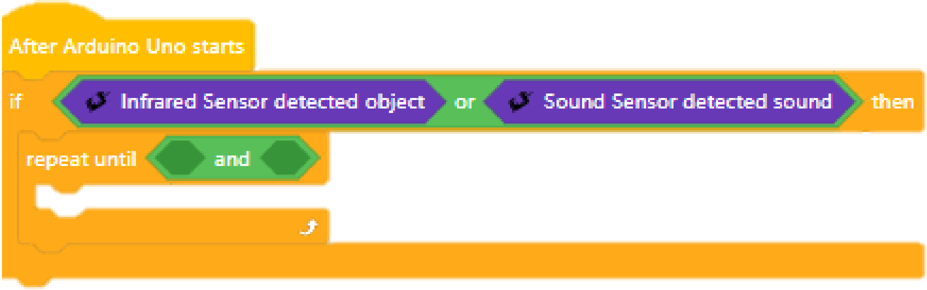
This block is used if you want one of the statements to be true.

3. Then, insert the block Infrared sensor detected object and block sound sensor detected sound into or block.



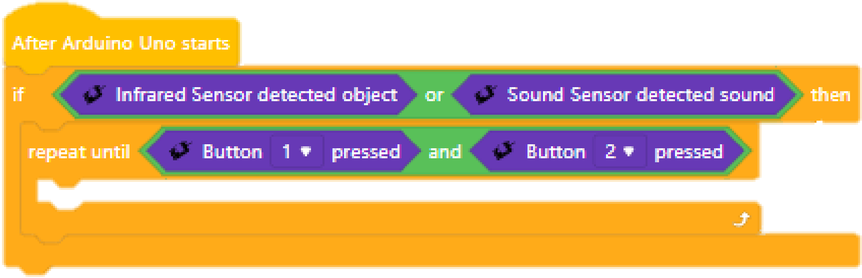
```
After Arduino Uno starts
if <Infrared Sensor detected object or Sound Sensor detected sound> then
  repeat until < >
```

4. Add blocks and into the hexagonal space inside the block repeat until.



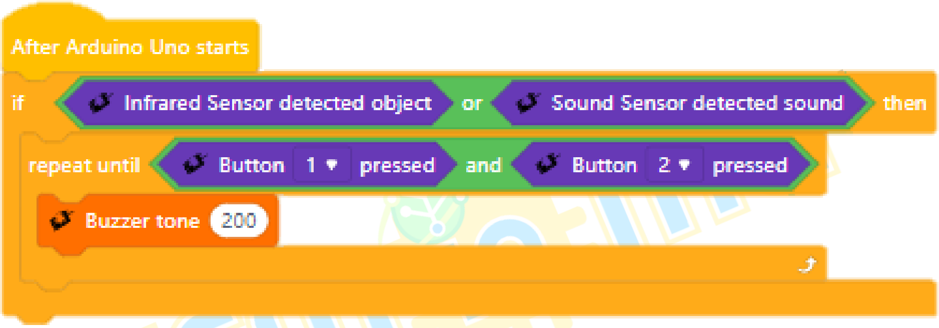
```
After Arduino Uno starts
if <Infrared Sensor detected object or Sound Sensor detected sound> then
  repeat until < and >
```

5. Next, add the block button 1 pressed and button 2 pressed into the block and.



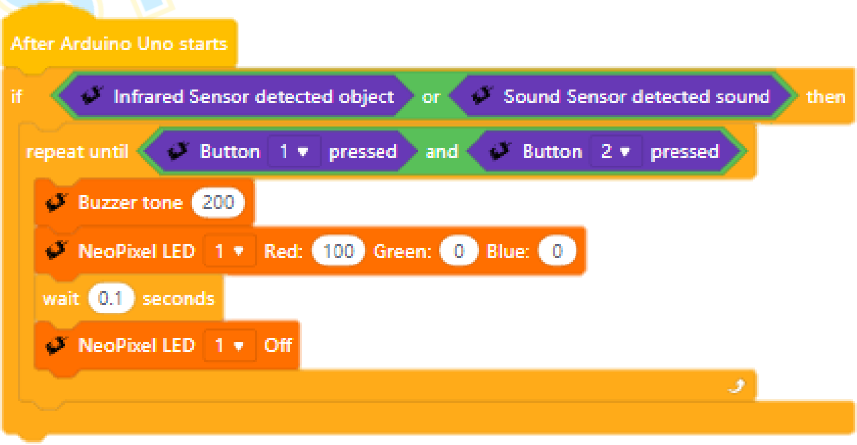
```
After Arduino Uno starts
if Infrared Sensor detected object or Sound Sensor detected sound then
    repeat until Button 1 ▾ pressed and Button 2 ▾ pressed
```

6. Add block buzzer tone **under the block** repeat until.



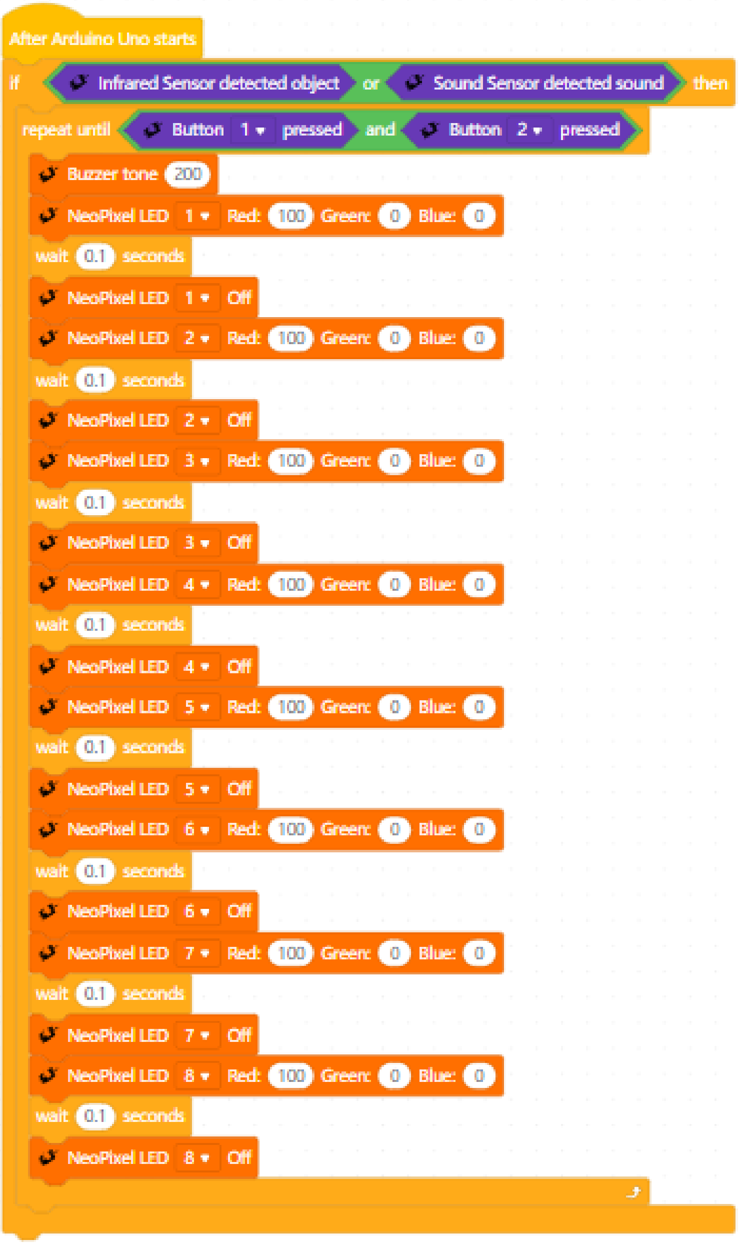
```
After Arduino Uno starts
if Infrared Sensor detected object or Sound Sensor detected sound then
    repeat until Button 1 ▾ pressed and Button 2 ▾ pressed
        Buzzer tone 200
```

