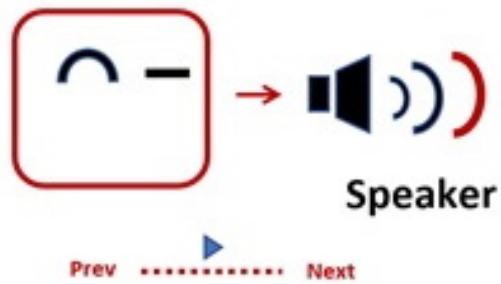
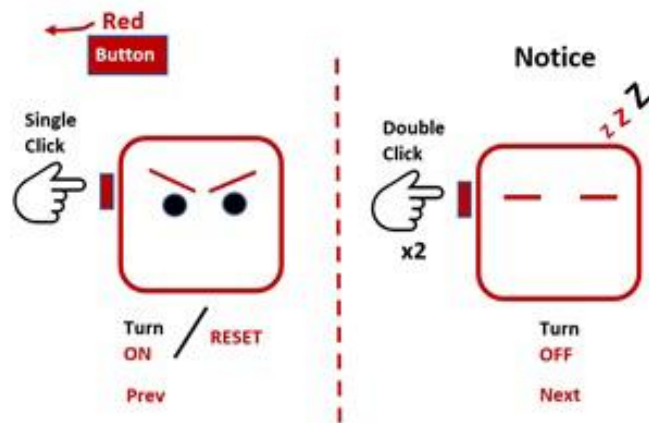
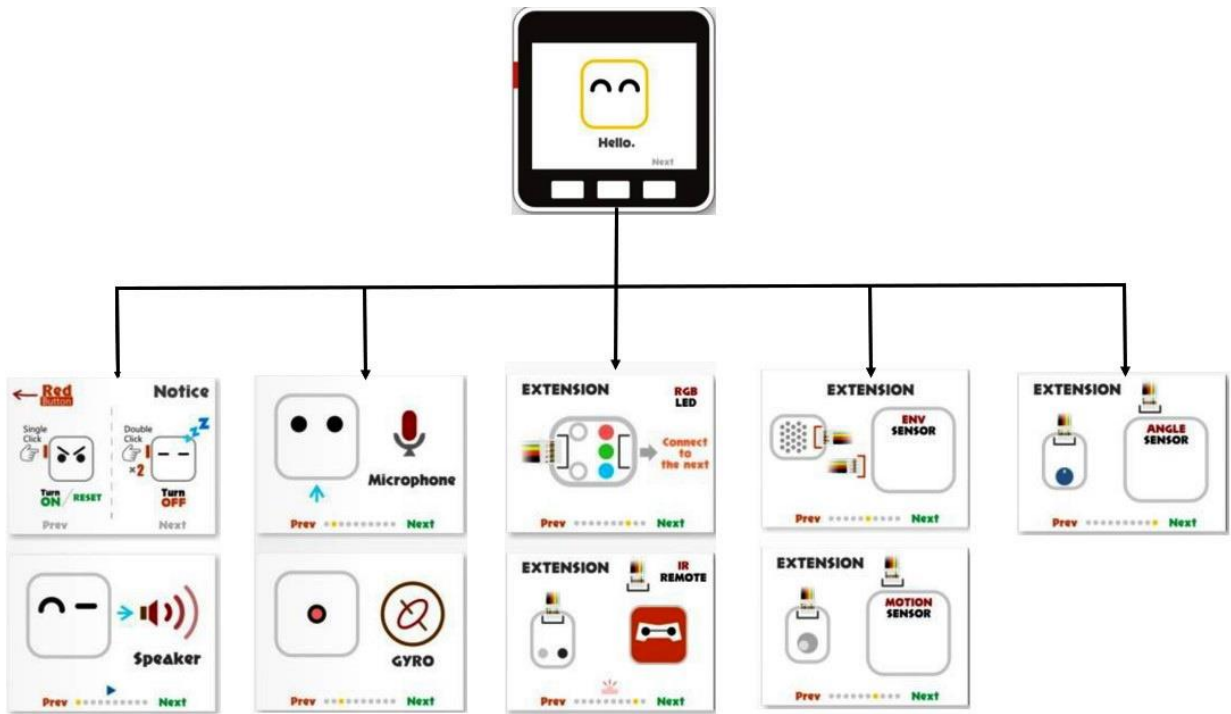
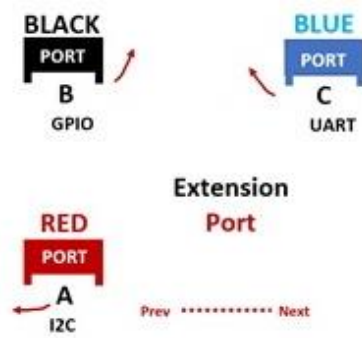
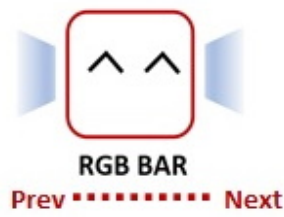
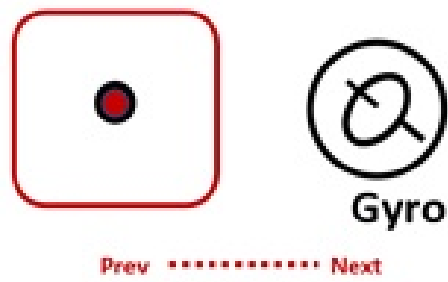
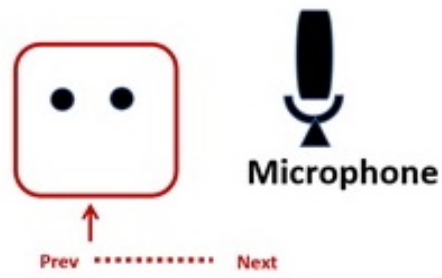


Chapter 1: Exploring the M5Stack Core





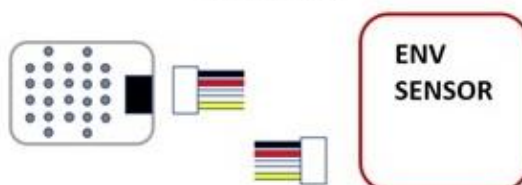




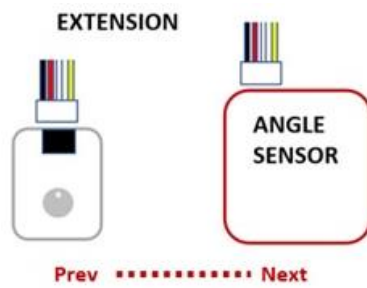
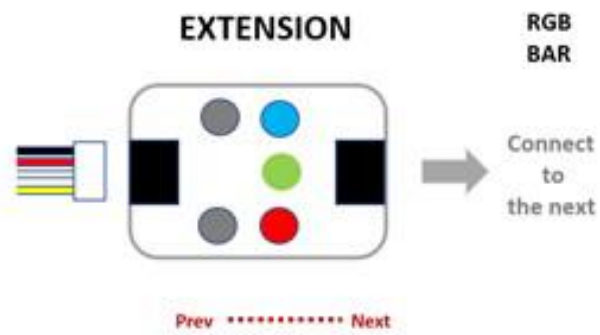
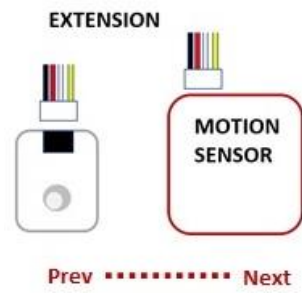
M5STACK

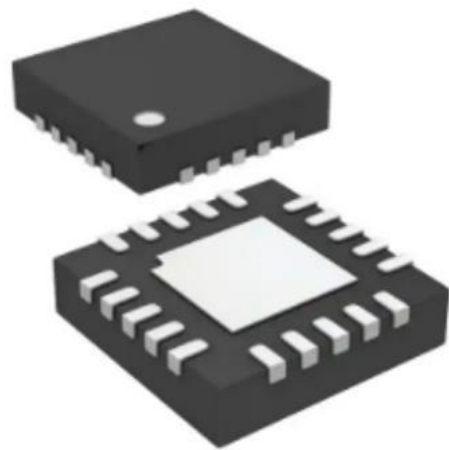
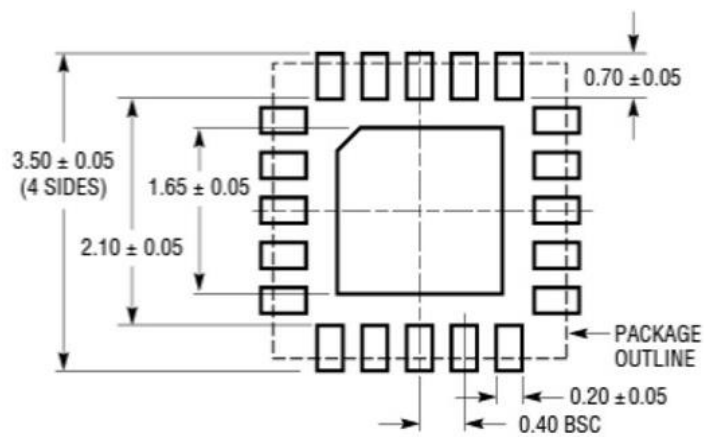
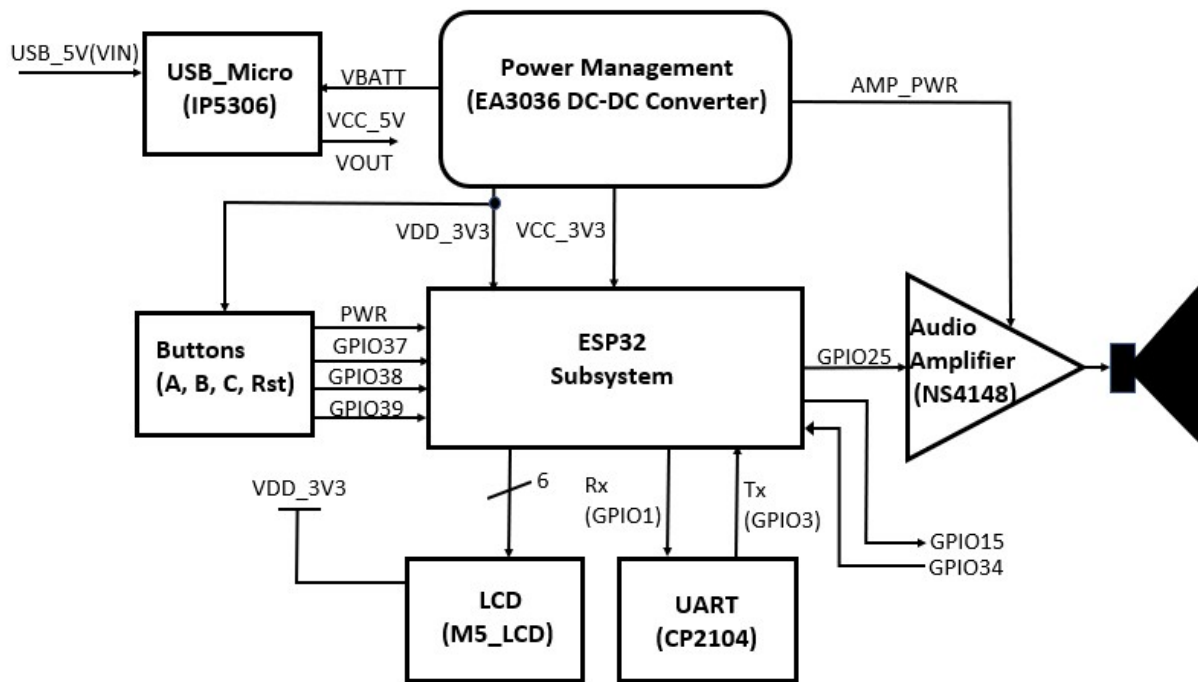


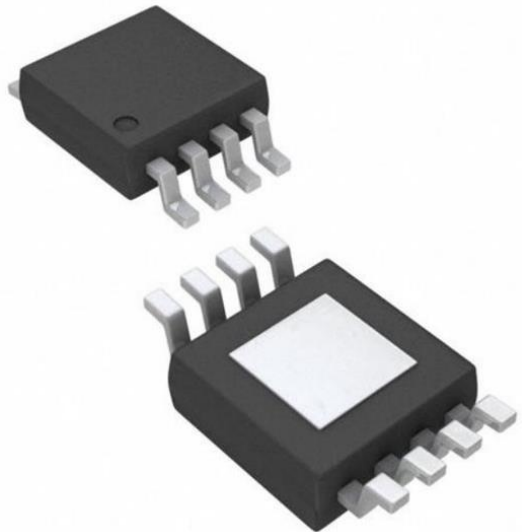
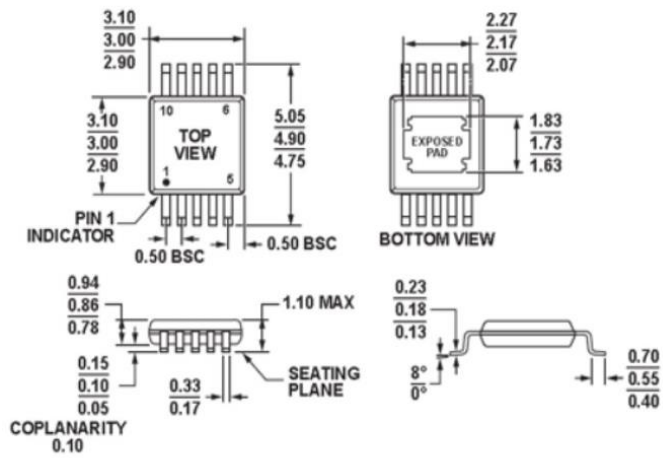
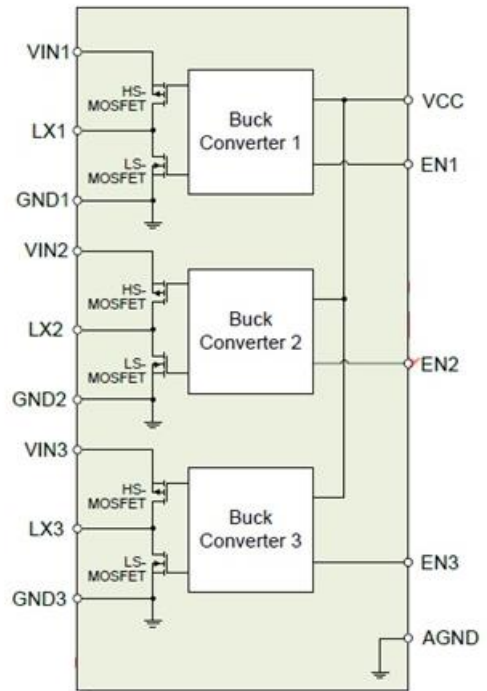
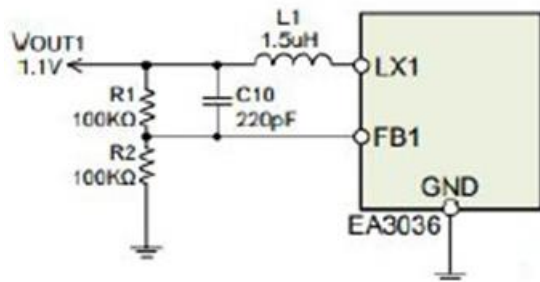
EXTENSION

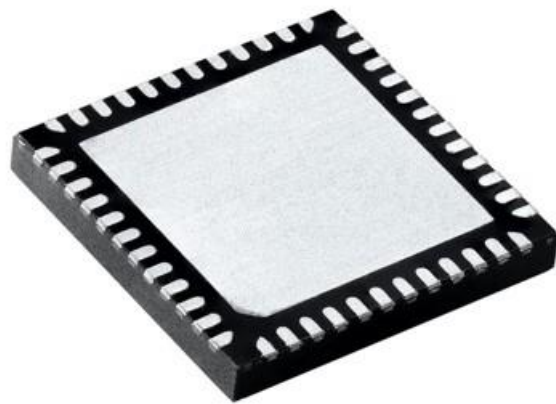
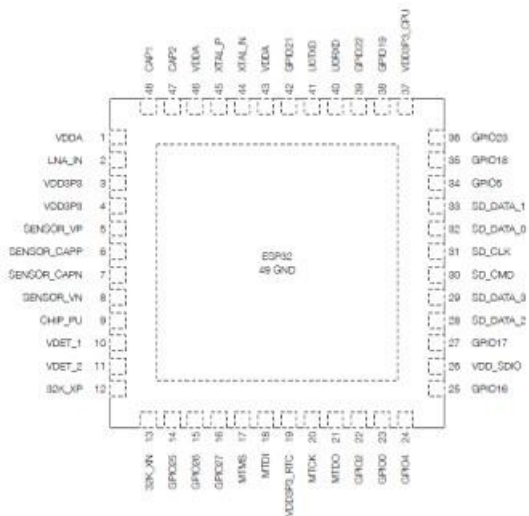


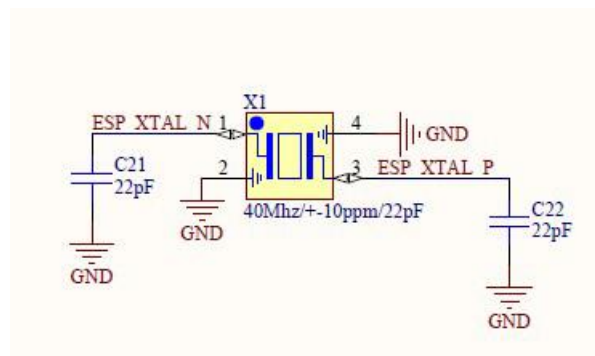
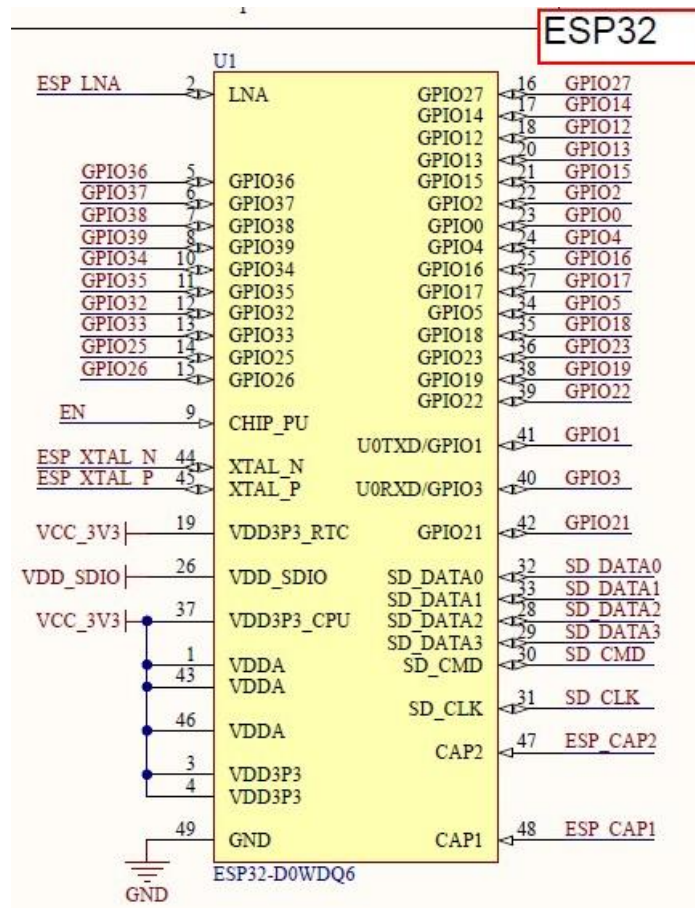
Prev Next