7. Add blocks Neopixel LED followed by a block delay and block Neopixel LED off.



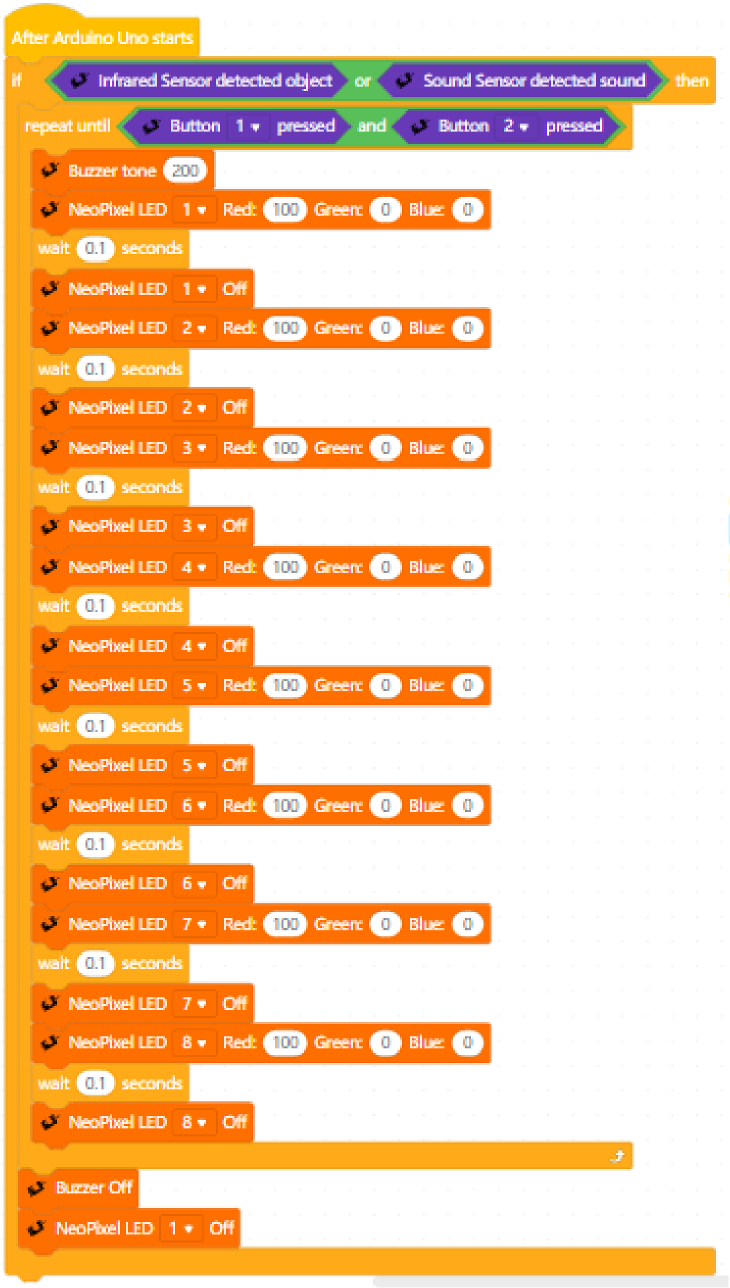
```
After Arduino Uno starts
if Infrared Sensor detected object or Sound Sensor detected sound then
    repeat until Button 1 ▾ pressed and Button 2 ▾ pressed
        Buzzer tone 200
        NeoPixel LED 1 ▾ Red: 100 Green: 0 Blue: 0
        wait 0.1 seconds
        NeoPixel LED 1 ▾ Off
```

8. Repeat Step 7 for each of the 8 Neopixel LEDs



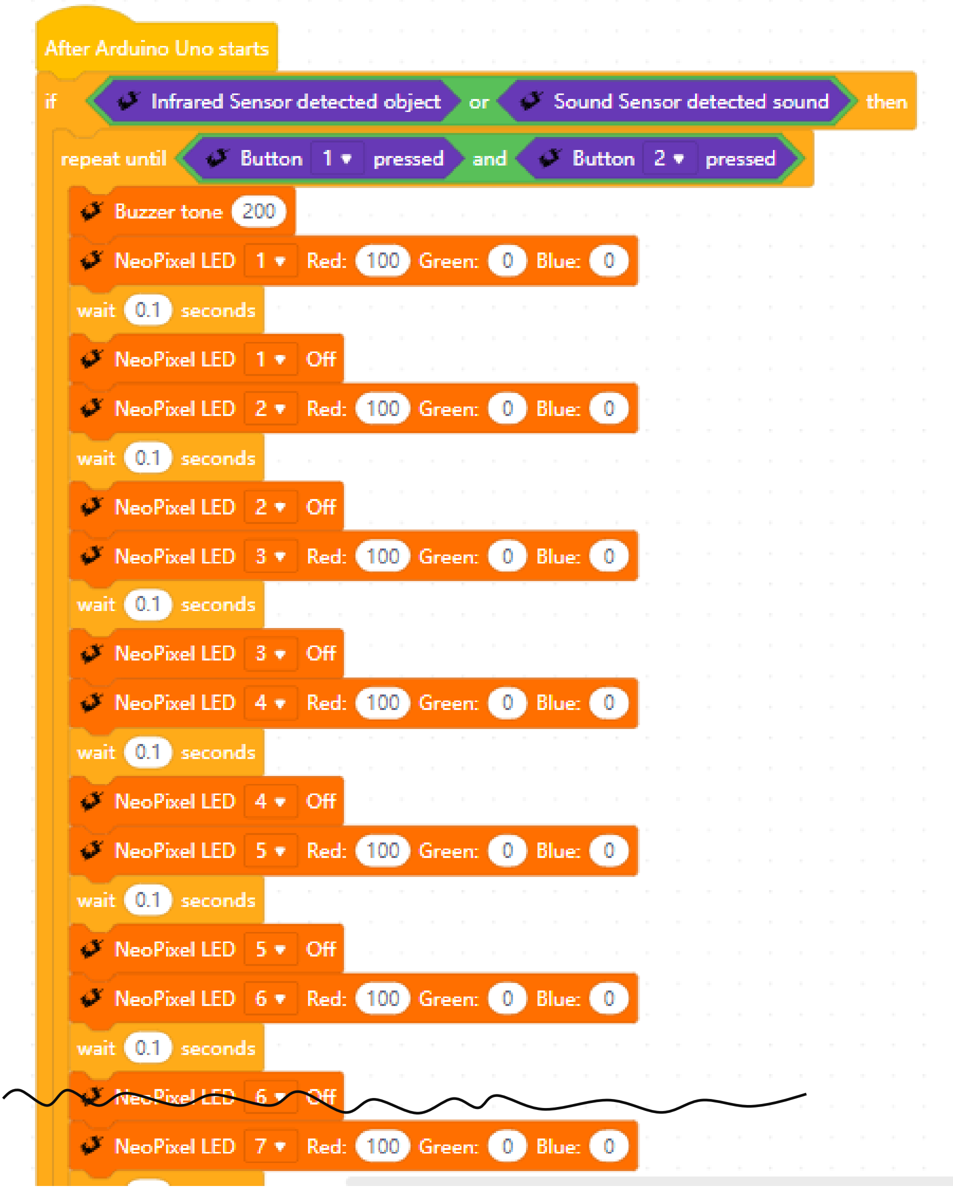
```
After Arduino Uno starts
if  Infrared Sensor detected object  or  Sound Sensor detected sound  then
  repeat until  Button 1 pressed  and  Button 2 pressed
    Buzzer tone 200
    NeoPixel LED 1 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 1 Off
    NeoPixel LED 2 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 2 Off
    NeoPixel LED 3 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 3 Off
    NeoPixel LED 4 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 4 Off
    NeoPixel LED 5 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 5 Off
    NeoPixel LED 6 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 6 Off
    NeoPixel LED 7 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 7 Off
    NeoPixel LED 8 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 8 Off
```

9. Then, insert the block buzzer off and block Neopixel LED off in the block if, then.

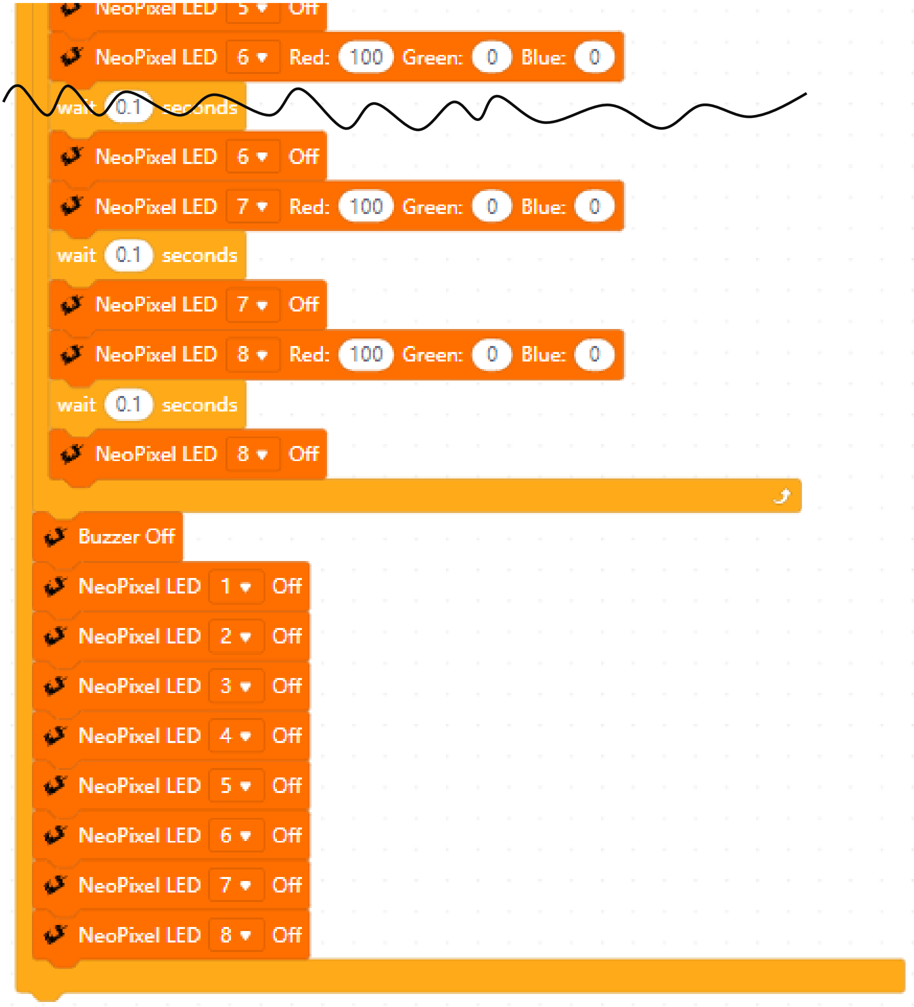


```
After Arduino Uno starts
if  Infrared Sensor detected object  or  Sound Sensor detected sound  then
  repeat until  Button 1 ▾ pressed  and  Button 2 ▾ pressed
    Buzzer tone 200
    NeoPixel LED 1 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 1 ▾ Off
    NeoPixel LED 2 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 2 ▾ Off
    NeoPixel LED 3 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 4 ▾ Off
    NeoPixel LED 5 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 8 ▾ Off
  Buzzer Off
  NeoPixel LED 1 ▾ Off
```

10. Add blocks Neopixel LED off for every 8 neopixel LEDs.



```
After Arduino Uno starts
if  Infrared Sensor detected object  or  Sound Sensor detected sound  then
  repeat until  Button 1 pressed  and  Button 2 pressed
    Buzzer tone 200
    NeoPixel LED 1 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 1 Off
    NeoPixel LED 2 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 2 Off
    NeoPixel LED 3 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 3 Off
    NeoPixel LED 4 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 4 Off
    NeoPixel LED 5 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 5 Off
    NeoPixel LED 6 Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 6 Off
    NeoPixel LED 7 Red: 100 Green: 0 Blue: 0
```

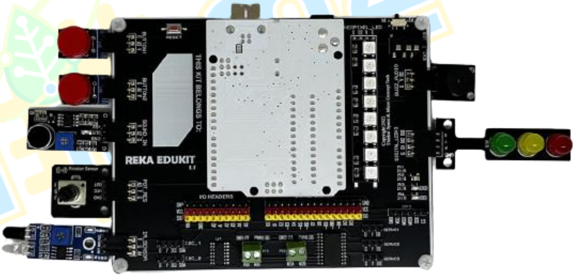


```
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Red: 100 Green: 0 Blue: 0
    wait 0.1 seconds
    NeoPixel LED 8 ▾ Off
  Buzzer Off
  NeoPixel LED 1 ▾ Off
  NeoPixel LED 2 ▾ Off
  NeoPixel LED 3 ▾ Off
  NeoPixel LED 4 ▾ Off
  NeoPixel LED 5 ▾ Off
  NeoPixel LED 6 ▾ Off
  NeoPixel LED 7 ▾ Off
  NeoPixel LED 8 ▾ Off
```

10: CLASSROOM SYSTEM

Let's build a classroom system in the classroom! The class system begins when the students enter the class. The infrared sensor will turn on a single color LED (GYR LED) in green. If the student is absent from school, we need to press button 2 to turn on the yellow light on the GYR LED. When the sound sensor detects the noise of a noisy student in the class, a buzzer will sound indicating that the student needs to be quiet. To stop the buzzer, we need to press the push button 1. Next, we can turn on the lights in the classroom according to the brightness we want using multi-colored LEDs (Neopixel LEDs) and a potentiometer as an adjustable switch.

DEVICE USED :

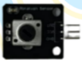


1. RekaEdukit Circuit Board

2. Push Button

3. Sound Sensor

4. Infrared Sensor

5. Potentiometer 

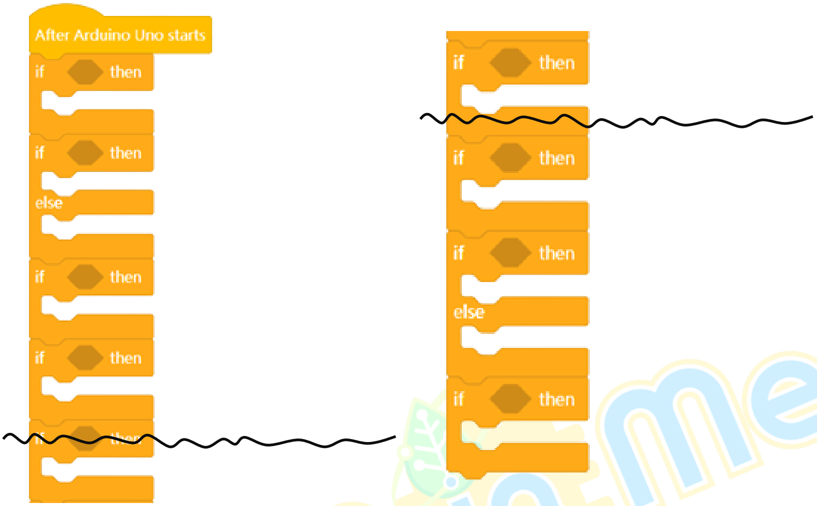
6. Single color LED (GYR LED)

7. Multi-coloured LED (Neopixel LED)

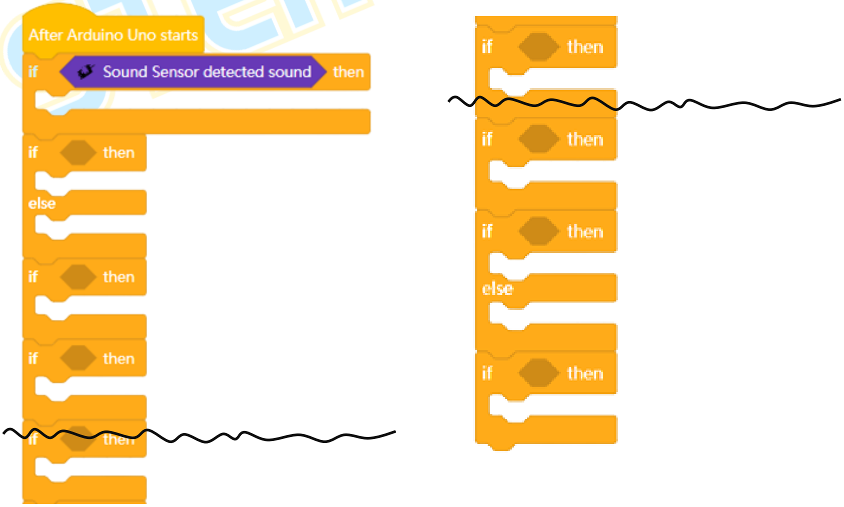
8. Buzzer

TUTORIAL

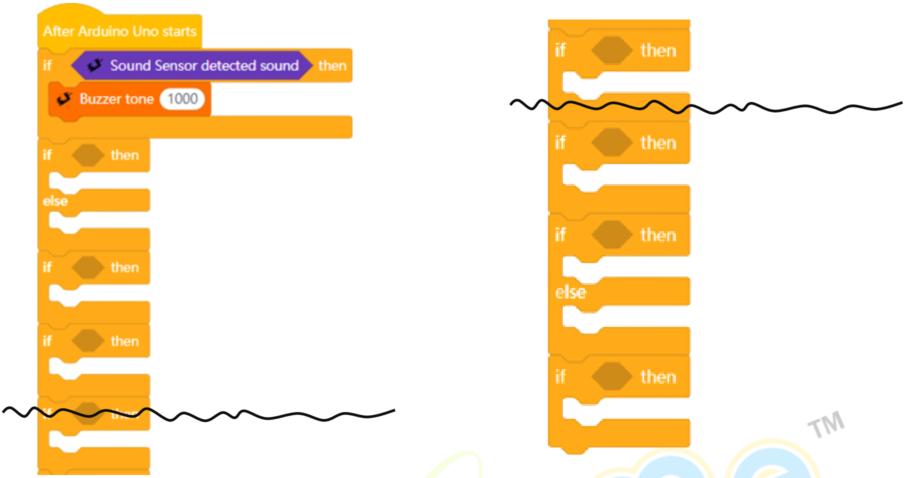
1. Drag After Arduino Uno starts along with block if, then and block if, then, else.