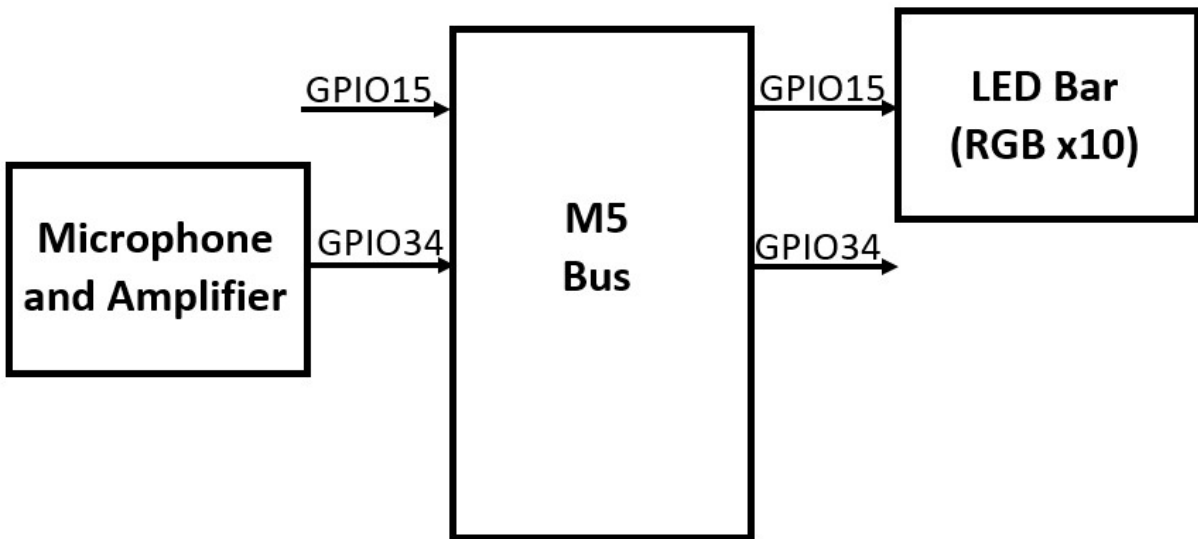
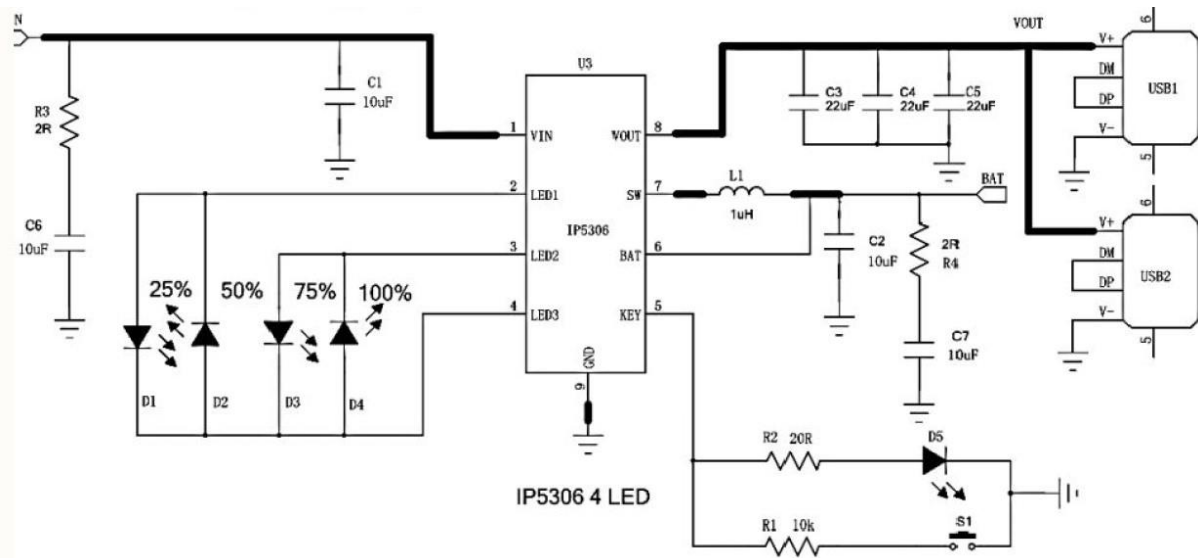


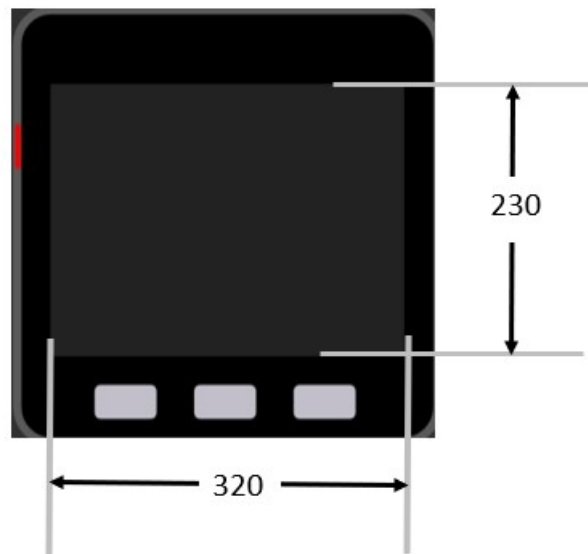
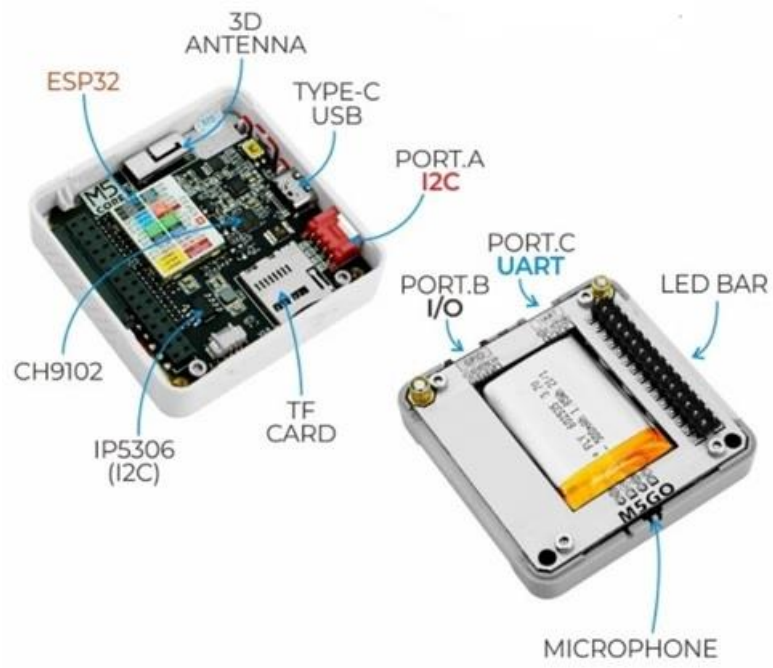


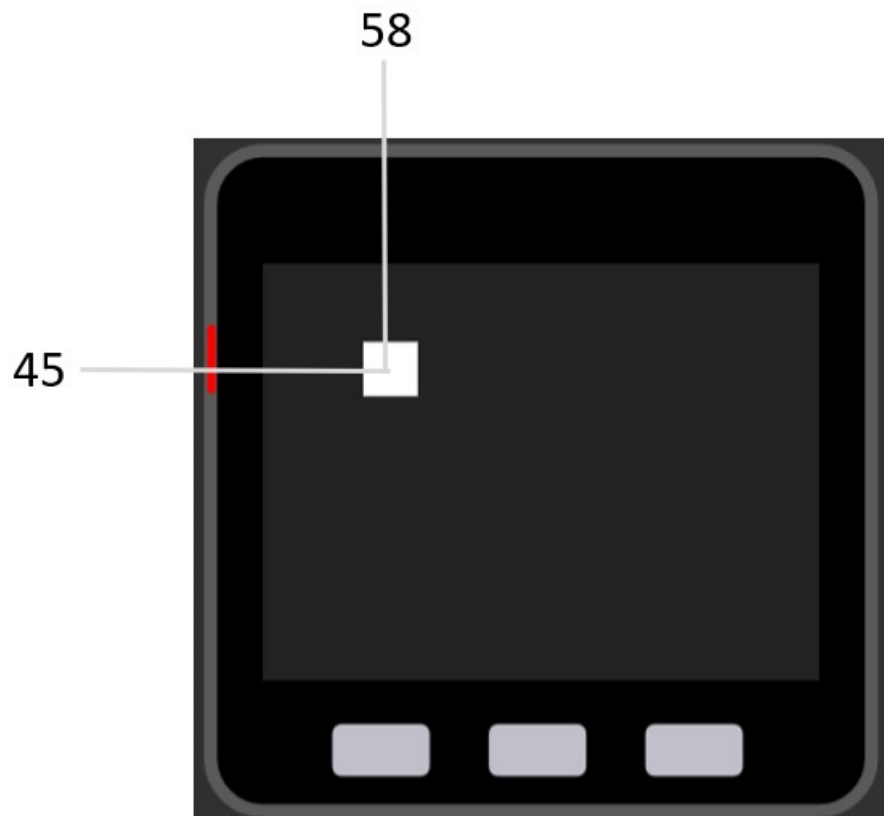












rectangle0

name: rectangle0

Units

x: 58

Modify: 45

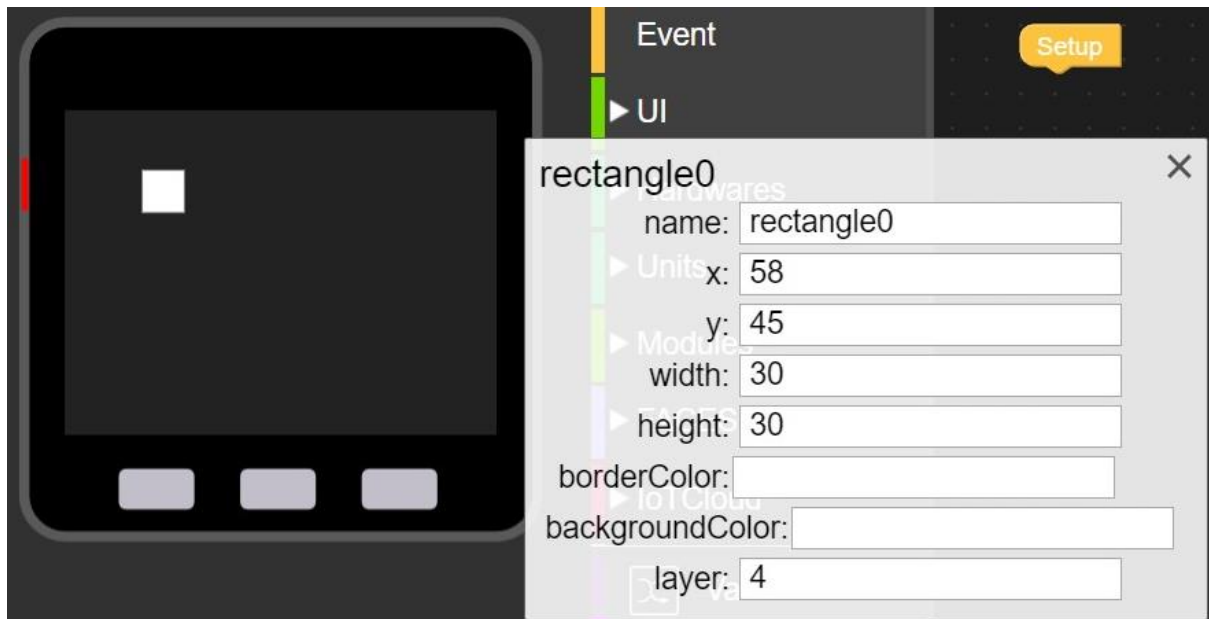
width: 30

height: 30

borderColor:

backgroundColor:

layer: 4



Wi-Fi

Description

Configure Wi-Fi network connection for M5 device





Title 1	Title 2
<input checked="" type="radio"/> Text entry <input type="radio"/> Text	<input type="text" value="Item 1"/> <input type="text" value="Item 4"/>
<input type="radio"/> Text entry <input checked="" type="radio"/> Text	<input type="text" value="Item 2"/> <input type="text" value="Item 5"/>
<input type="radio"/> Text entry <input type="radio"/> Text	<input type="text" value="Item 3"/> <input type="text" value="Item 6"/>
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>

SOFTWARE

UIFlow-Desktop-IDE

Download

M5Burner

Download

CP2104 Driver

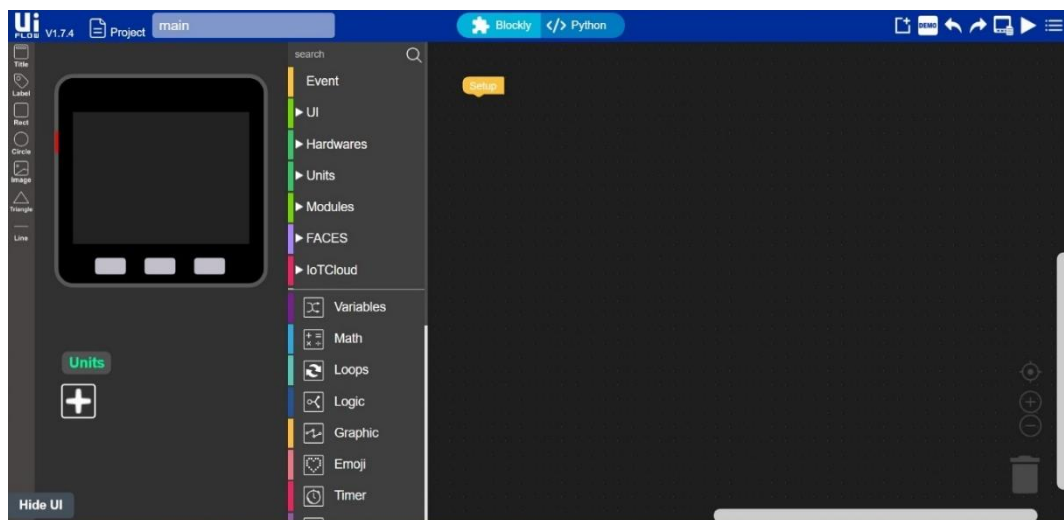
Download

Arduino-IDE

Download

M5Stack Library

Download



Project Title

Enter your project name here. Whenever you download a program to your computer or M5Stack Core it will retain this name.

Block Python

This button allows you to see the Python code that your blocks have generated and edit the code.

Menu Tab

From this bar you can access the forum, documentation, example undo and redo actions, upload files to the M5Stack Core, run your code on the device and alter the settings.



Hide UI

Hide the UI Manager panel to free up more space in the coding area.

UI Preview

Drag text and visual elements on to the M5Stack Core to create a UI. Python code will be generated and blocks will appear in the UI panel to allow you to manipulate these elements.

Code blocks menu

Here you can find all the blocks you need to create programs.

Coding Area

The coding area is where we drag blocks to build your program.

Units

Here you can pick which units you want to use in your project and which port you plan to connect them to.

Event
Here you can find Loop and button press event blocks.

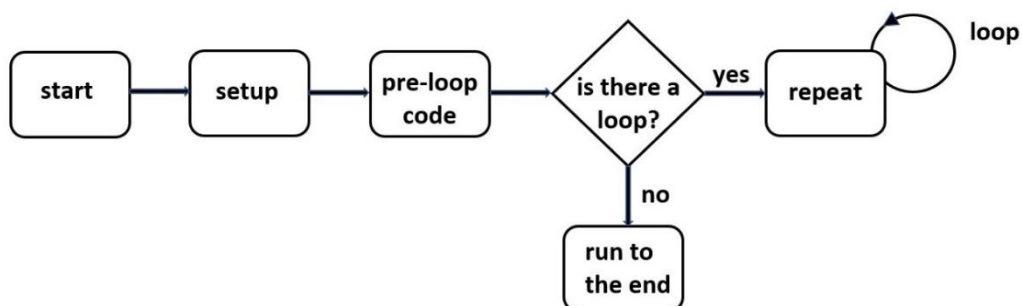
Hardwarees
Program the internal peripherals of the M5GO such as the RGB Bar, Speaker, Accelerometer and Power management.

Units
Whenever you add a unit, it will appear hear along with all the code blocks related to it.

Math
Maths is essential in programming. Here you will find all the blocks necessary to make both simple and complex calculations.

Logic
Every program needs logic to decide which action to take when an event occurs.

Advanced
Advanced blocks for experienced coders. You'll find blocks for networking, digital/ analog pin control and more here.



Setup

The setup block is essential for any program to run. It defines the first thing that will happen when the code is uploaded or the device is switched on. It will only run once.

Setup

Loop

The loop block will run any code placed inside it indefinitely. That means unless you turn off the device it will continue to run without stopping.

Loop

Wait

The wait block will delay your program for however many seconds you input. Sometimes this is necessary to see the result of some code that might have otherwise run so fast that you blinked and missed it.

Wait 1 s

Forum

This links directly to the official forum where you can share your code or get help if you're stuck.

Docs

The Documentation pages have all kinds of information about each unit and the functions of uiflow.

Run

This button is intended for quick testing. It will run the code on your device, but it won't save it. If you want the code to remain on the device use "Download."

Example

Here you can find some example programs to adapt and learn from.

Undo/Redo

If you make an error, don't worry, you can always trace your steps back.

File Manager

It's possible to upload .m5f M5GO program files and pictures to the M5GO. The manager helps us to do this.

Download

Want to download your program to the device so you can show your friends. Press download and then look for your program in the app list on boot.

Settings

Here you can setup the device's connection, change your language, device and color scheme.

The diagram illustrates the M5GO application interface. At the top, a row of icons represents different features: a speech bubble for the Forum, a document for Docs, a folder for Example, undo and redo arrows for Undo/Redo, a file folder for File Manager, a download arrow for Download, and a play button for Run. Below these icons is a dark blue menu with four options: 'Open' (with a folder icon), 'Save' (with a floppy disk icon), 'Download' (with a download arrow icon), and 'Setting' (with a gear icon). Red lines connect each of these menu items to a corresponding text box on the left or right side of the interface, providing detailed explanations for each function.