2. Insert block Sound Sensor detected sound into the hexagonal space inside the block if, then.



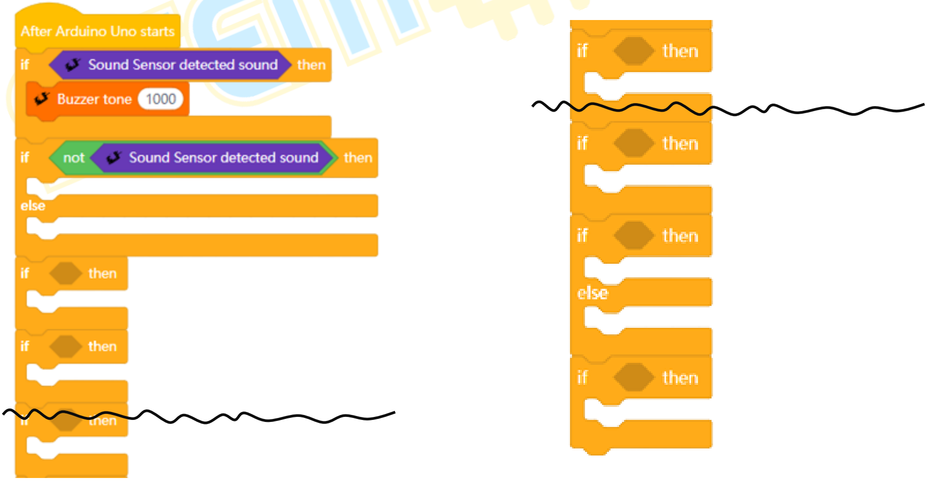
3. Insert block Buzzer tone into the block if, then.



```
After Arduino Uno starts
if [Sound Sensor detected sound] then
    Buzzer tone (1000)
if < > then
else
if < > then
if < > then
```

```
if < > then
if < > then
if < > then
else
if < > then
```

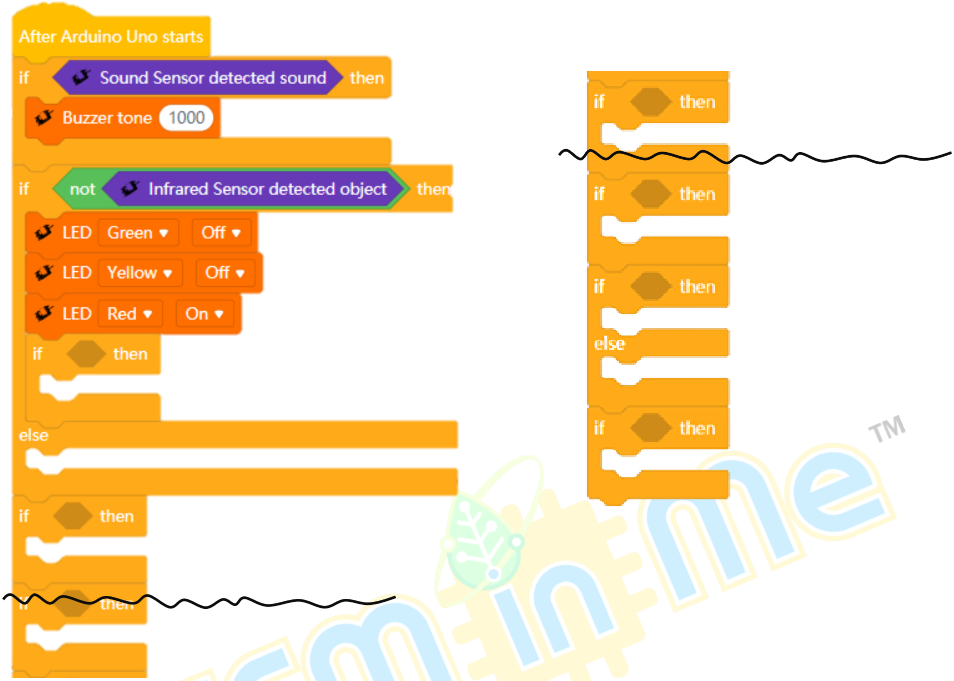
4. Then, insert the block not into the hexagonal space inside the block if, then and insert the block Infrared Sensor detected object into the block not.



```
After Arduino Uno starts
if [Sound Sensor detected sound] then
    Buzzer tone (1000)
if not [Sound Sensor detected sound] then
else
if < > then
if < > then
```

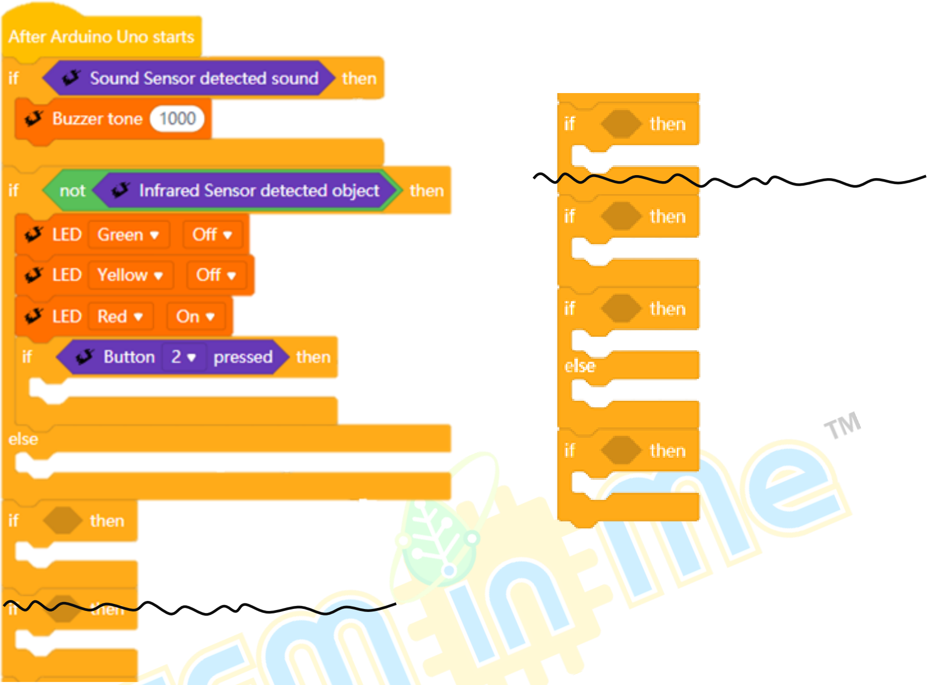
```
if < > then
if < > then
if < > then
else
if < > then
```

5. Insert the block LED for each Green, Yellow and Red color and block if, then into the block if, then.

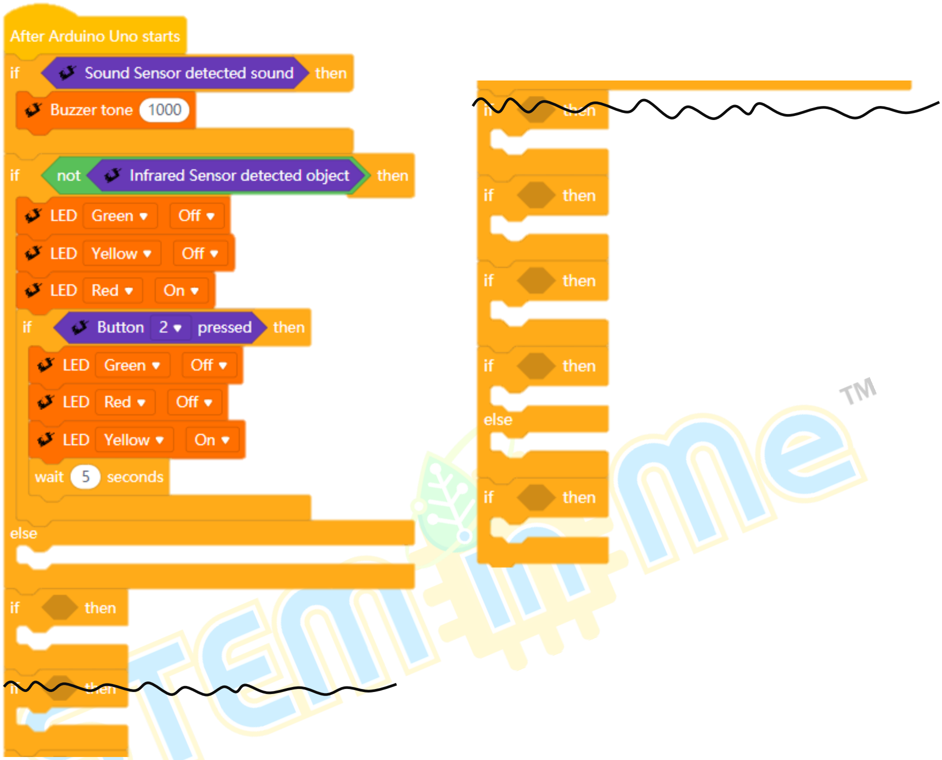

```
After Arduino Uno starts
if  Sound Sensor detected sound  then
    Buzzer tone 1000
if  not  Infrared Sensor detected object  then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
    if  then
else
if  then
if  then
```

```
if  then
if  then
if  then
else
if  then
```

6. Insert the block Button pressed into the hexagonal space inside the block if, then.



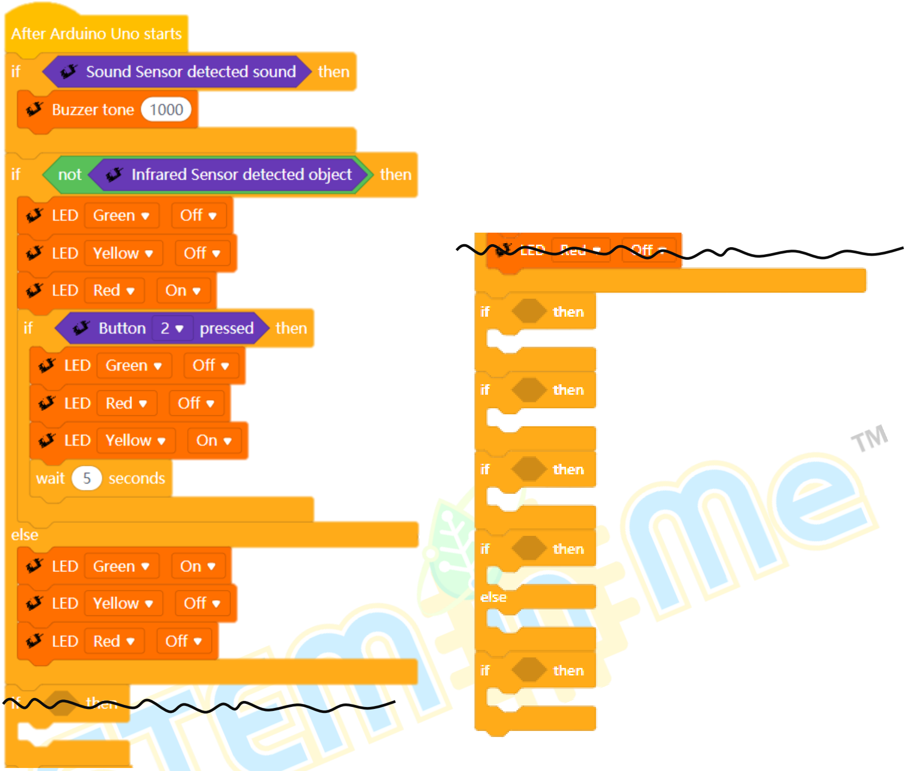
7. Add blocks LED for each color Green, Yellow and Red as well as blocks delay into the block if, then.



```
After Arduino Uno starts
if  Sound Sensor detected sound  then
    Buzzer tone 1000
if  not  Infrared Sensor detected object  then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
    if  Button 2 ▾ pressed  then
        LED Green ▾ Off ▾
        LED Red ▾ Off ▾
        LED Yellow ▾ On ▾
        wait 5 seconds
else
if  then
if  then
```

```
if  then
if  then
if  then
if  then
else
if  then
```

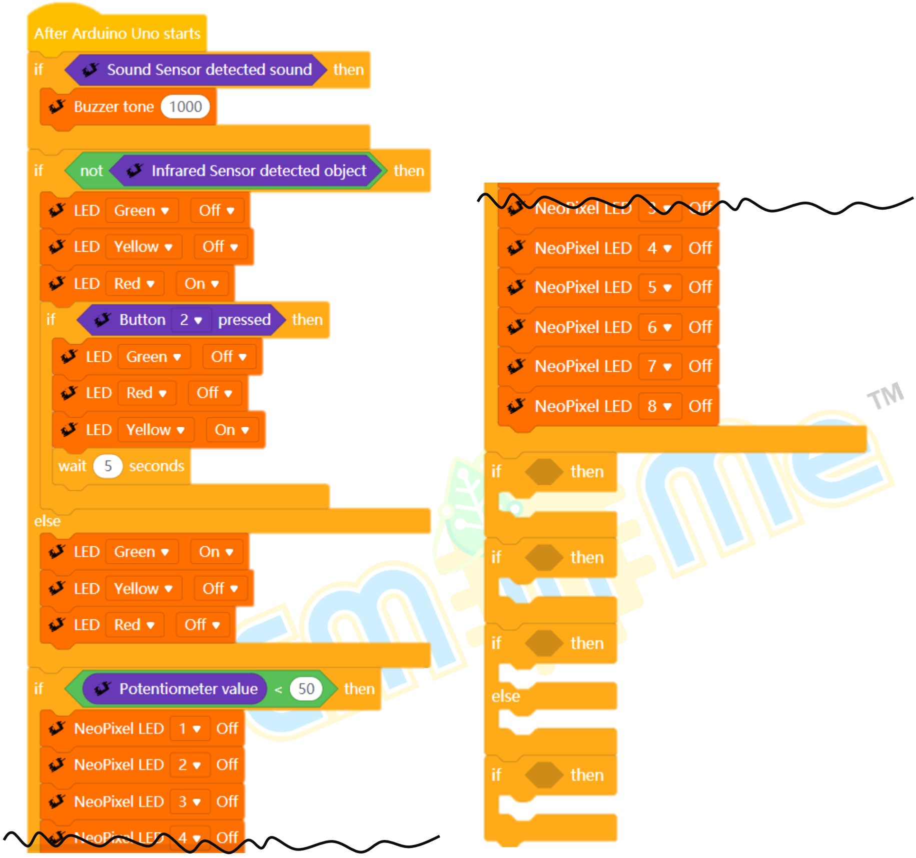
8. Insert the block LED for each colour Green, Yellow and Red into the block else.



```
After Arduino Uno starts
if  Sound Sensor detected sound  then
    Buzzer tone 1000
if  not  Infrared Sensor detected object  then
    LED Green Off
    LED Yellow Off
    LED Red On
    if  Button 2 pressed  then
        LED Green Off
        LED Red Off
        LED Yellow On
        wait 5 seconds
else
    LED Green On
    LED Yellow Off
    LED Red Off
if  then
```

```
LED Red Off
if  then
if  then
if  then
if  then
else
if  then
```

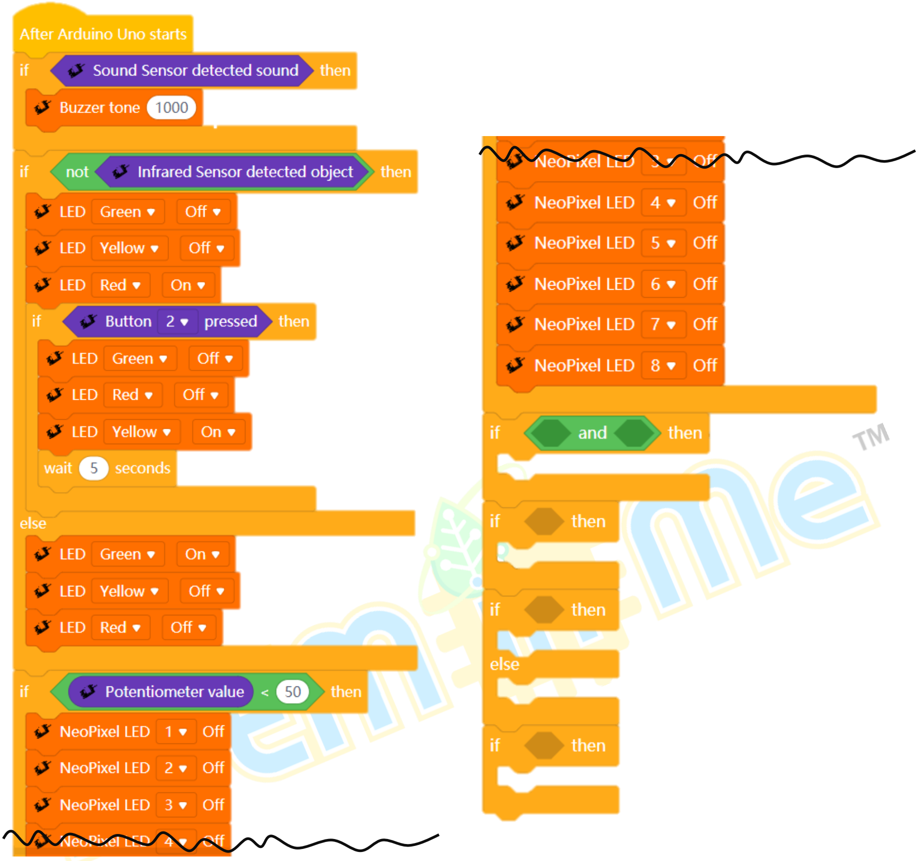
9. Insert the block less than into the hexagonal space inside the block if, then and insert the block potentiometer value into the block less than.



```
After Arduino Uno starts
if  Sound Sensor detected sound  then
    Buzzer tone 1000
if  not  Infrared Sensor detected object  then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
    if  Button 2 ▾ pressed  then
        LED Green ▾ Off ▾
        LED Red ▾ Off ▾
        LED Yellow ▾ On ▾
        wait 5 seconds
else
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾
if  Potentiometer value < 50  then
    NeoPixel LED 1 ▾ Off
    NeoPixel LED 2 ▾ Off
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
```

```
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Off
if  then
if  then
if  then
else
if  then
```