Power
ON
button



1



2



3

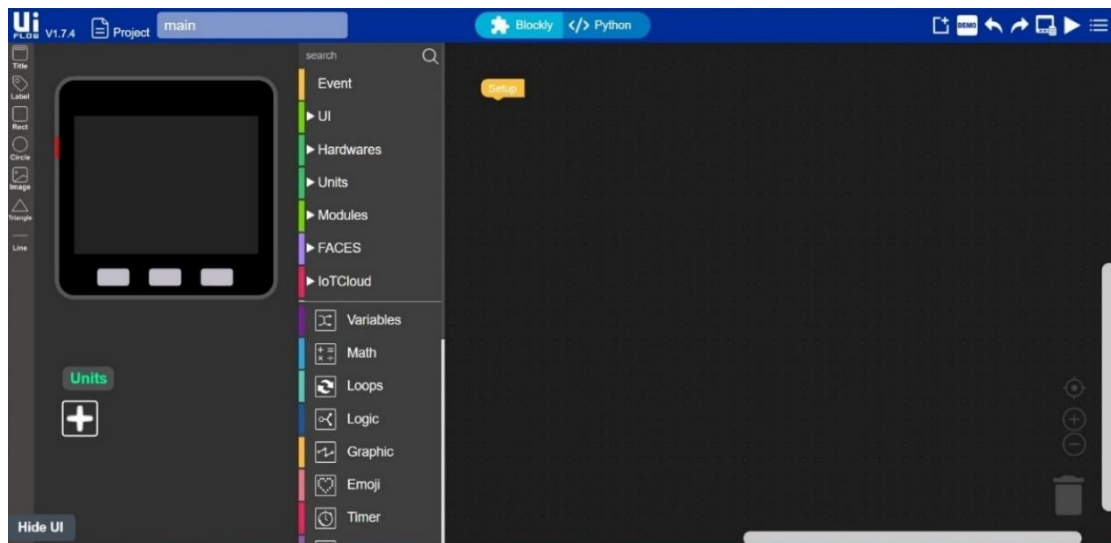


4

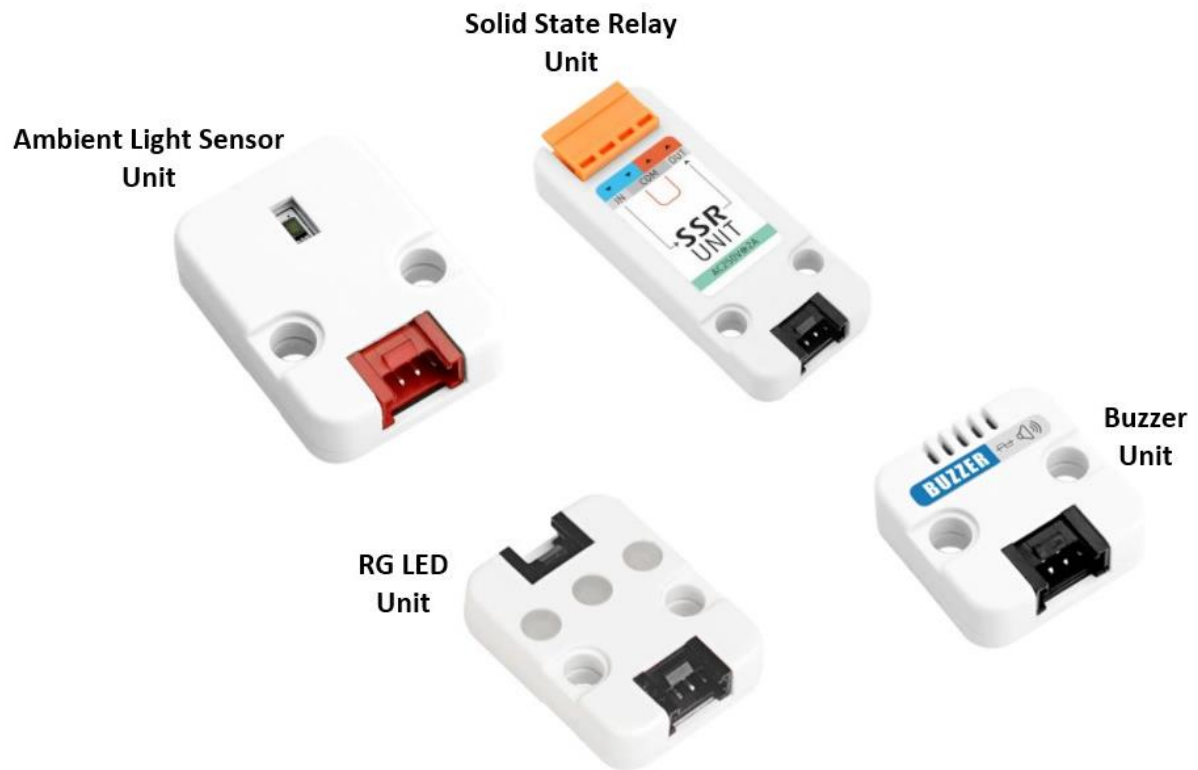


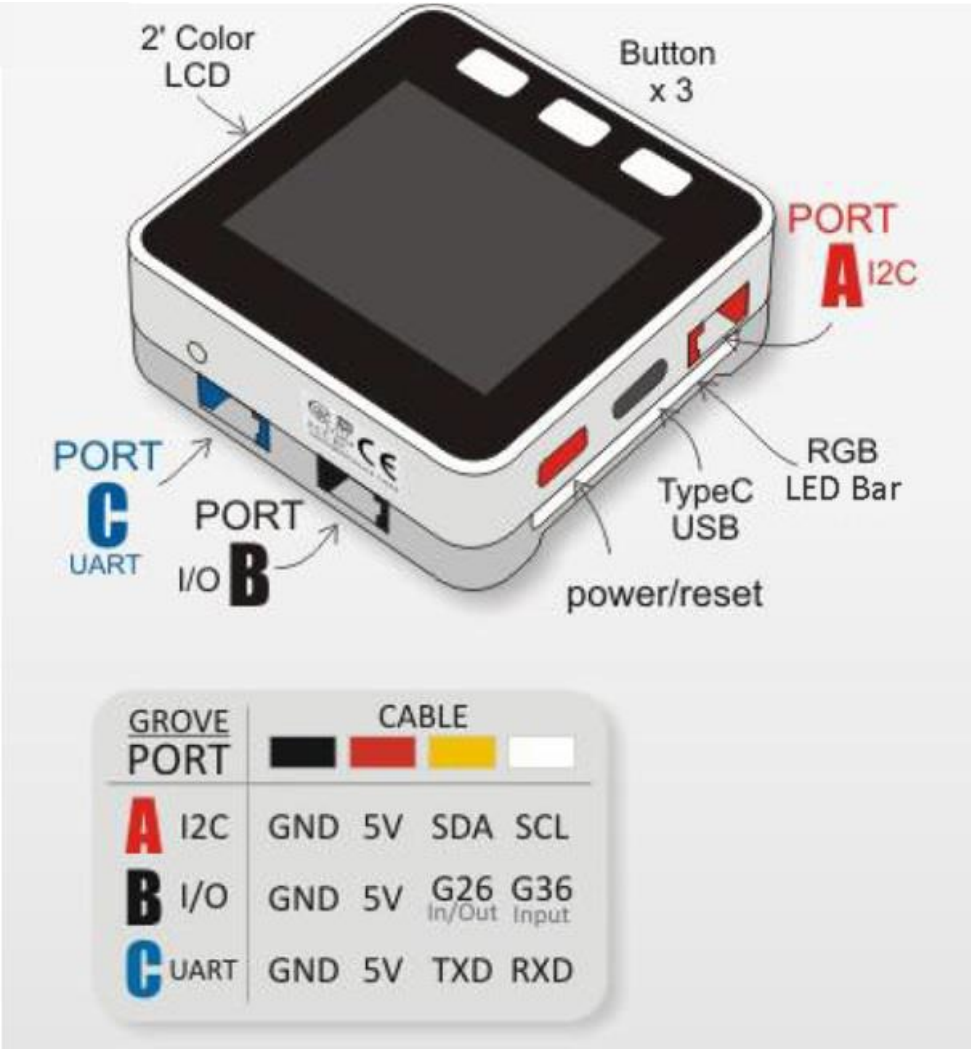
Selected
Communication port

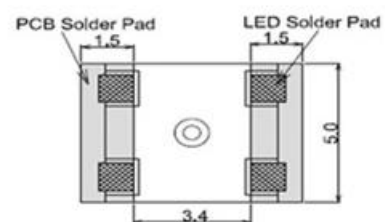
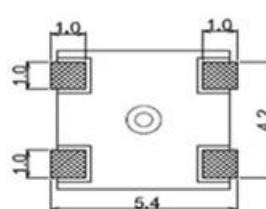
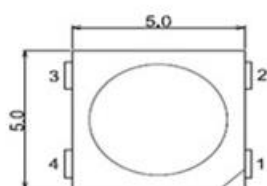
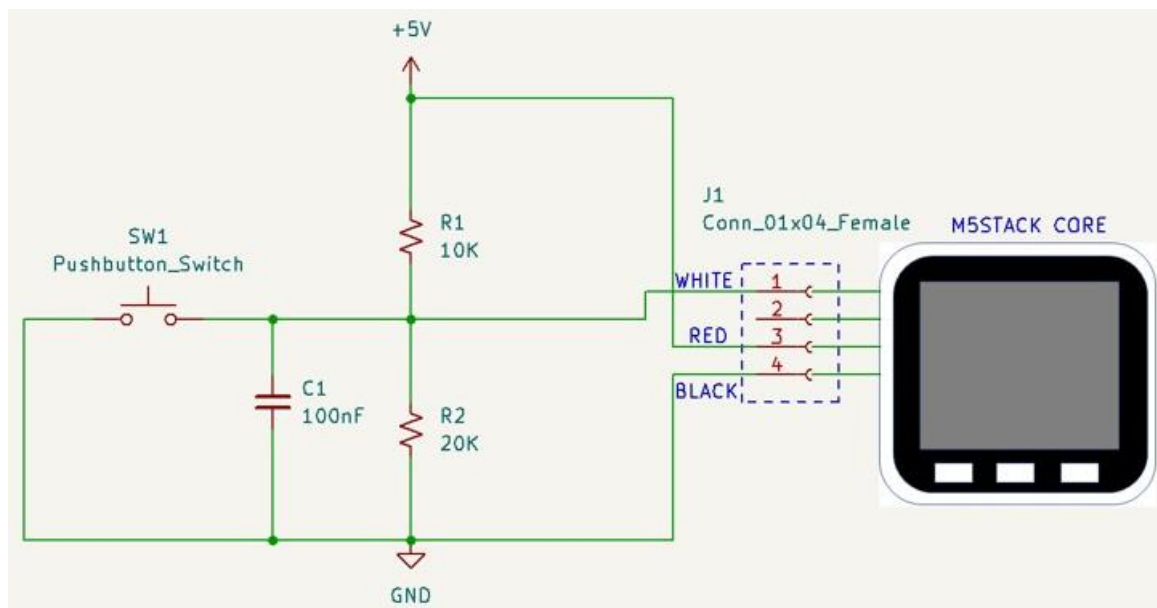
Selected M5Stack Core



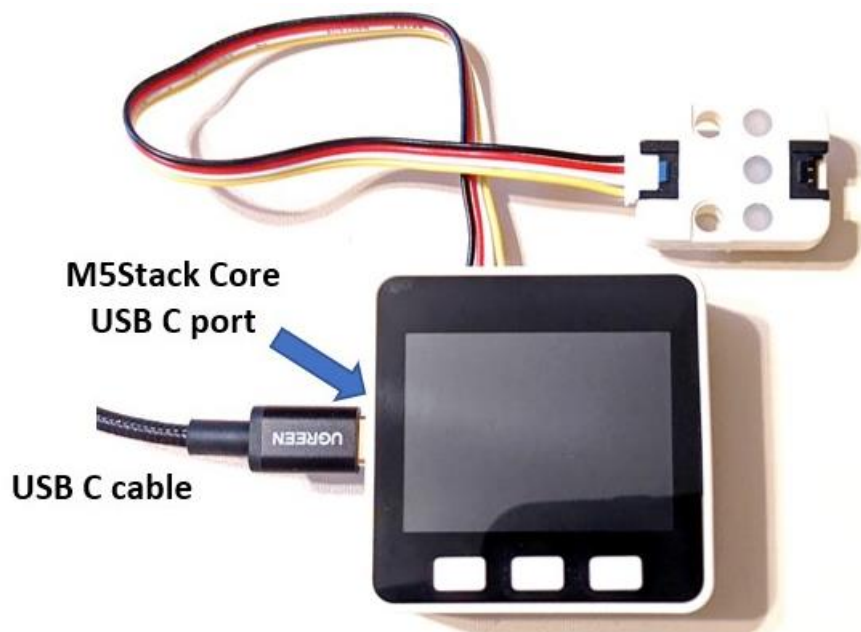
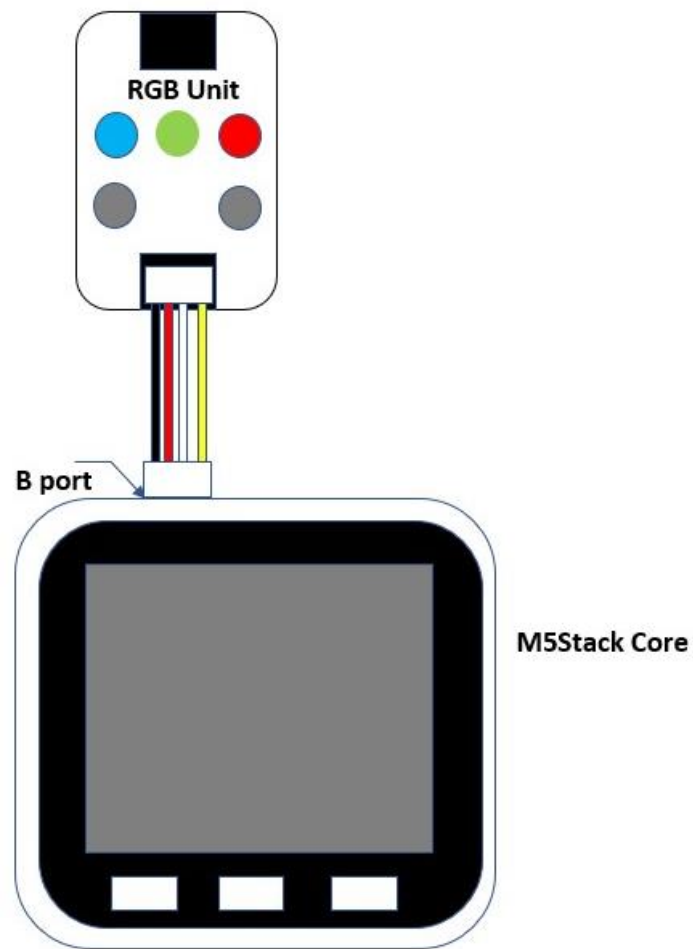
Chapter 2: Hands-On with M5Stack Units



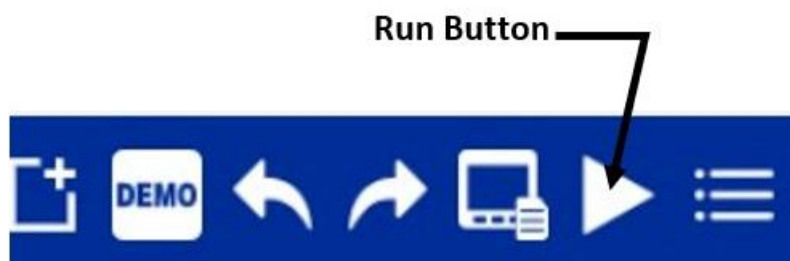
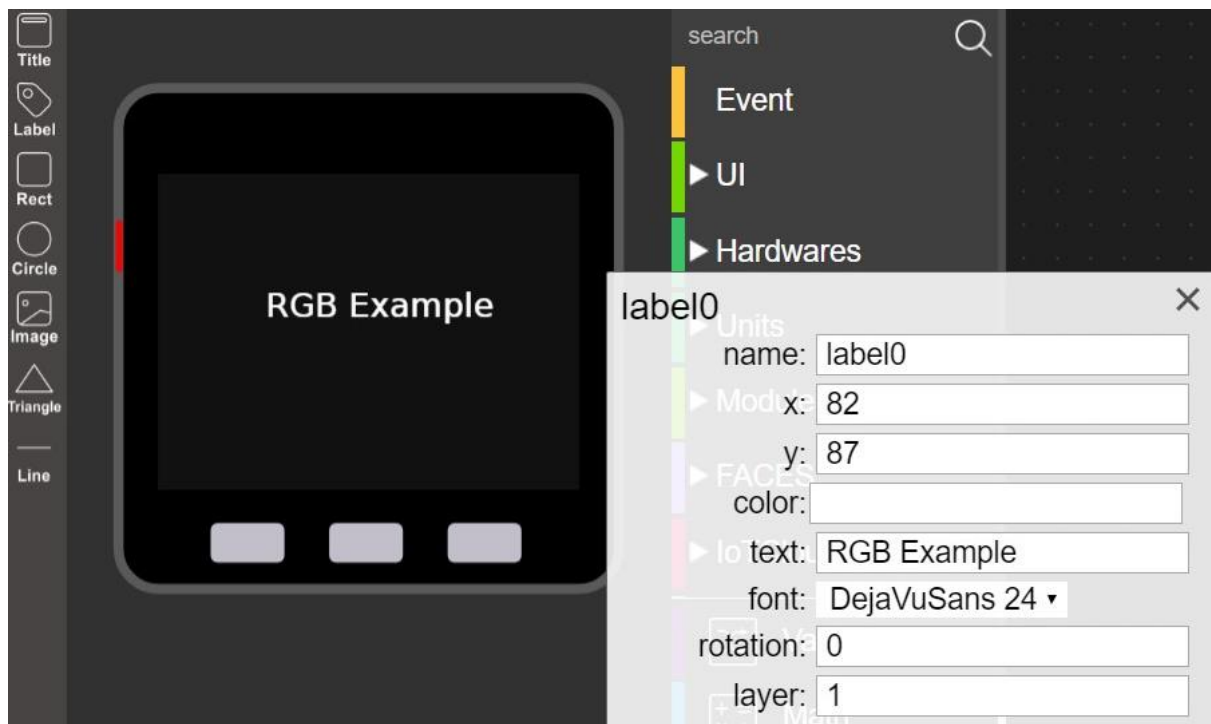