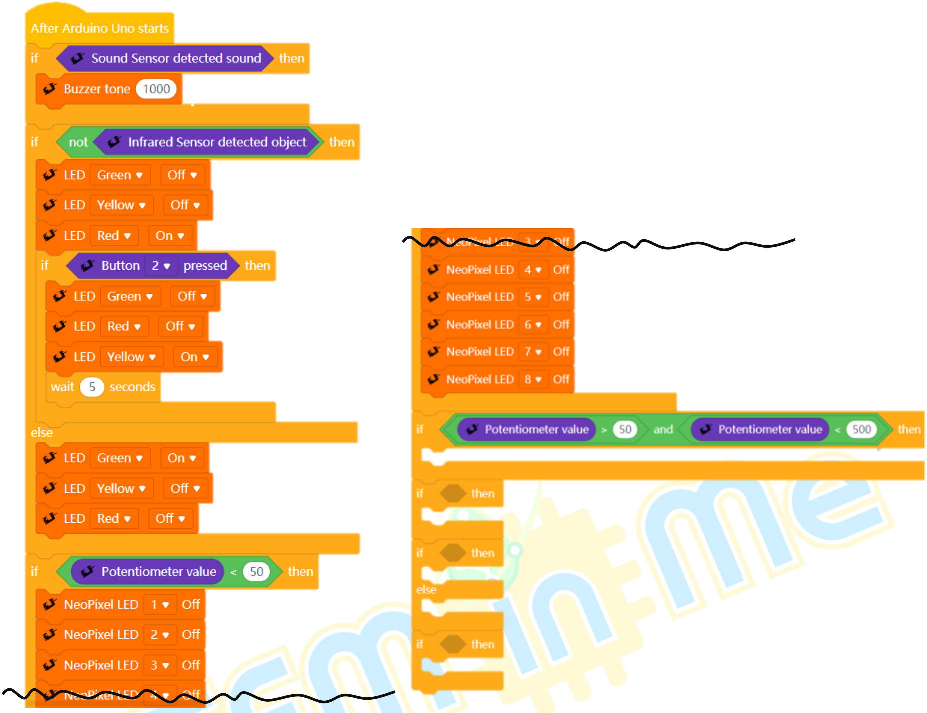
10. Insert 8 blocks Neopixel LED off into the block if. then.



```
After Arduino Uno starts
if   Sound Sensor detected sound   then
    Buzzer tone 1000
if   not   Infrared Sensor detected object   then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
    if   Button 2 ▾ pressed   then
        LED Green ▾ Off ▾
        LED Red ▾ Off ▾
        LED Yellow ▾ On ▾
        wait 5 seconds
else
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾
if   Potentiometer value < 50   then
    NeoPixel LED 1 ▾ Off
    NeoPixel LED 2 ▾ Off
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
```

```
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Off
if   and   then
if   then
if   then
else
if   then
```

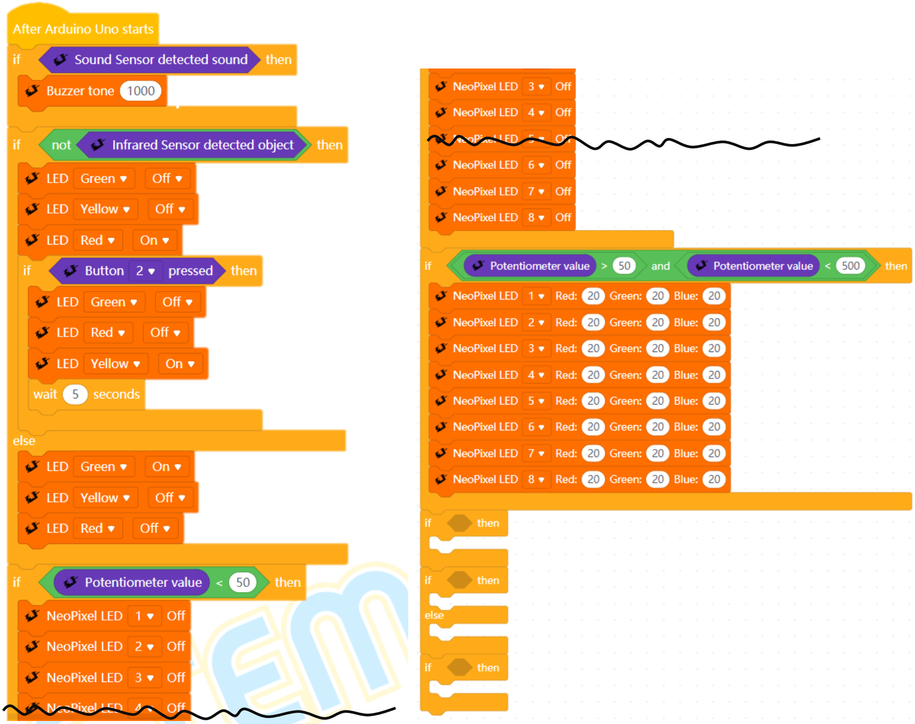
11. Insert the block and into the hexagonal space inside the block if, then.



```
After Arduino Uno starts
  if  Sound Sensor detected sound  then
    Buzzer tone 1000
  if  not  Infrared Sensor detected object  then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
    if  Button 2 ▾ pressed  then
      LED Green ▾ Off ▾
      LED Red ▾ Off ▾
      LED Yellow ▾ On ▾
      wait 5 seconds
  else
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾
  if  Potentiometer value < 50  then
    NeoPixel LED 1 ▾ Off
    NeoPixel LED 2 ▾ Off
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
```

```
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Off
  if  Potentiometer value > 50  and  Potentiometer value < 500  then
  if  then
  if  then
  else
    if  then
```

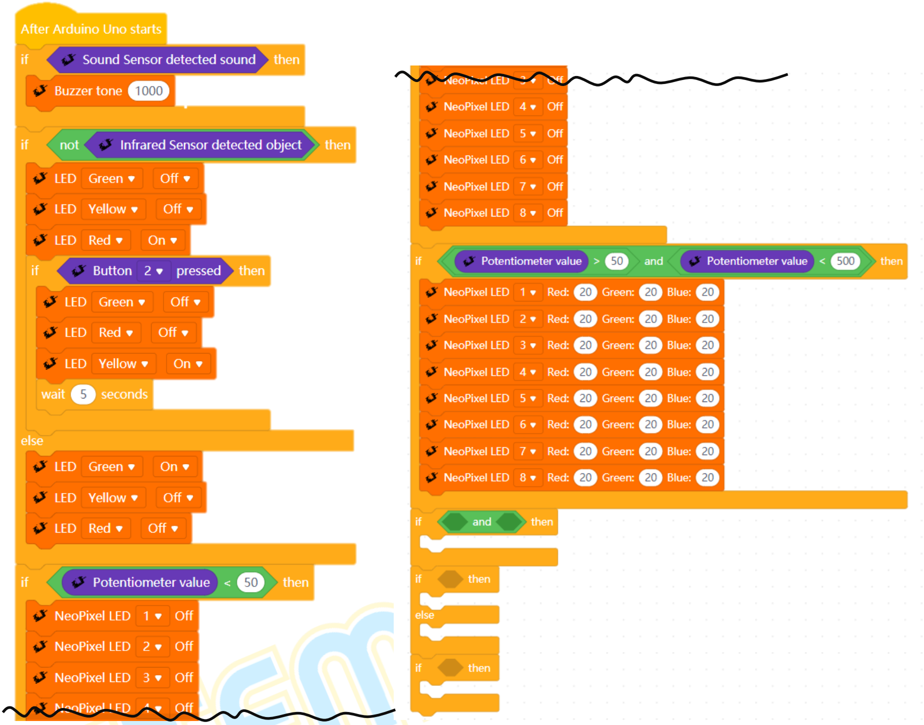
12. Insert the block greater than and less than into the block and and insert the block potentiometer value into the block greater than and less than.



```
After Arduino Uno starts
if Sound Sensor detected sound then
    Buzzer tone 1000

if not Infrared Sensor detected object then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
    if Button 2 ▾ pressed then
        LED Green ▾ Off ▾
        LED Red ▾ Off ▾
        LED Yellow ▾ On ▾
        wait 5 seconds

else
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾

if Potentiometer value < 50 then
    NeoPixel LED 1 ▾ Off
    NeoPixel LED 2 ▾ Off
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
```

```
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Off

if Potentiometer value > 50 and Potentiometer value < 500 then
    NeoPixel LED 1 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 2 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 3 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 4 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 5 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 6 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 7 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 8 ▾ Red: 20 Green: 20 Blue: 20

if then

if then
else

if then
```