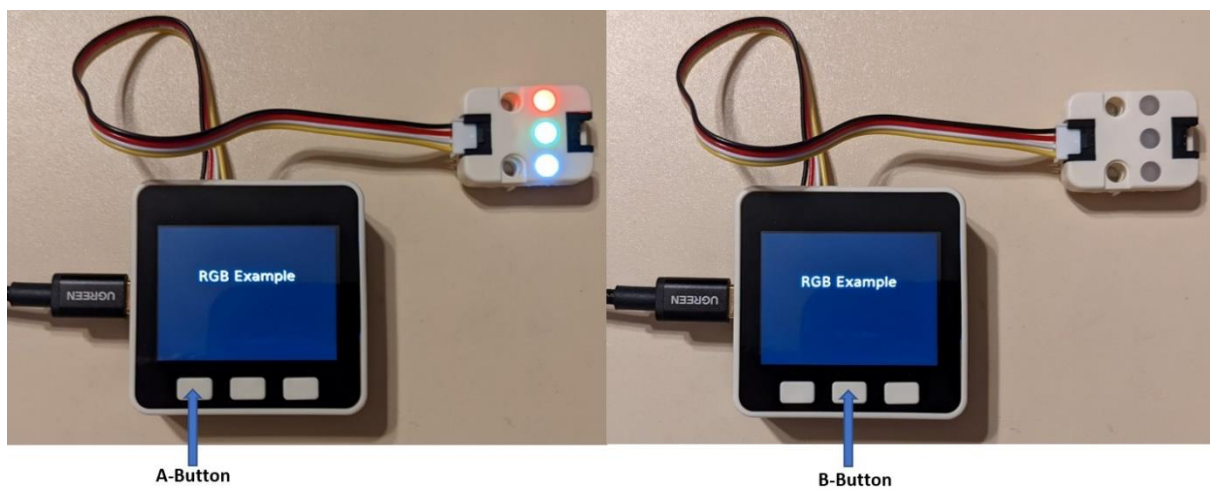
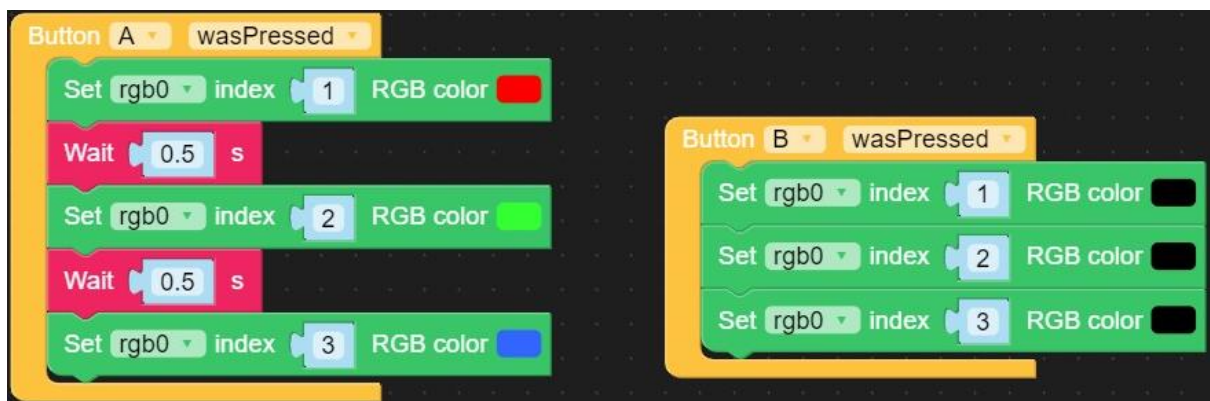


Dimensions in millimeters (mm)

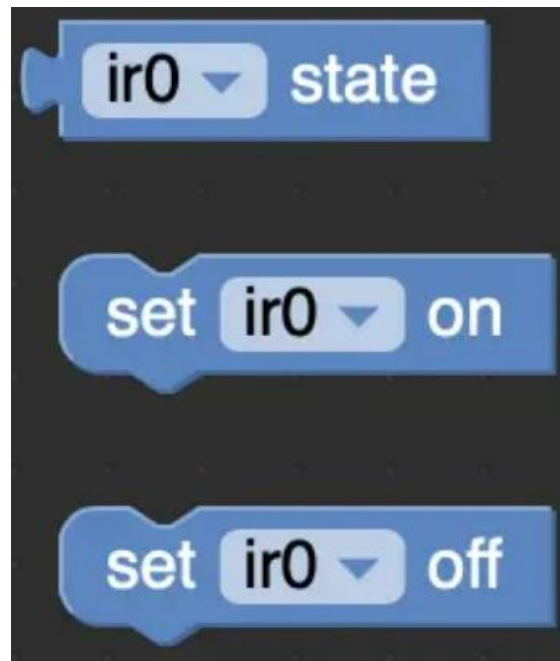
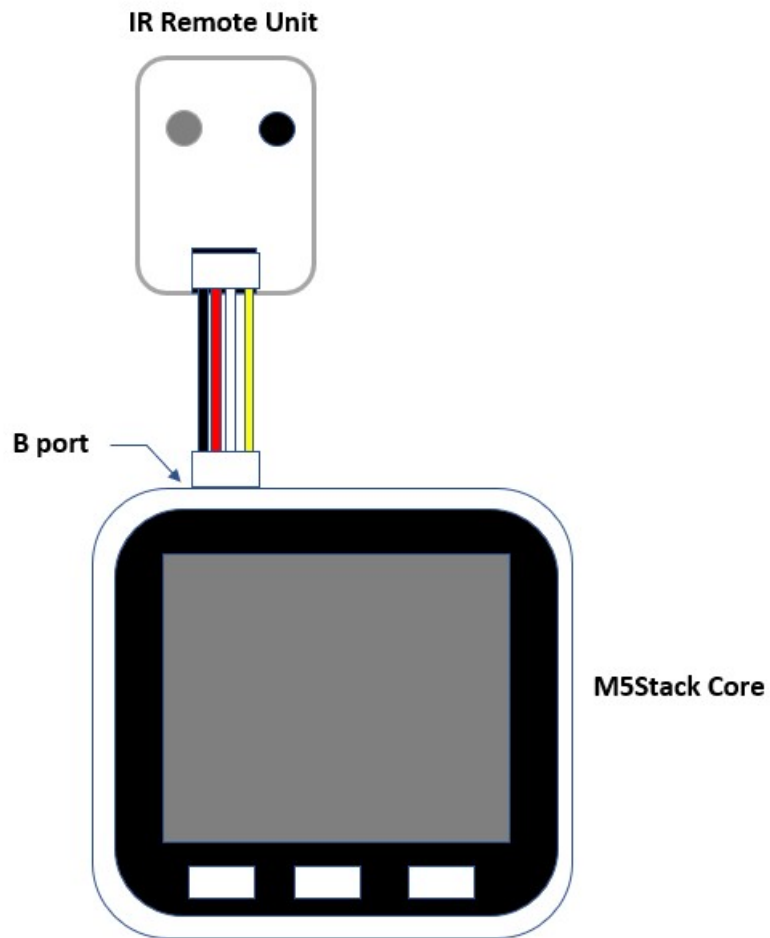


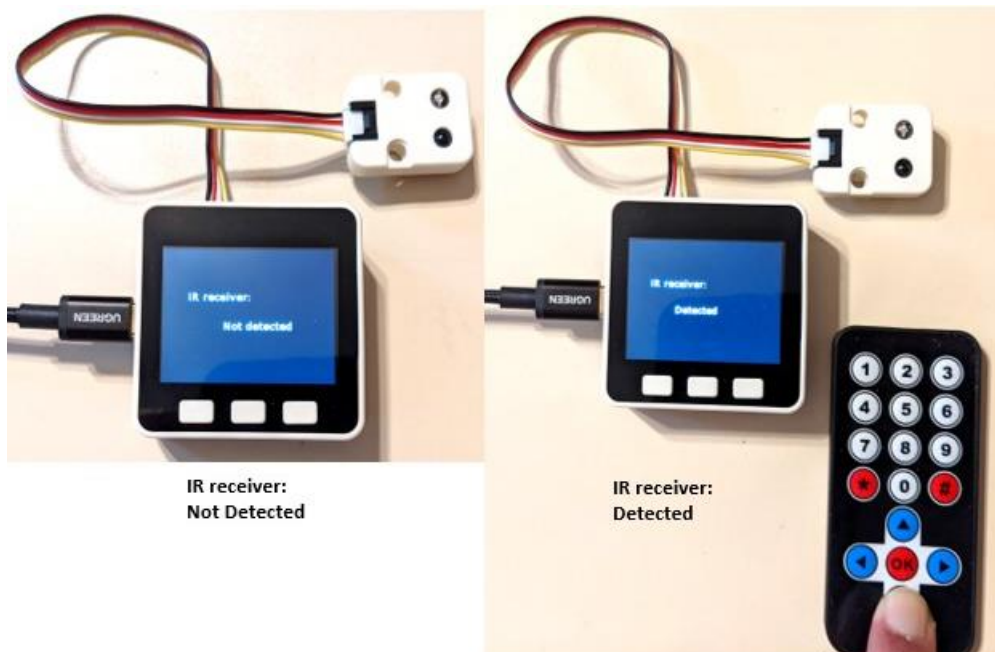
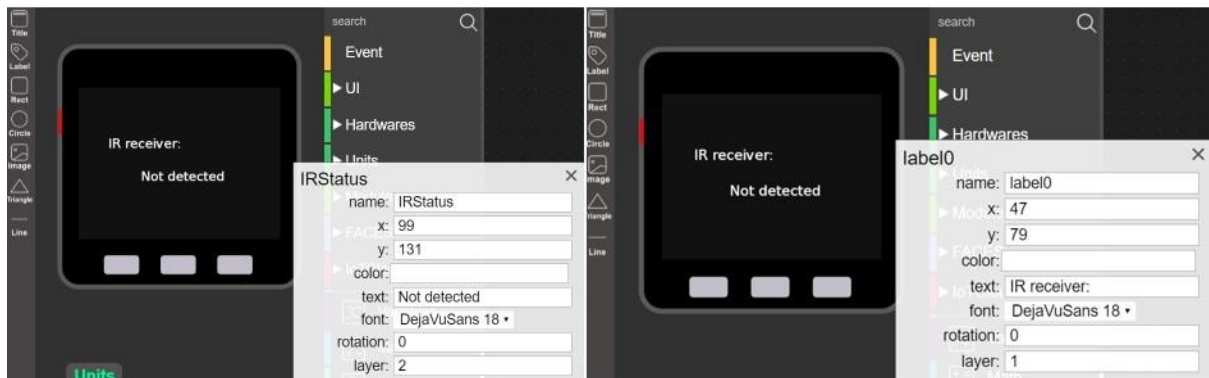
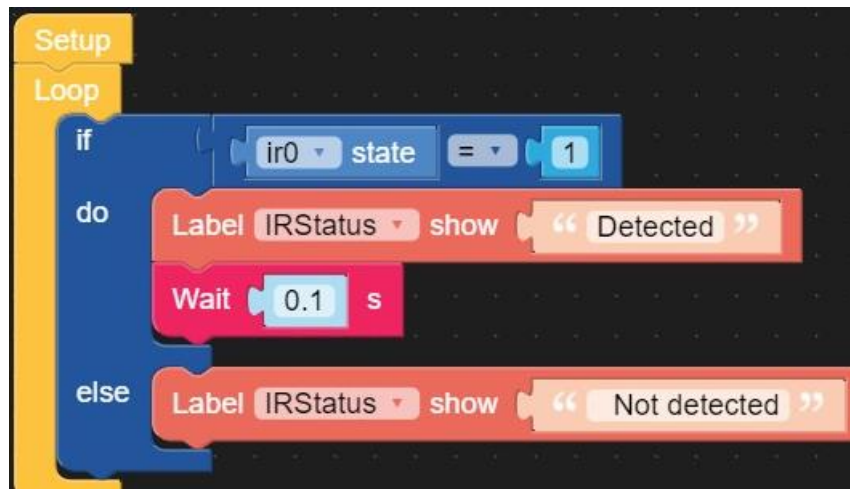


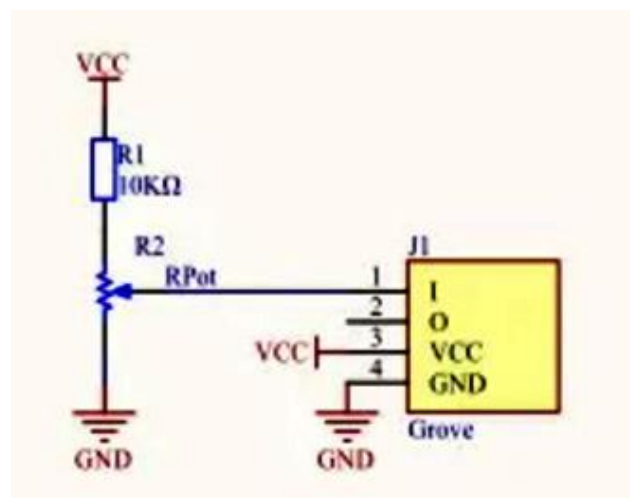


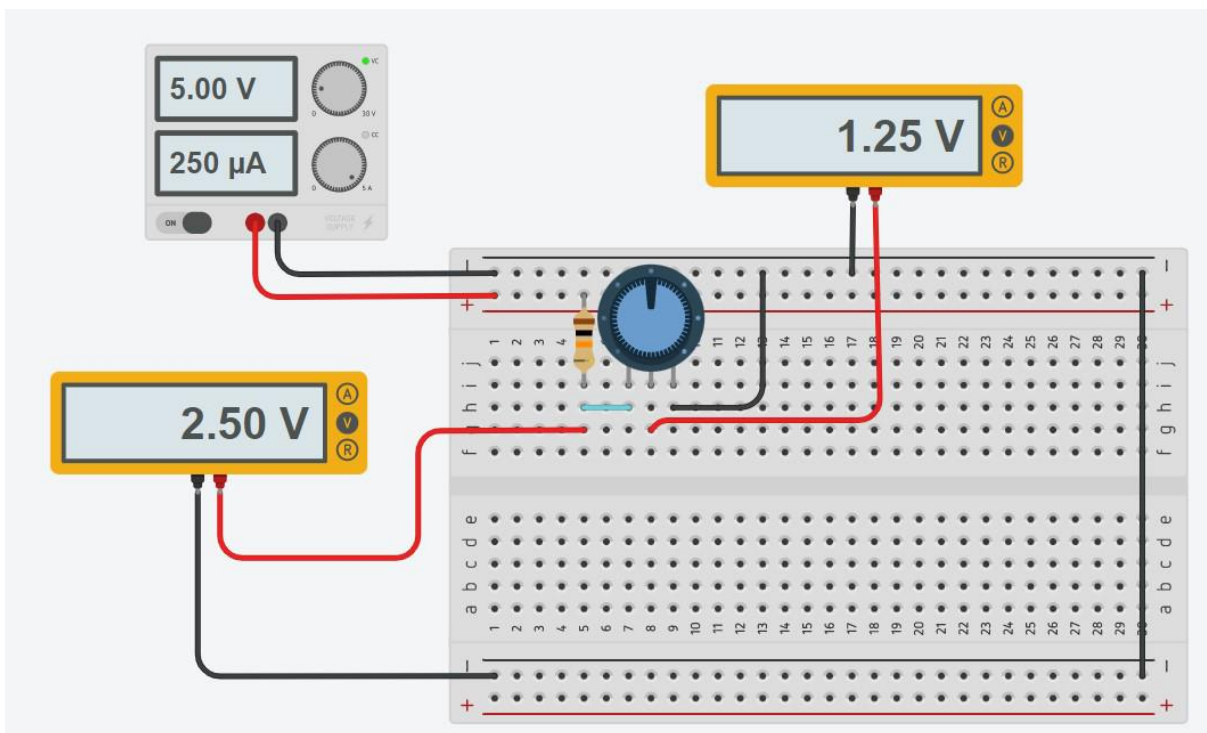
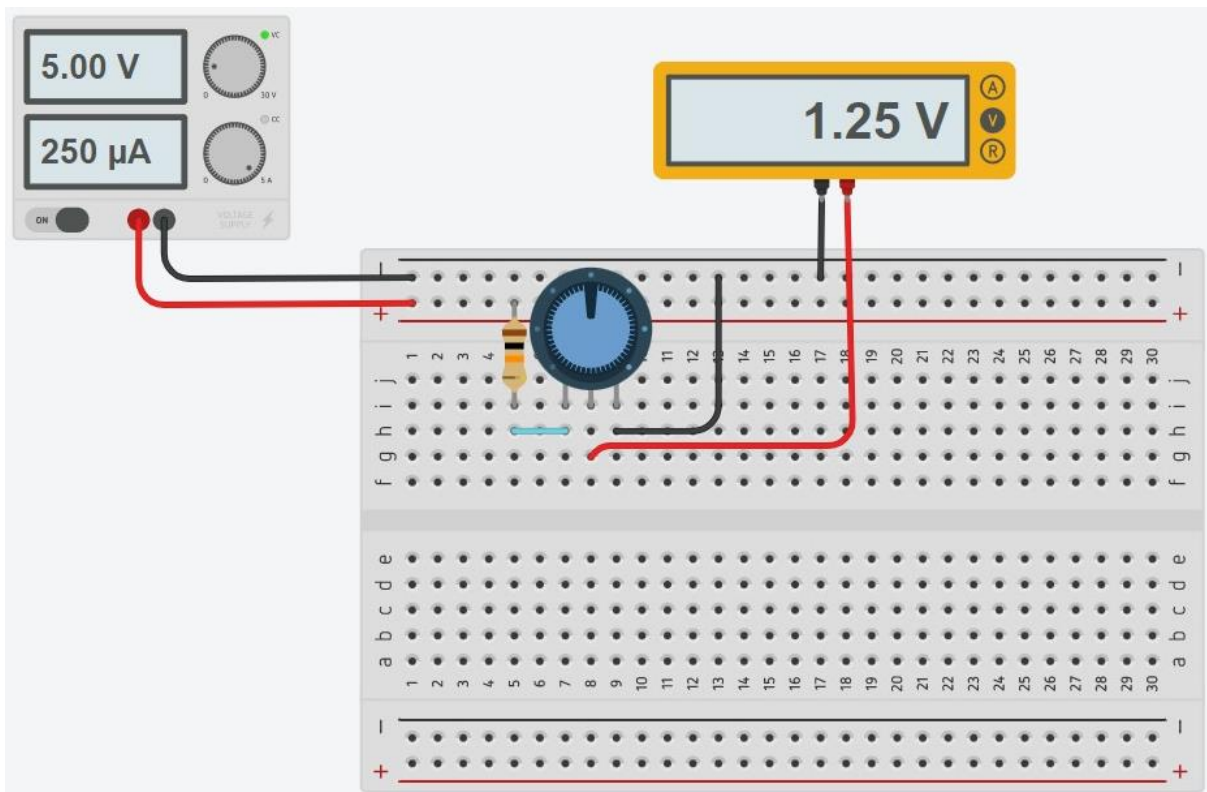


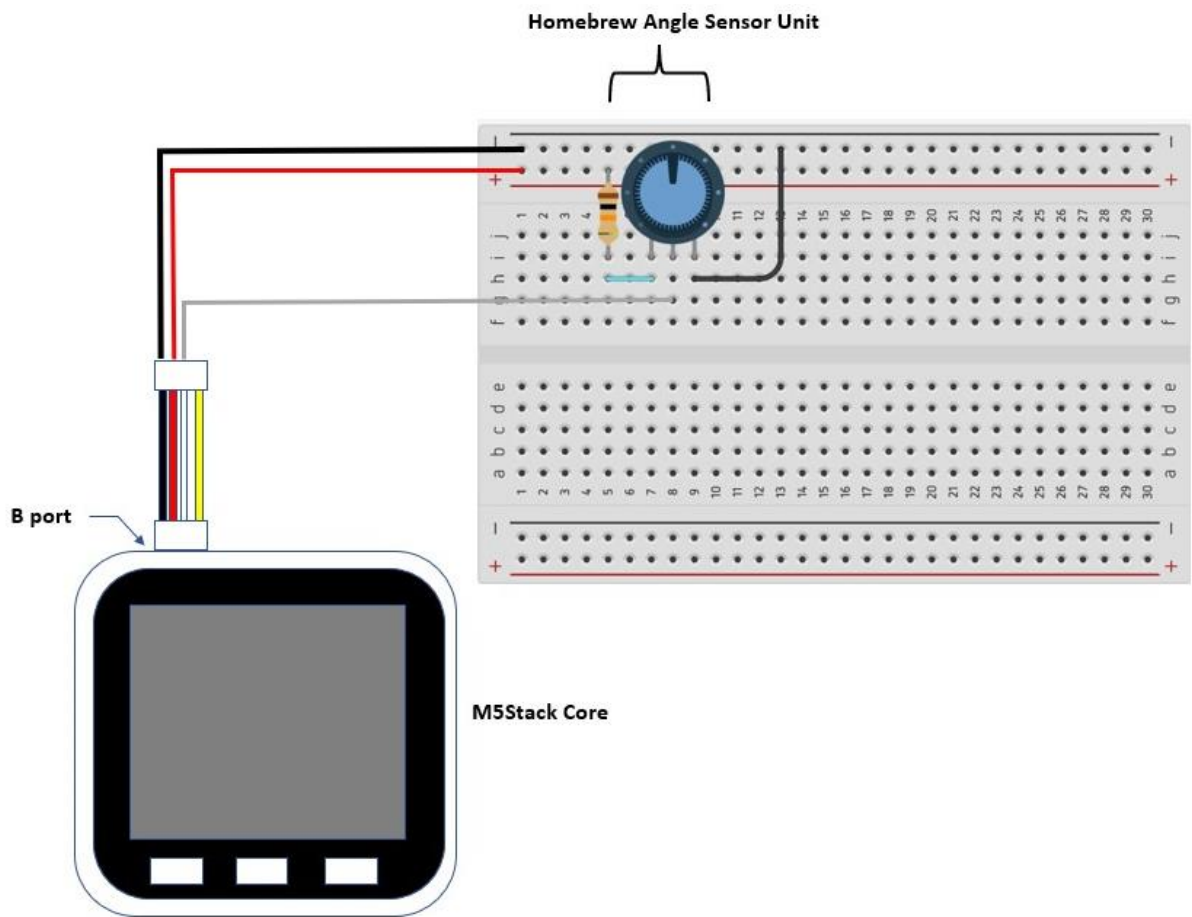


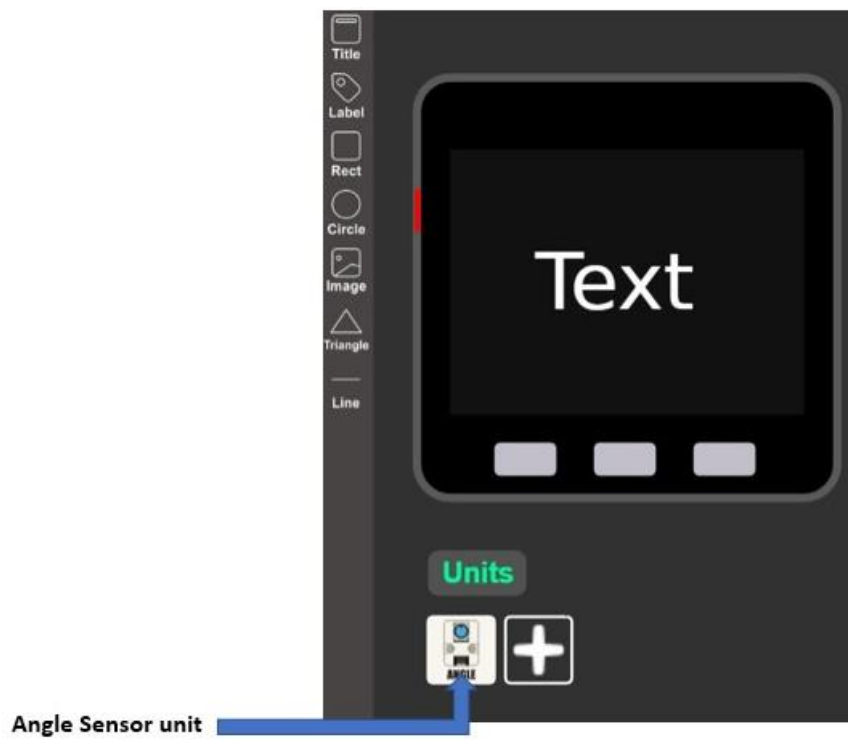
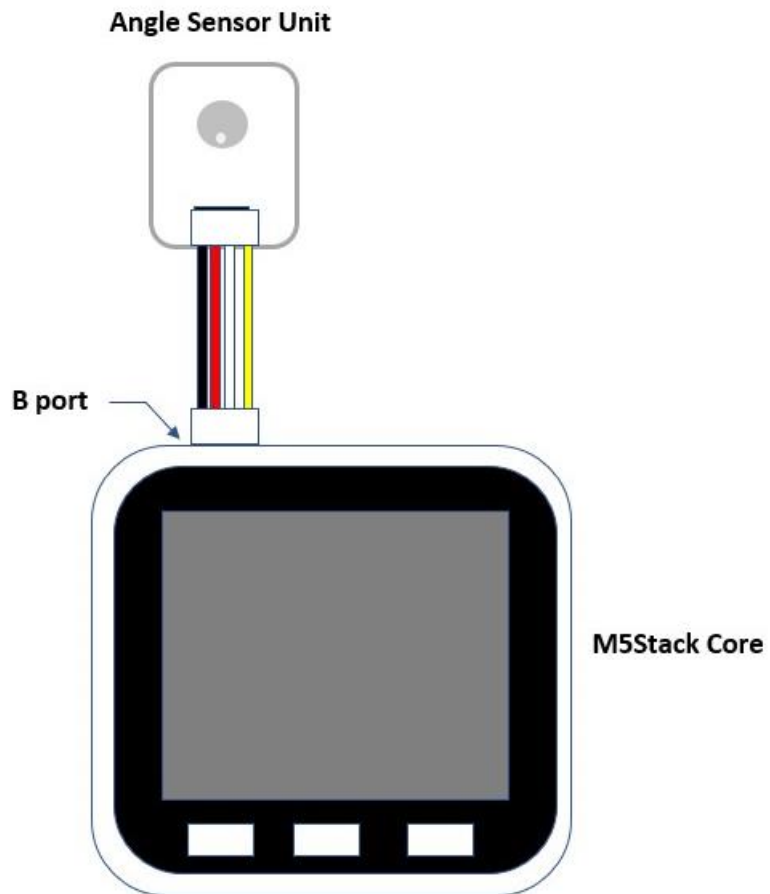


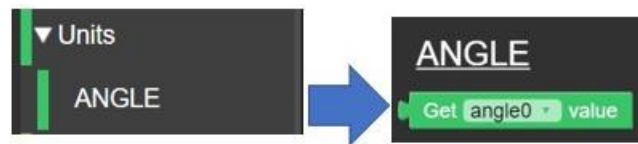


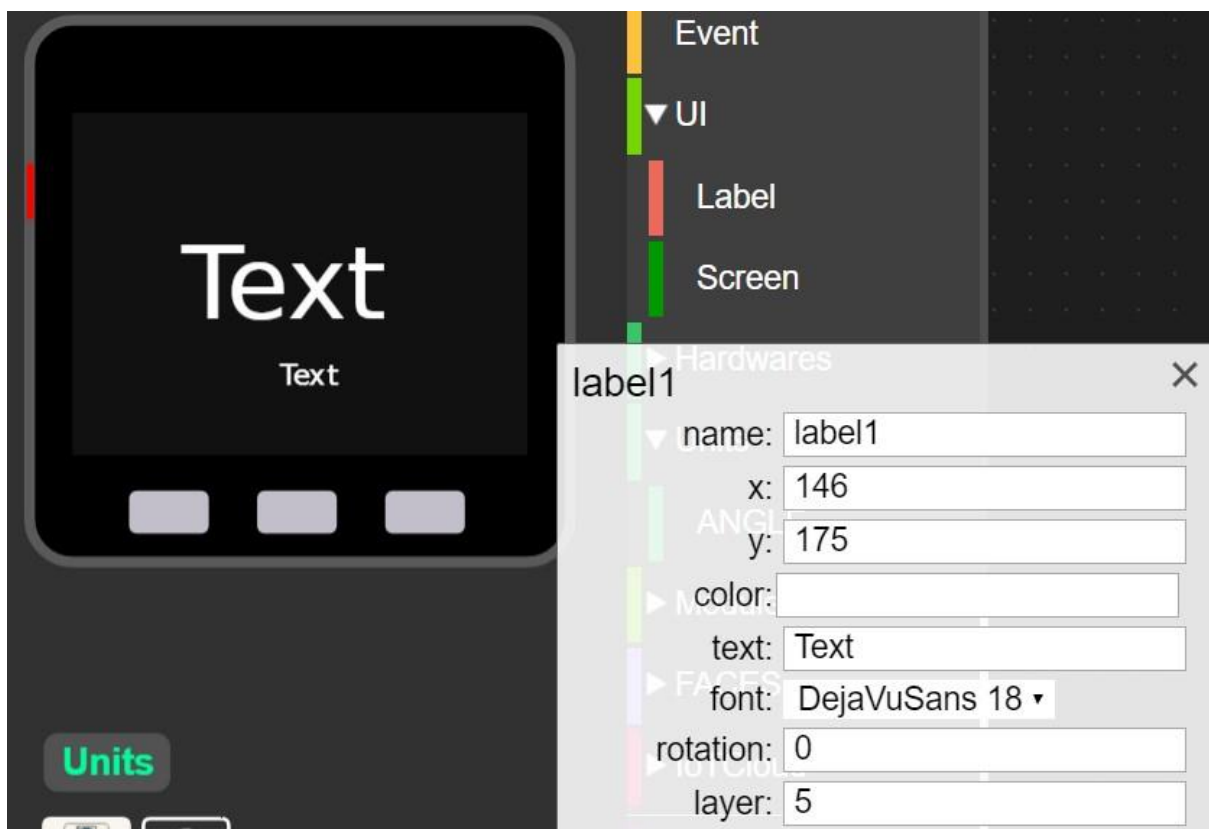
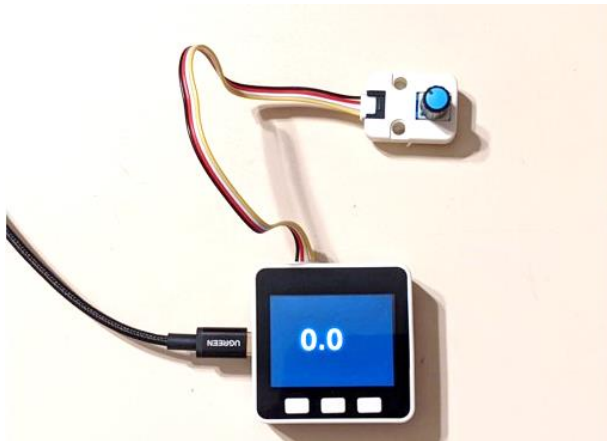


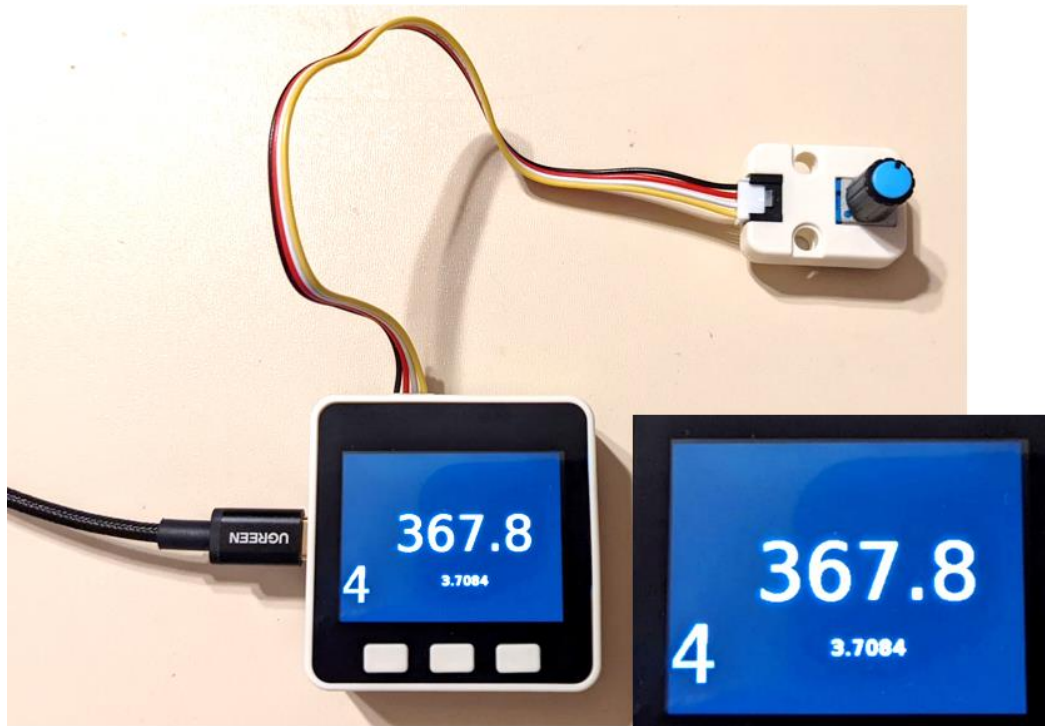




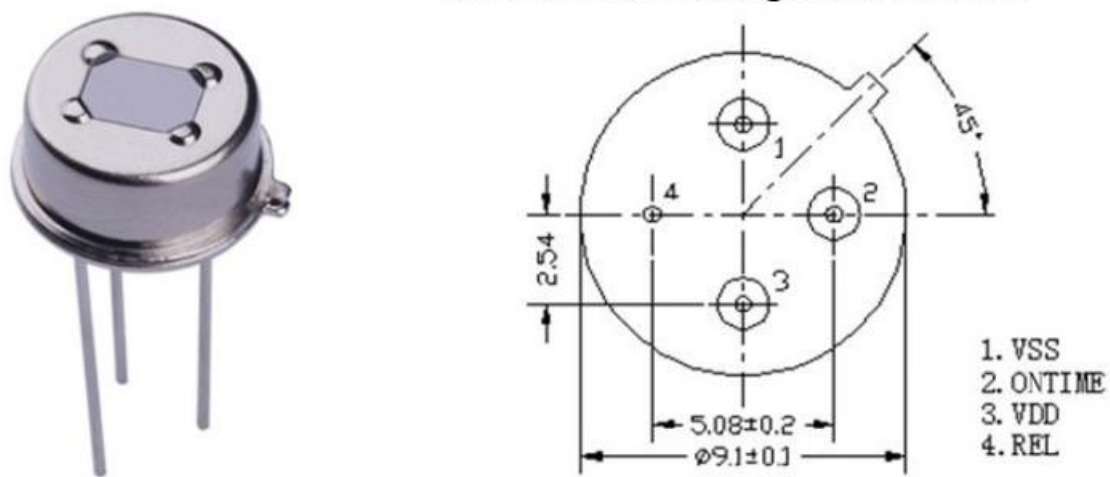




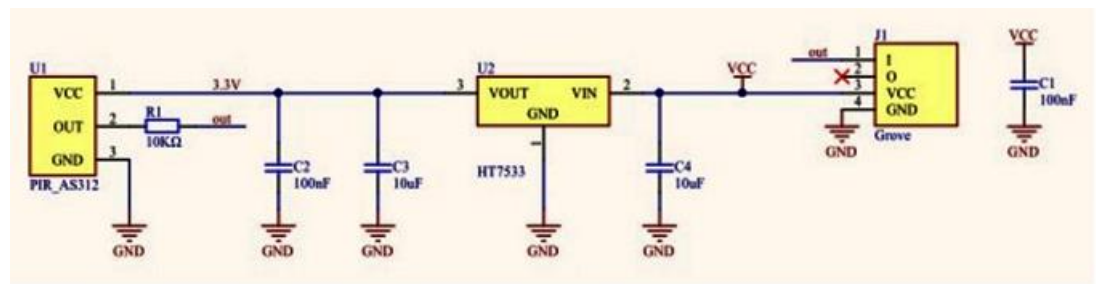




Mechanical Package with Pinout

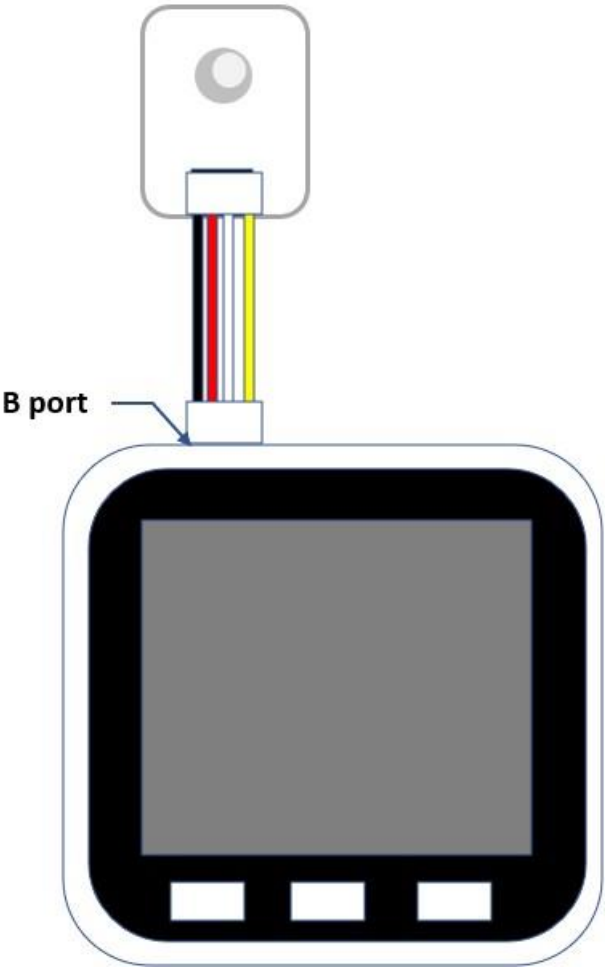


Dimensions in millimeters (mm)