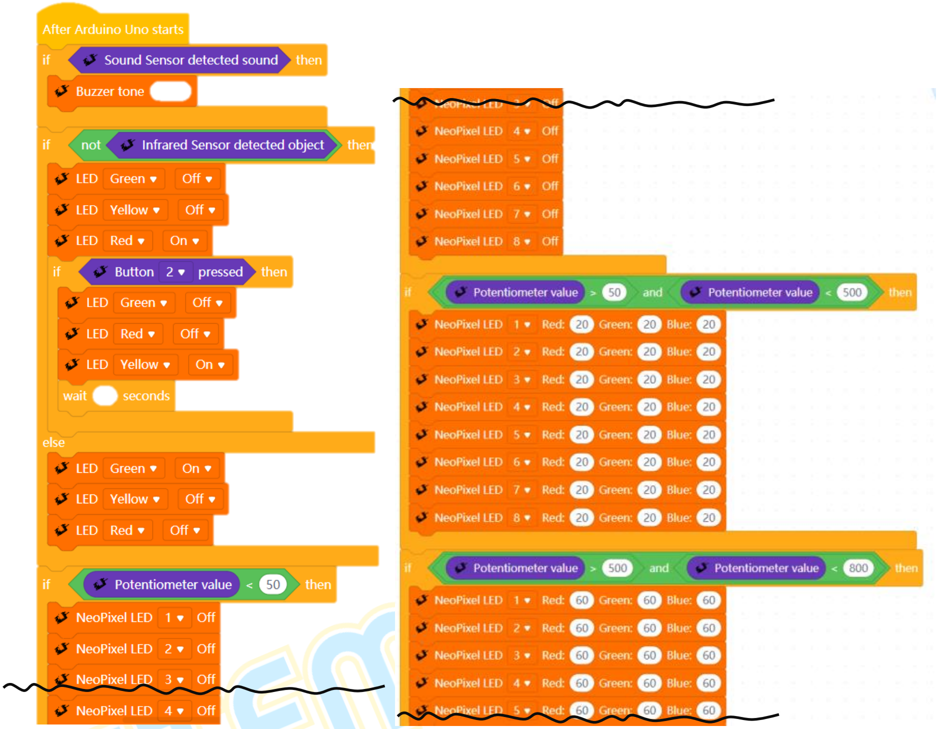
13. Then, insert 8 blocks Neopixel LED into the block if, then.



```
After Arduino Uno starts
if  Sound Sensor detected sound  then
    Buzzer tone 1000
if  not  Infrared Sensor detected object  then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
    if  Button 2 ▾ pressed  then
        LED Green ▾ Off ▾
        LED Red ▾ Off ▾
        LED Yellow ▾ On ▾
        wait 5 seconds
else
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾
if  Potentiometer value < 50  then
    NeoPixel LED 1 ▾ Off
    NeoPixel LED 2 ▾ Off
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
```

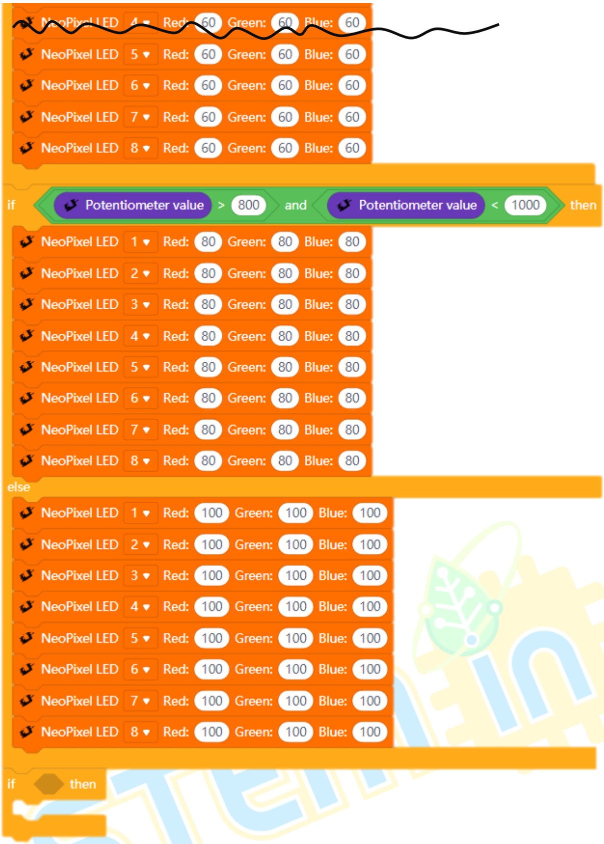
```
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Off
if  Potentiometer value > 50  and  Potentiometer value < 500  then
    NeoPixel LED 1 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 2 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 3 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 4 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 5 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 6 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 7 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 8 ▾ Red: 20 Green: 20 Blue: 20
if  and  then
if  then
else
if  then
```

14. Repeat this step until the block else.



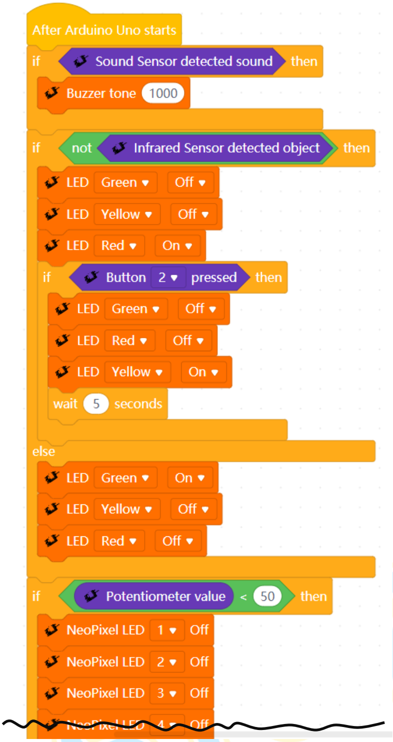
```
After Arduino Uno starts
if Sound Sensor detected sound then
    Buzzer tone
if not Infrared Sensor detected object then
    LED Green ▾ Off ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ On ▾
    if Button 2 ▾ pressed then
        LED Green ▾ Off ▾
        LED Red ▾ Off ▾
        LED Yellow ▾ On ▾
        wait seconds
else
    LED Green ▾ On ▾
    LED Yellow ▾ Off ▾
    LED Red ▾ Off ▾
if Potentiometer value < 50 then
    NeoPixel LED 1 ▾ Off
    NeoPixel LED 2 ▾ Off
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
```

```
    NeoPixel LED 3 ▾ Off
    NeoPixel LED 4 ▾ Off
    NeoPixel LED 5 ▾ Off
    NeoPixel LED 6 ▾ Off
    NeoPixel LED 7 ▾ Off
    NeoPixel LED 8 ▾ Off
if Potentiometer value > 50 and Potentiometer value < 500 then
    NeoPixel LED 1 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 2 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 3 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 4 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 5 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 6 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 7 ▾ Red: 20 Green: 20 Blue: 20
    NeoPixel LED 8 ▾ Red: 20 Green: 20 Blue: 20
if Potentiometer value > 500 and Potentiometer value < 800 then
    NeoPixel LED 1 ▾ Red: 60 Green: 60 Blue: 60
    NeoPixel LED 2 ▾ Red: 60 Green: 60 Blue: 60
    NeoPixel LED 3 ▾ Red: 60 Green: 60 Blue: 60
    NeoPixel LED 4 ▾ Red: 60 Green: 60 Blue: 60
    NeoPixel LED 5 ▾ Red: 60 Green: 60 Blue: 60
```