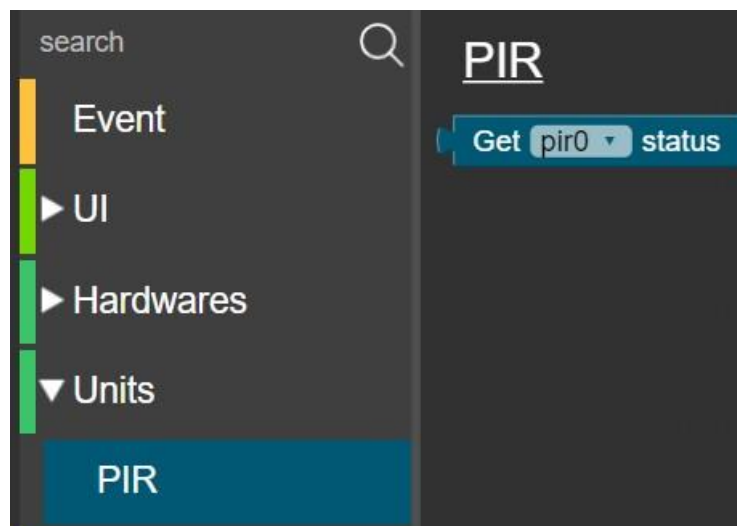
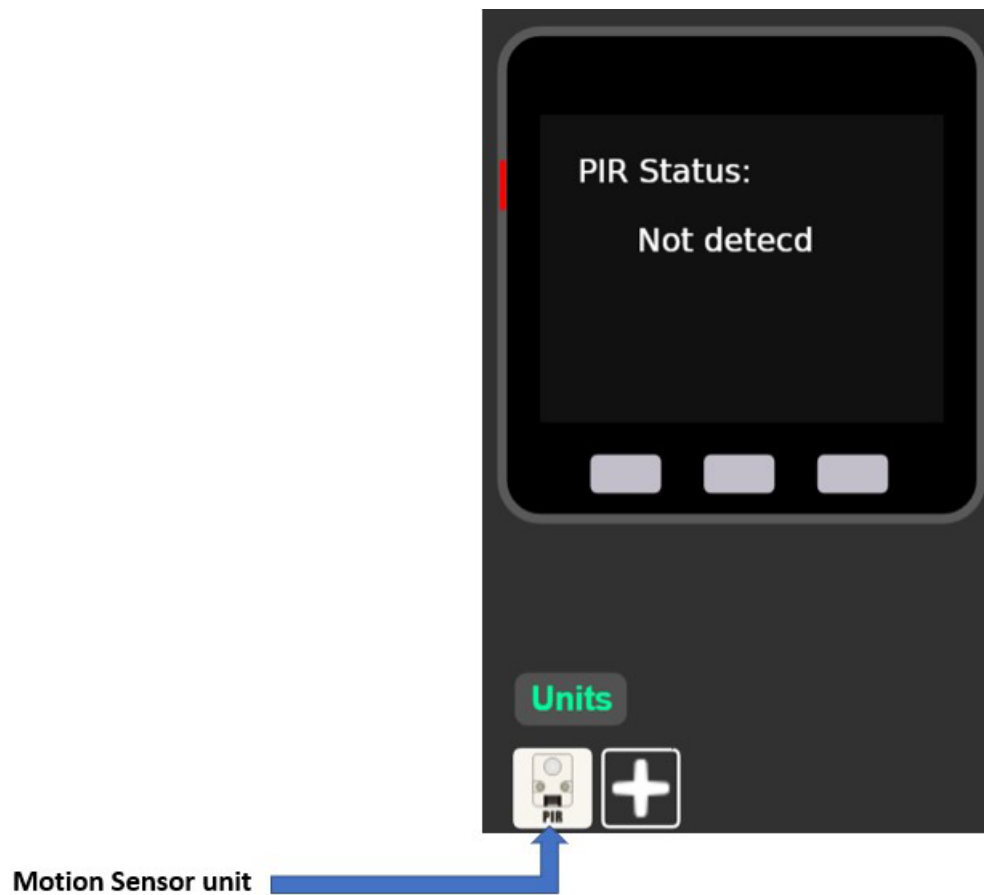


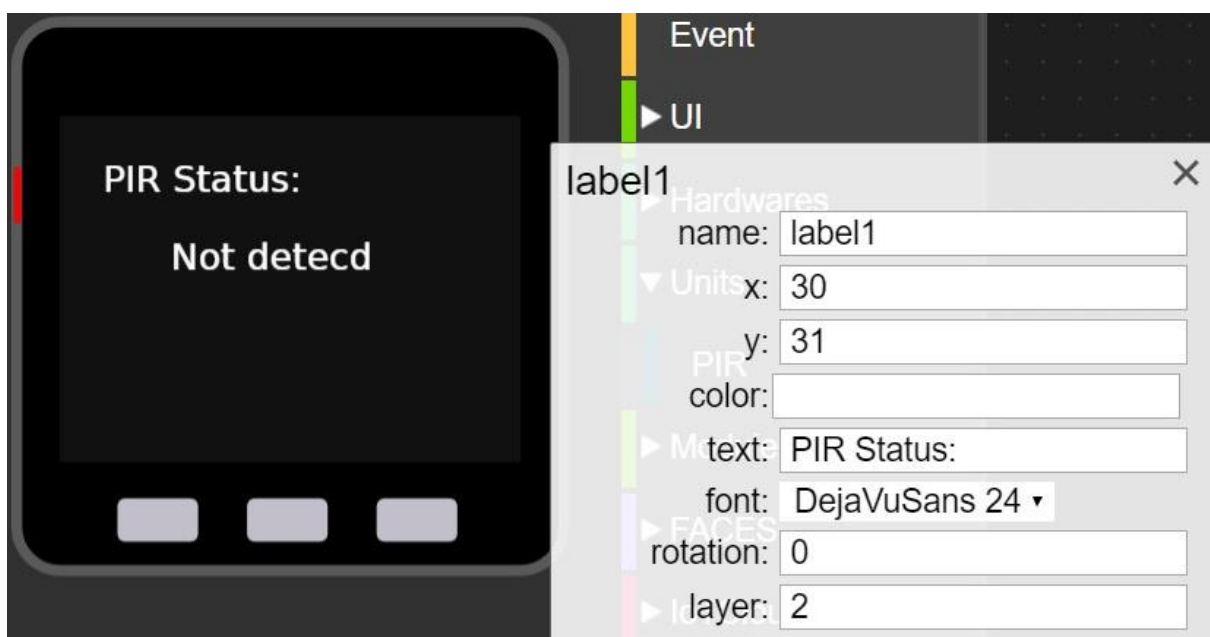
M5Core(GROVE B)	GPIO36	GPIO26	5V	GND
PIR Unit	Sensor Pin		5V	GND

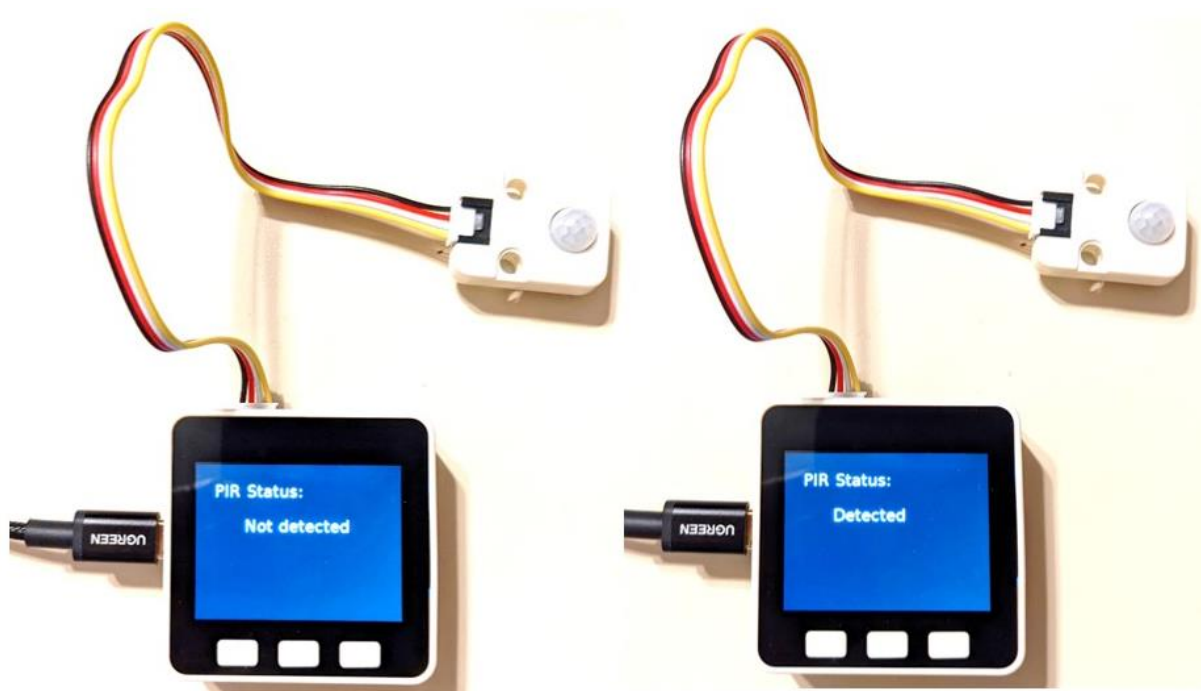
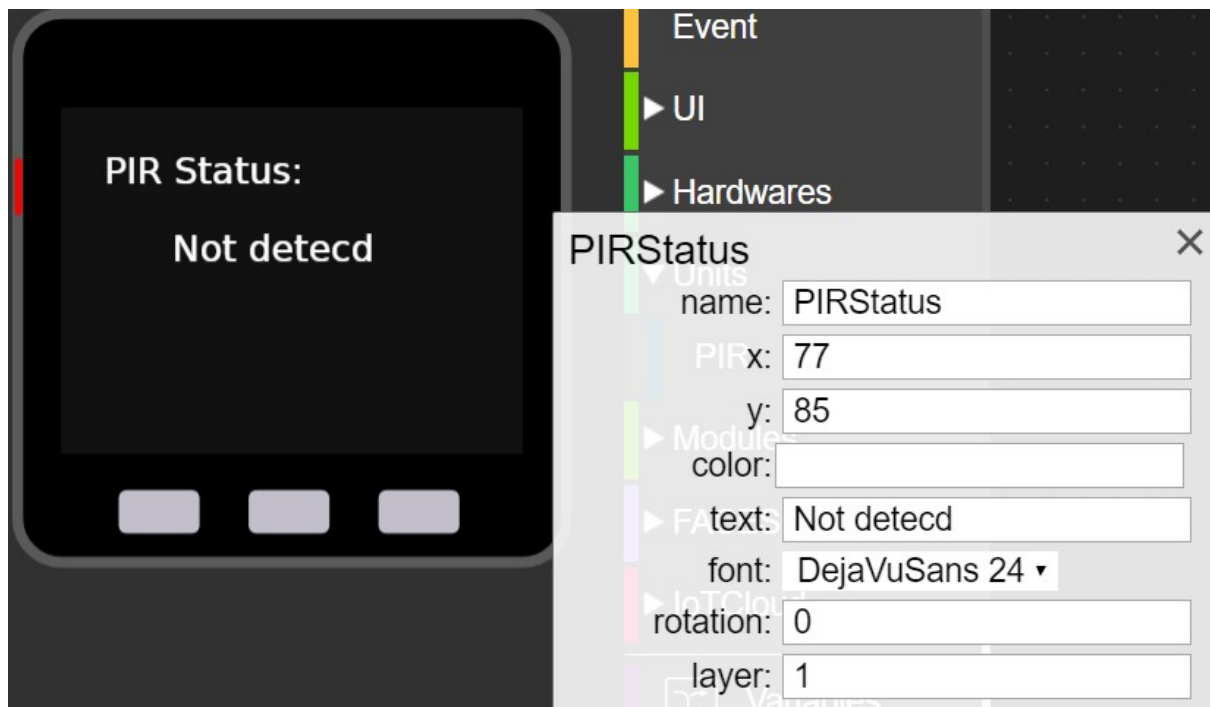
Motion Sensor



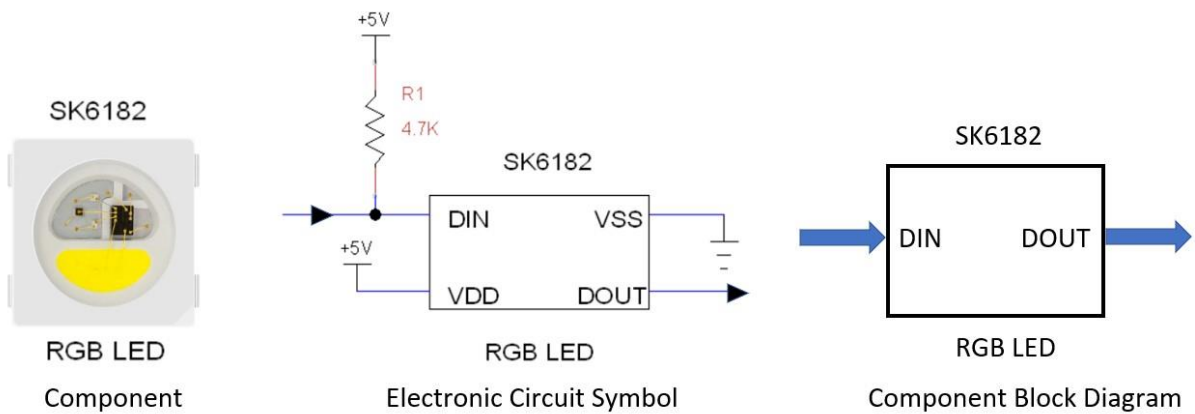
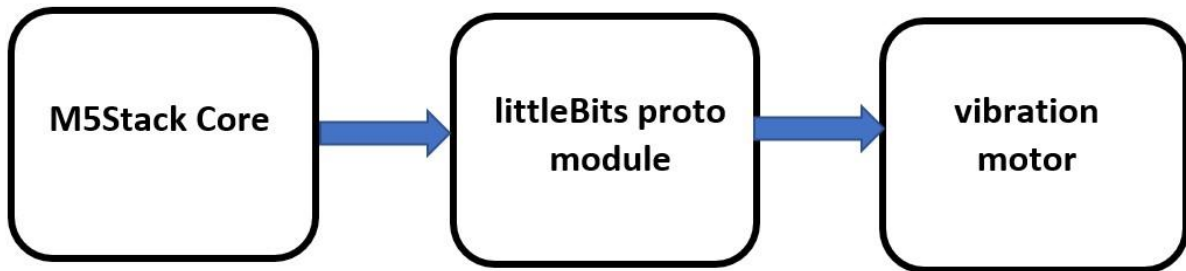
M5Stack Core

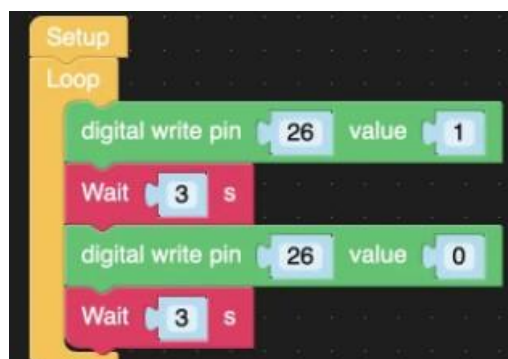
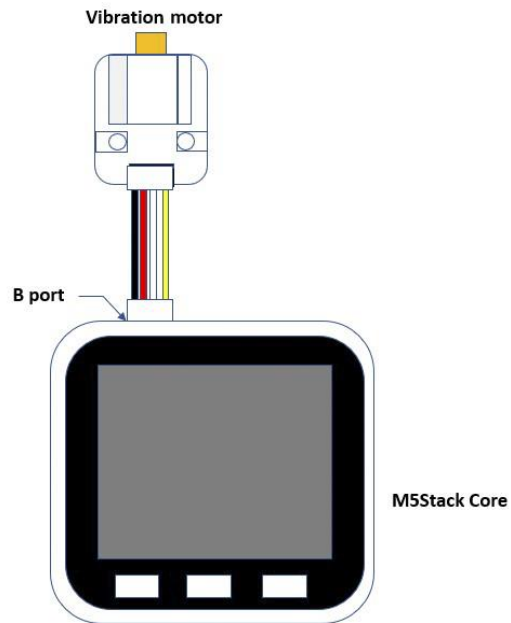


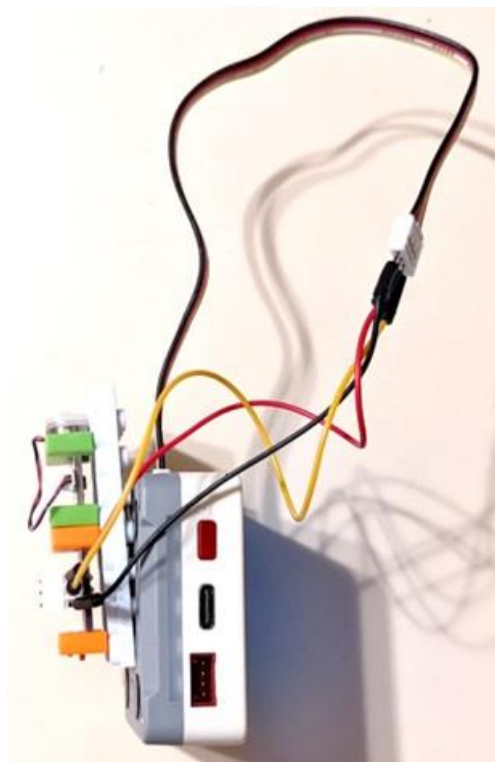
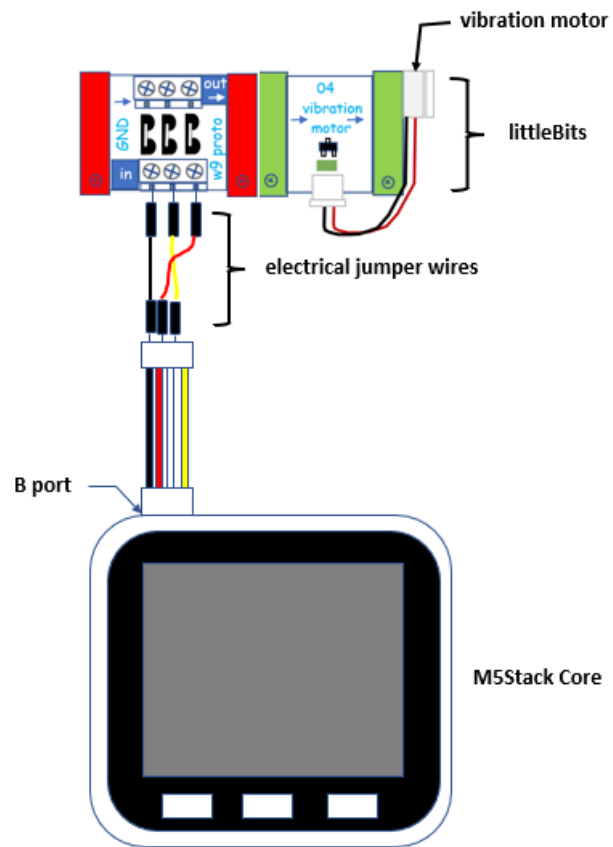


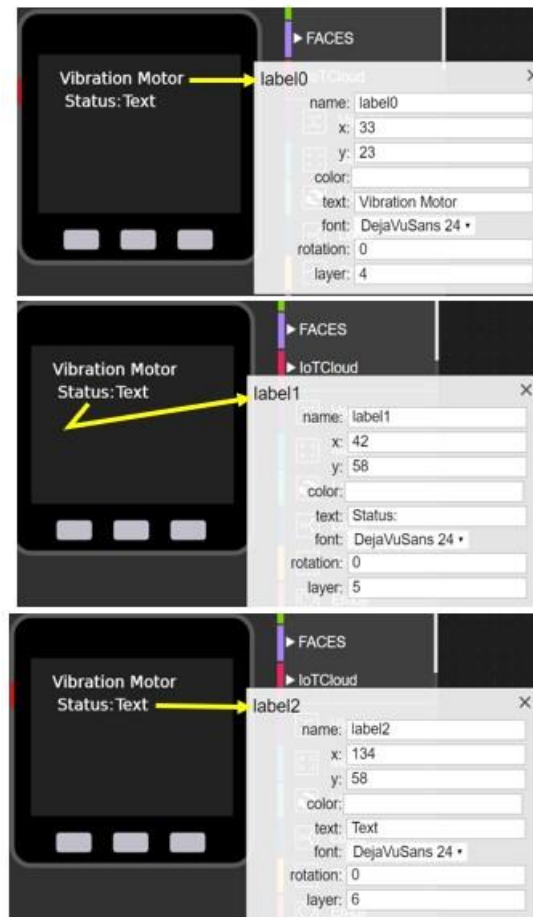


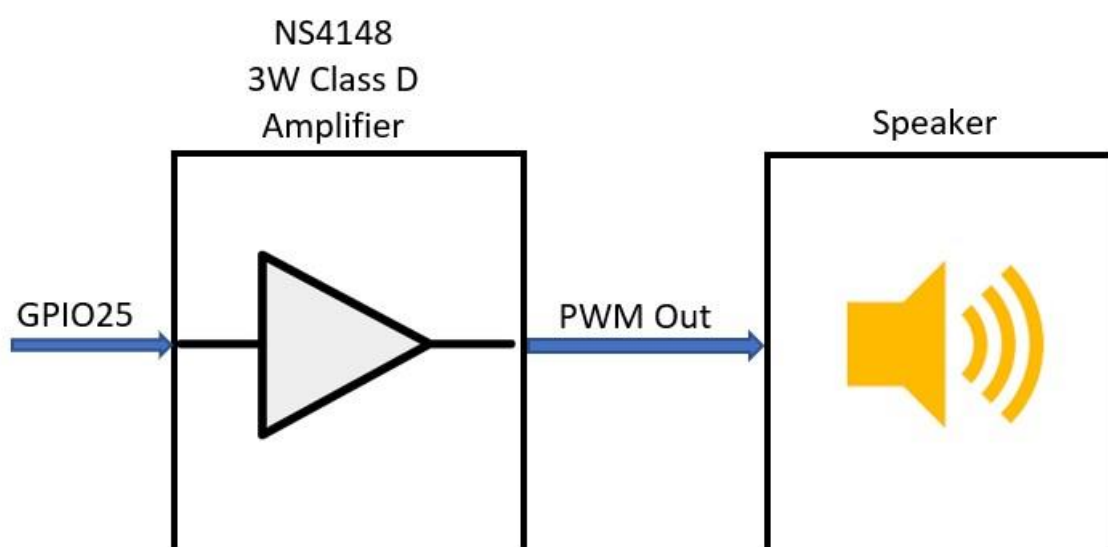
Chapter 3: Lights, Sound, and Motion with M5Stack

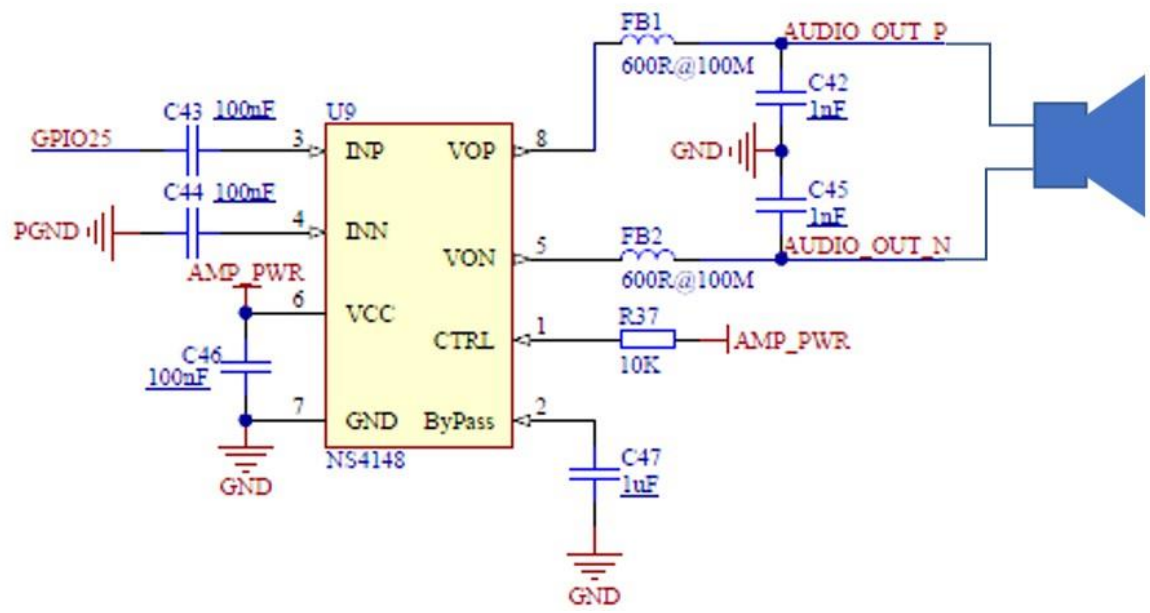










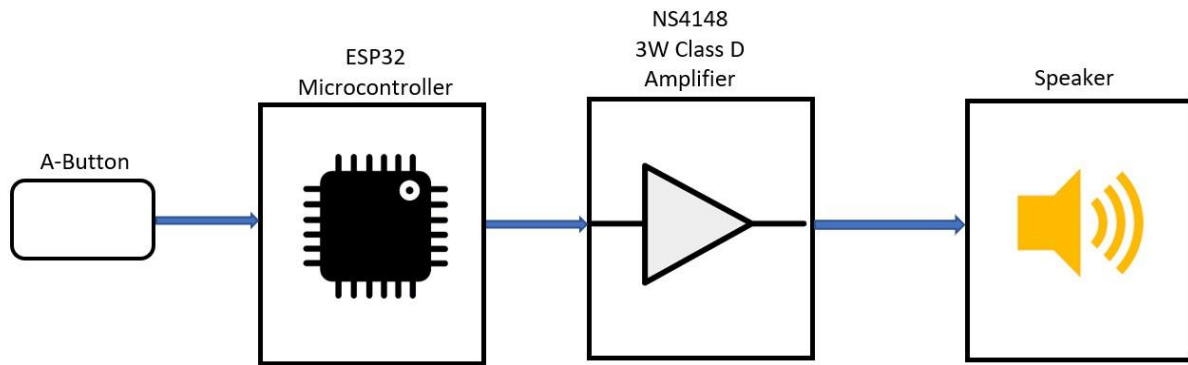


$$x_c = \frac{1}{2\pi f c}$$

$$x_c = \frac{1}{2\pi f c}$$

$$x_c = \frac{1}{2\pi_{1800100 \times 10^{-9}}}$$

$$x_c = 885\Omega$$





label0

name:

RC x:

y:

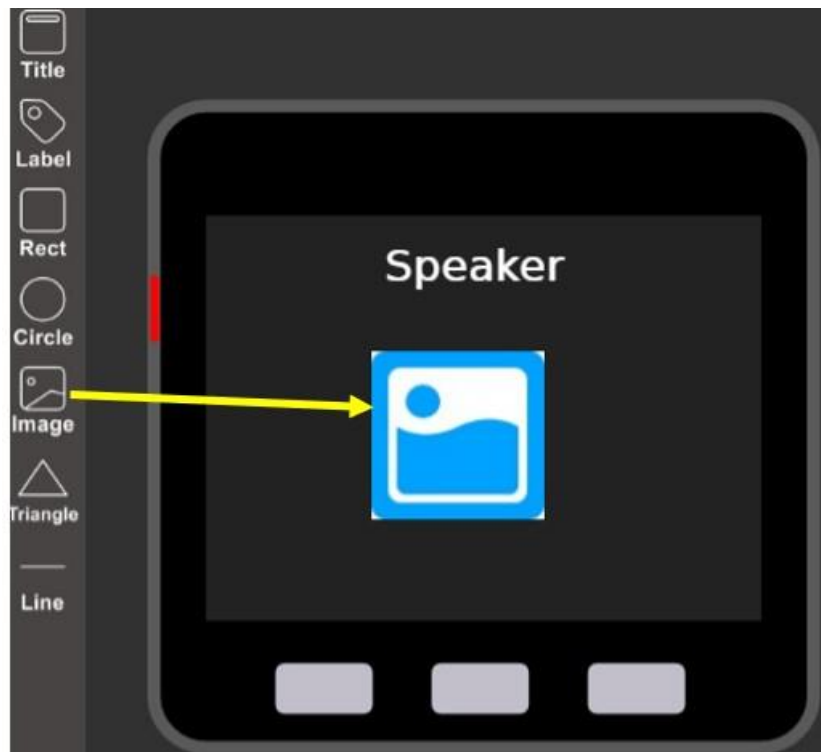
color:

text:

font:

rotation:

layer:





label0

name: label0

x: 103

y: 17

color:

text: Speaker

font: DejaVuSans 24

rotation: 0

layer: 4

image0

name: image0

x: 95

y: 80

imagePath: Reload

visibility: True

layer: 19



image0

name: image0

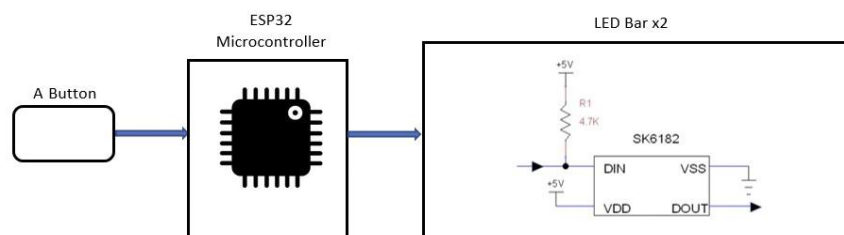
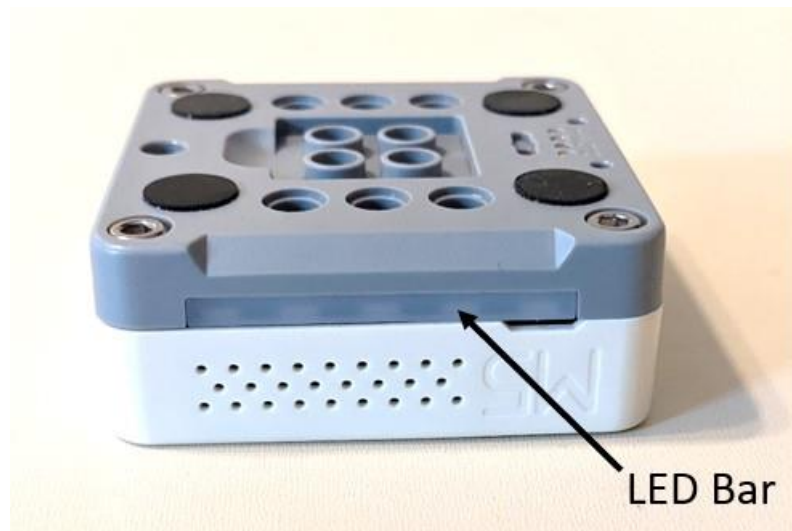
x: 95

y: 80

imagePath: speaker.png Reload

visibility: True

layer: 19



search

Event

► UI

▼ Hardwares

Speaker

RGB

IMU




Power

► Units

► Modules

► FACES

RGB

- Set RGB Bar color 
- Set RGB Bar color R G B
- Set side RGB Bar color 
- Set side RGB Bar color by R G B
- Set the RGB color 
- Set the RGB color by R G B
- Set RGB brightness

Button A wasPressed

Set RGB Bar color R G B

Button A wasPressed

Set RGB Bar color R G B

255



Button A wasPressed

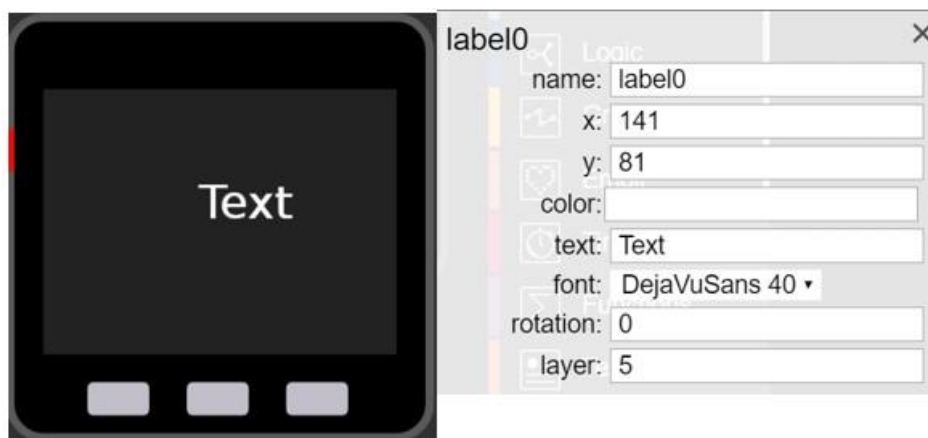
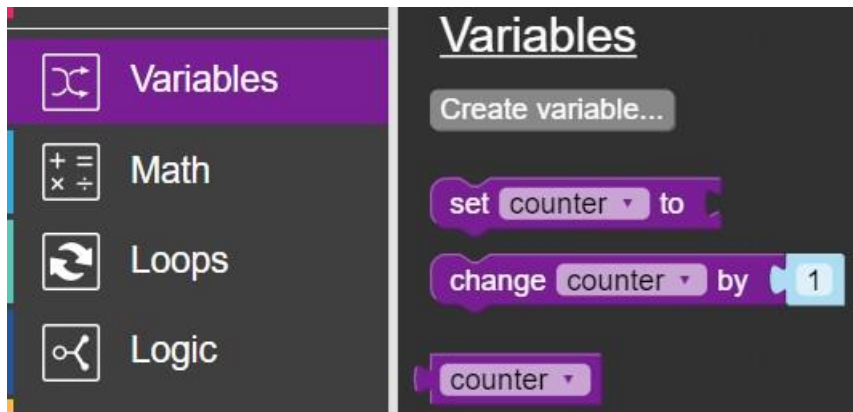
Set RGB Bar color R G B

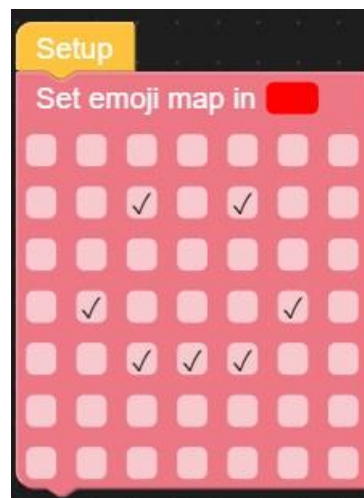
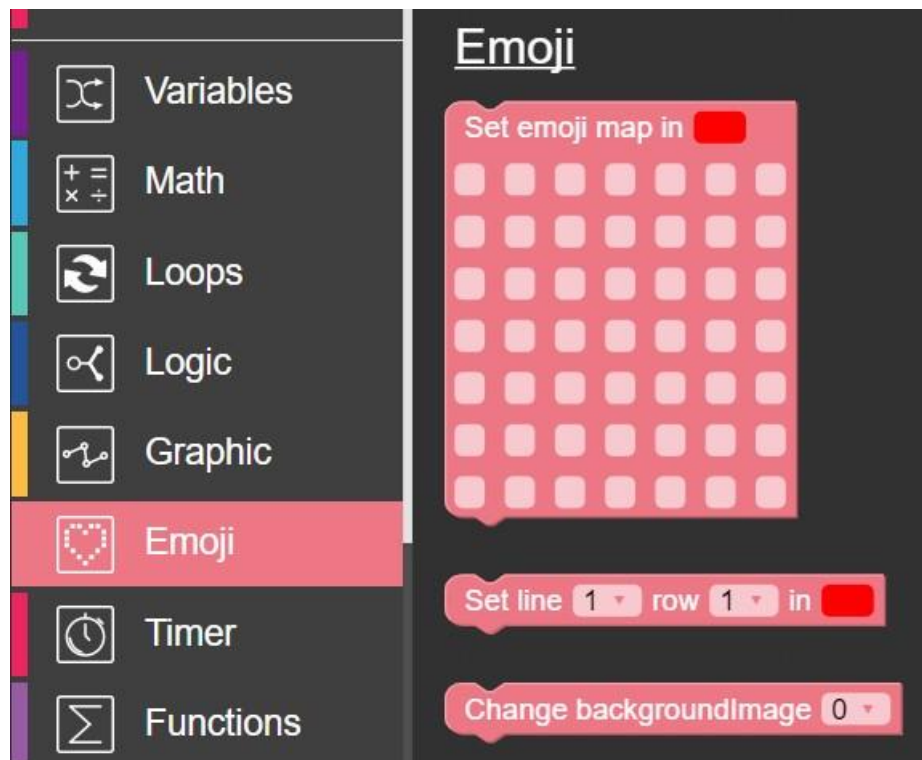


```
Button A wasPressed
  Set RGB Bar color R 255 G 255 B 255

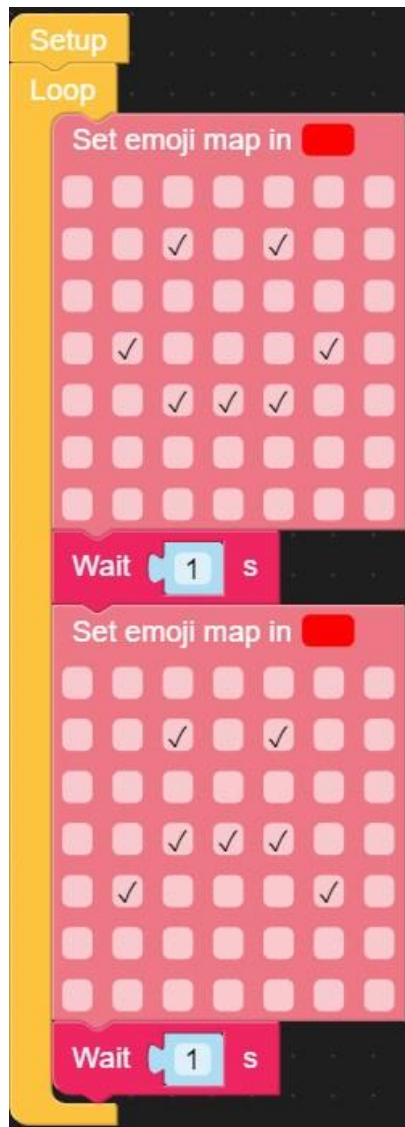
Button B wasPressed
  Set RGB Bar color R 0 G 0 B 0
```