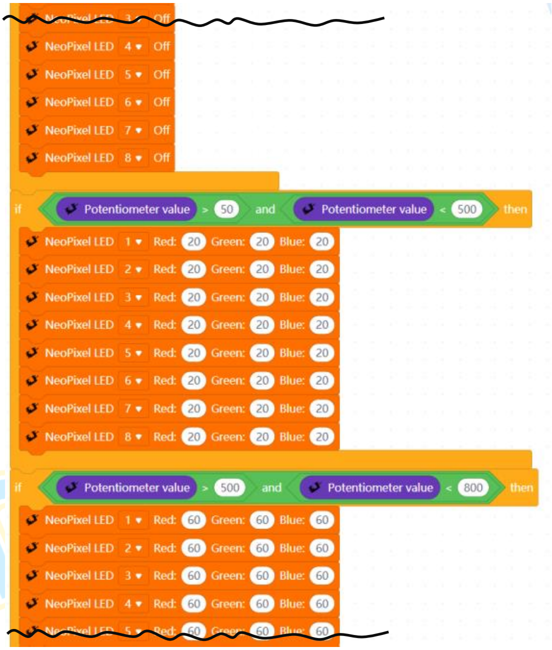


```
  NeoPixel LED 4 ▾ Red: 60 Green: 60 Blue: 60
  NeoPixel LED 5 ▾ Red: 60 Green: 60 Blue: 60
  NeoPixel LED 6 ▾ Red: 60 Green: 60 Blue: 60
  NeoPixel LED 7 ▾ Red: 60 Green: 60 Blue: 60
  NeoPixel LED 8 ▾ Red: 60 Green: 60 Blue: 60
if  Potentiometer value > 800 and Potentiometer value < 1000 then
  NeoPixel LED 1 ▾ Red: 80 Green: 80 Blue: 80
  NeoPixel LED 2 ▾ Red: 80 Green: 80 Blue: 80
  NeoPixel LED 3 ▾ Red: 80 Green: 80 Blue: 80
  NeoPixel LED 4 ▾ Red: 80 Green: 80 Blue: 80
  NeoPixel LED 5 ▾ Red: 80 Green: 80 Blue: 80
  NeoPixel LED 6 ▾ Red: 80 Green: 80 Blue: 80
  NeoPixel LED 7 ▾ Red: 80 Green: 80 Blue: 80
  NeoPixel LED 8 ▾ Red: 80 Green: 80 Blue: 80
else
  NeoPixel LED 1 ▾ Red: 100 Green: 100 Blue: 100
  NeoPixel LED 2 ▾ Red: 100 Green: 100 Blue: 100
  NeoPixel LED 3 ▾ Red: 100 Green: 100 Blue: 100
  NeoPixel LED 4 ▾ Red: 100 Green: 100 Blue: 100
  NeoPixel LED 5 ▾ Red: 100 Green: 100 Blue: 100
  NeoPixel LED 6 ▾ Red: 100 Green: 100 Blue: 100
  NeoPixel LED 7 ▾ Red: 100 Green: 100 Blue: 100
  NeoPixel LED 8 ▾ Red: 100 Green: 100 Blue: 100
if  then
```

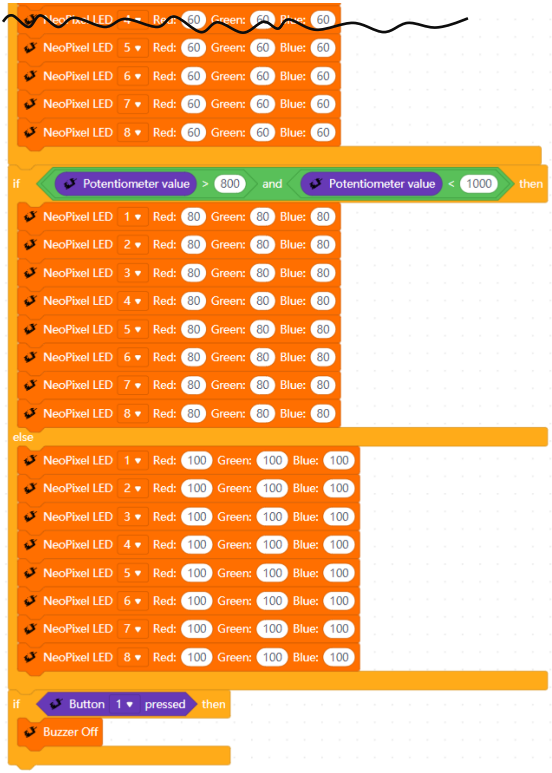
15. Insert the block button 1 pressed into the hexagonal space inside the block if, then. Then, insert the block Buzzer Off into the block if, then.



```
After Arduino Uno starts
if Sound Sensor detected sound then
  Buzzer tone 1000
if not Infrared Sensor detected object then
  LED Green ▾ Off ▾
  LED Yellow ▾ Off ▾
  LED Red ▾ On ▾
  if Button 2 ▾ pressed then
    LED Green ▾ Off ▾
    LED Red ▾ Off ▾
    LED Yellow ▾ On ▾
    wait 5 seconds
else
  LED Green ▾ On ▾
  LED Yellow ▾ Off ▾
  LED Red ▾ Off ▾
if Potentiometer value < 50 then
  NeoPixel LED 1 ▾ Off
  NeoPixel LED 2 ▾ Off
  NeoPixel LED 3 ▾ Off
  NeoPixel LED 4 ▾ Off
```



```
  NeoPixel LED 3 ▾ Off
  NeoPixel LED 4 ▾ Off
  NeoPixel LED 5 ▾ Off
  NeoPixel LED 6 ▾ Off
  NeoPixel LED 7 ▾ Off
  NeoPixel LED 8 ▾ Off
if Potentiometer value > 50 and Potentiometer value < 500 then
  NeoPixel LED 1 ▾ Red: 20 Green: 20 Blue: 20
  NeoPixel LED 2 ▾ Red: 20 Green: 20 Blue: 20
  NeoPixel LED 3 ▾ Red: 20 Green: 20 Blue: 20
  NeoPixel LED 4 ▾ Red: 20 Green: 20 Blue: 20
  NeoPixel LED 5 ▾ Red: 20 Green: 20 Blue: 20
  NeoPixel LED 6 ▾ Red: 20 Green: 20 Blue: 20
  NeoPixel LED 7 ▾ Red: 20 Green: 20 Blue: 20
  NeoPixel LED 8 ▾ Red: 20 Green: 20 Blue: 20
if Potentiometer value > 500 and Potentiometer value < 800 then
  NeoPixel LED 1 ▾ Red: 60 Green: 60 Blue: 60
  NeoPixel LED 2 ▾ Red: 60 Green: 60 Blue: 60
  NeoPixel LED 3 ▾ Red: 60 Green: 60 Blue: 60
  NeoPixel LED 4 ▾ Red: 60 Green: 60 Blue: 60
  NeoPixel LED 5 ▾ Red: 60 Green: 60 Blue: 60
```



```
    NeoPixel LED 4 ▾ Red: 60 Green: 60 Blue: 60
    NeoPixel LED 5 ▾ Red: 60 Green: 60 Blue: 60
    NeoPixel LED 6 ▾ Red: 60 Green: 60 Blue: 60
    NeoPixel LED 7 ▾ Red: 60 Green: 60 Blue: 60
    NeoPixel LED 8 ▾ Red: 60 Green: 60 Blue: 60
if Potentiometer value > 800 and Potentiometer value < 1000 then
    NeoPixel LED 1 ▾ Red: 80 Green: 80 Blue: 80
    NeoPixel LED 2 ▾ Red: 80 Green: 80 Blue: 80
    NeoPixel LED 3 ▾ Red: 80 Green: 80 Blue: 80
    NeoPixel LED 4 ▾ Red: 80 Green: 80 Blue: 80
    NeoPixel LED 5 ▾ Red: 80 Green: 80 Blue: 80
    NeoPixel LED 6 ▾ Red: 80 Green: 80 Blue: 80
    NeoPixel LED 7 ▾ Red: 80 Green: 80 Blue: 80
    NeoPixel LED 8 ▾ Red: 80 Green: 80 Blue: 80
else
    NeoPixel LED 1 ▾ Red: 100 Green: 100 Blue: 100
    NeoPixel LED 2 ▾ Red: 100 Green: 100 Blue: 100
    NeoPixel LED 3 ▾ Red: 100 Green: 100 Blue: 100
    NeoPixel LED 4 ▾ Red: 100 Green: 100 Blue: 100
    NeoPixel LED 5 ▾ Red: 100 Green: 100 Blue: 100
    NeoPixel LED 6 ▾ Red: 100 Green: 100 Blue: 100
    NeoPixel LED 7 ▾ Red: 100 Green: 100 Blue: 100
    NeoPixel LED 8 ▾ Red: 100 Green: 100 Blue: 100
if Button 1 ▾ pressed then
    Buzzer Off
```

MERAKYATKAN TEKNOLOGI

- Industry 4WRD
- Pemikiran Kreatif
- Pembudayaan Inovasi
- Kesejahteraan Hidup
- Kelestarian Alam
- Pembelajaran Menyeronokkan

PENGELUAR:

MICRO CONCEPT TECH SDN BHD
1230153-W

No. 5-5, Pusat Dagangan Shah Alam,
Persiaran Damai, Seksyen 11,
40100 Shah Alam, Selangor, Malaysia

@steminme

http://www.microconcept.com.my
steminme@microconcept.com.my