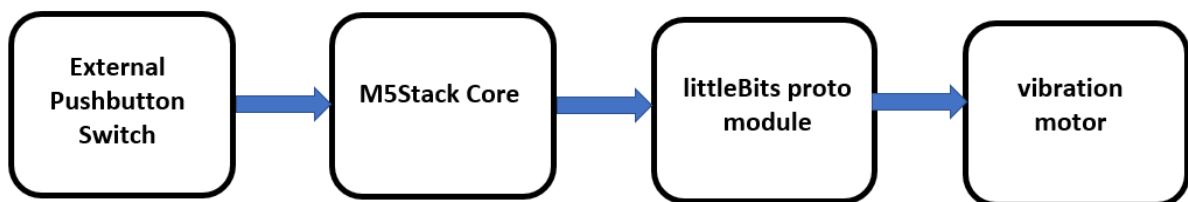
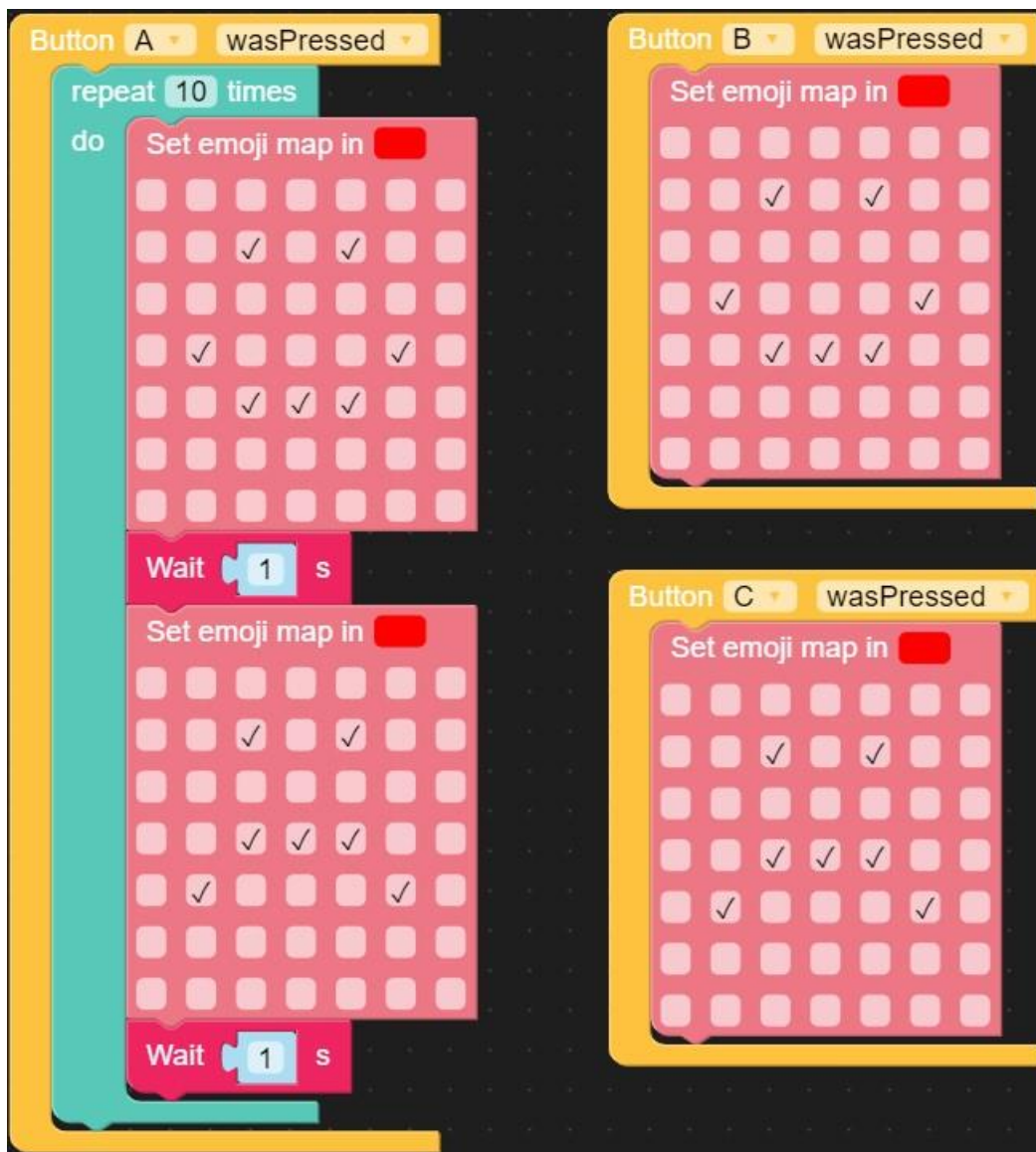
```
Button A wasPressed
  set counter to 0
  repeat 10 times
    do
      Set RGB Bar color R 255 G 255 B 255
      Wait 1 s
      Set RGB Bar color R 0 G 0 B 0
      Wait 1 s
      change counter by 1
      Label label0 show counter
```

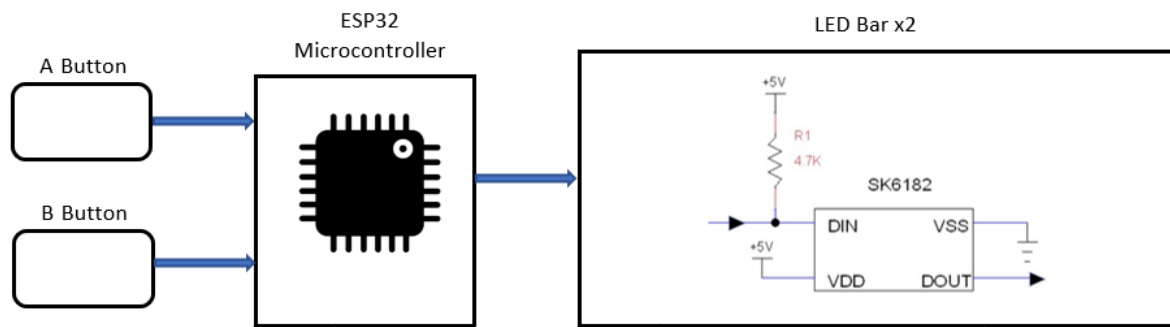






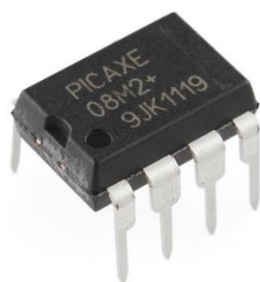






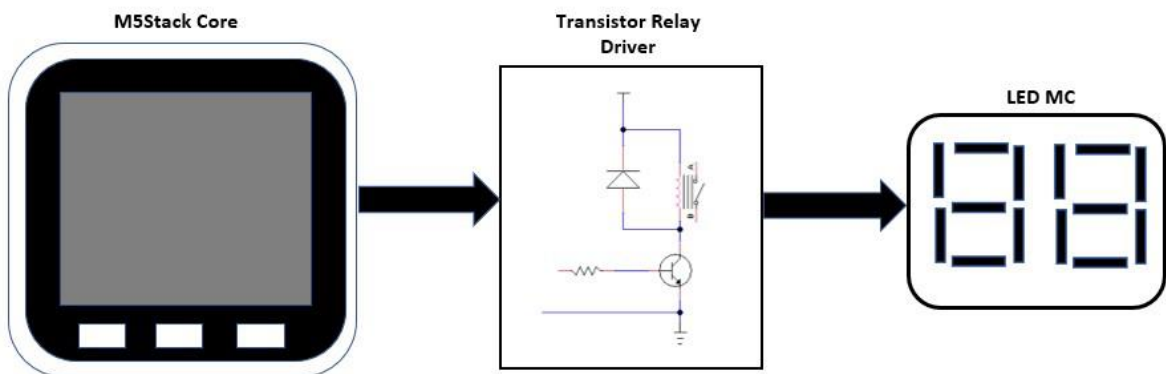
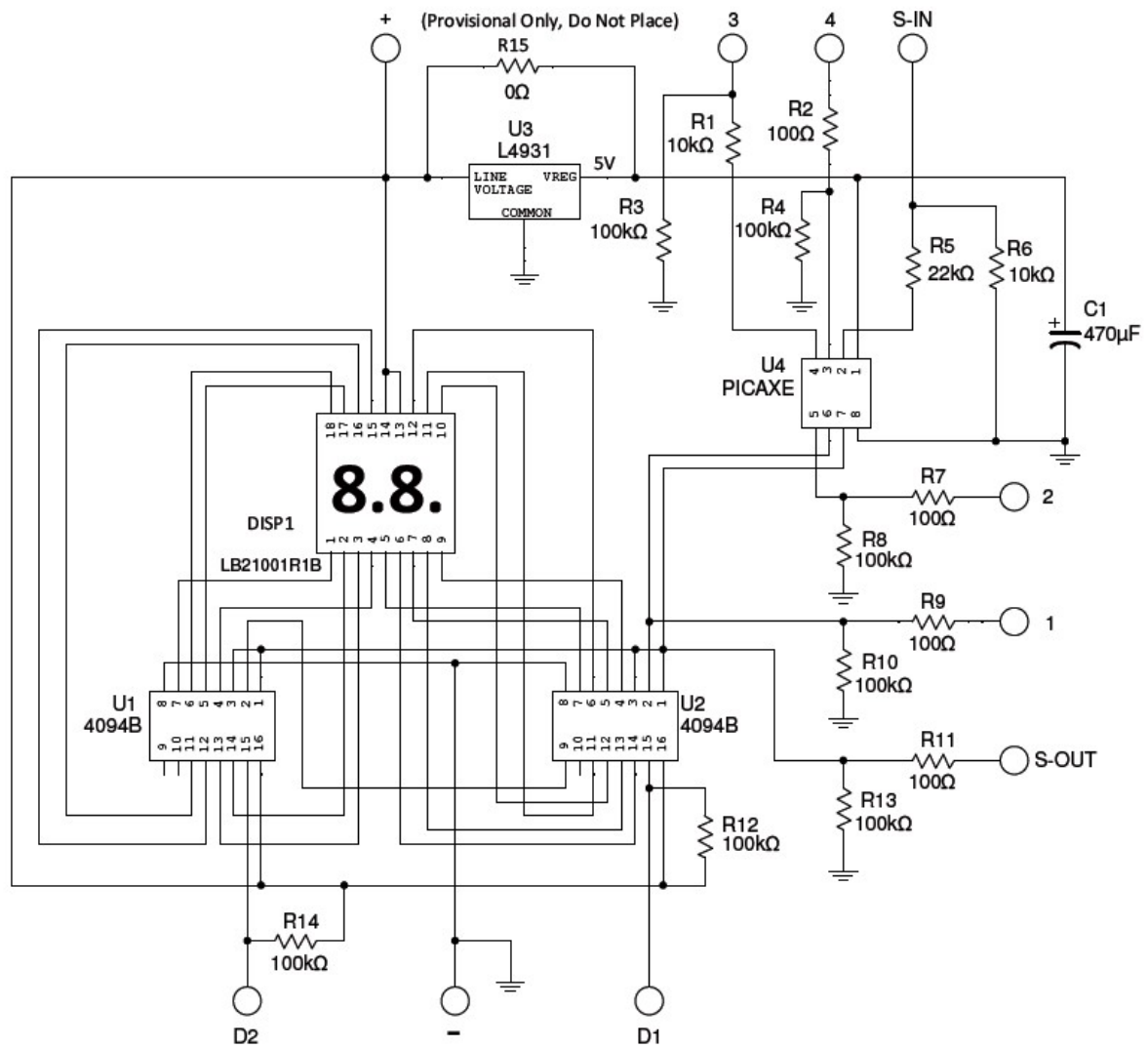
Chapter 4: It's a SNAP! Snap Circuits and the M5Stack Core

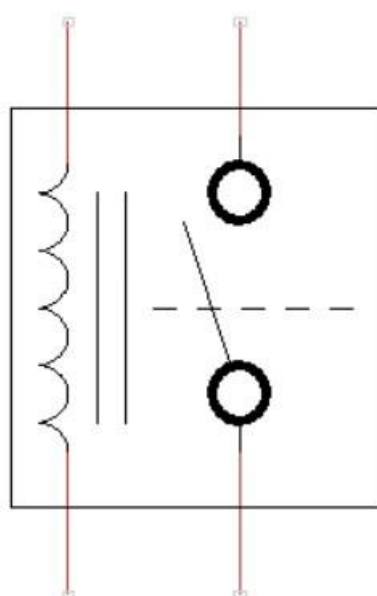


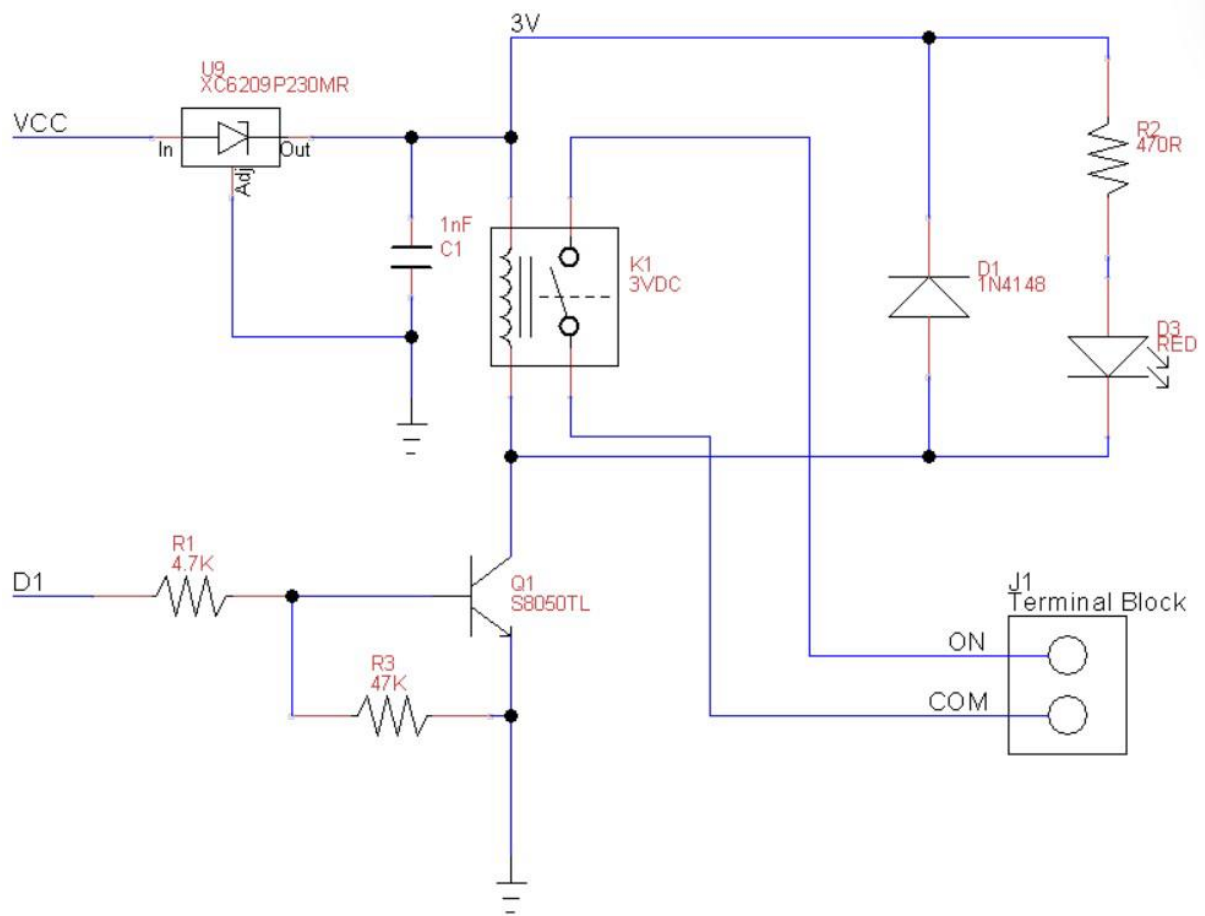


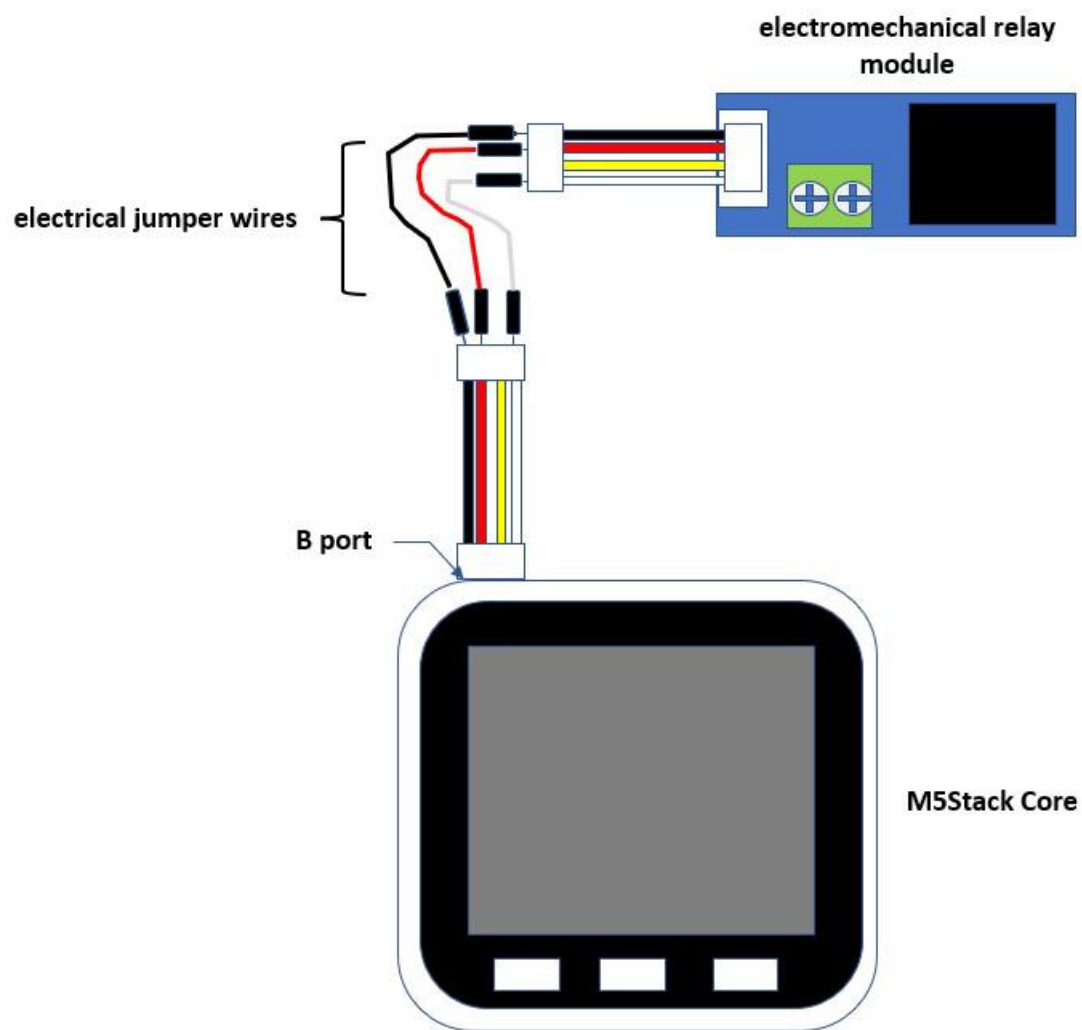


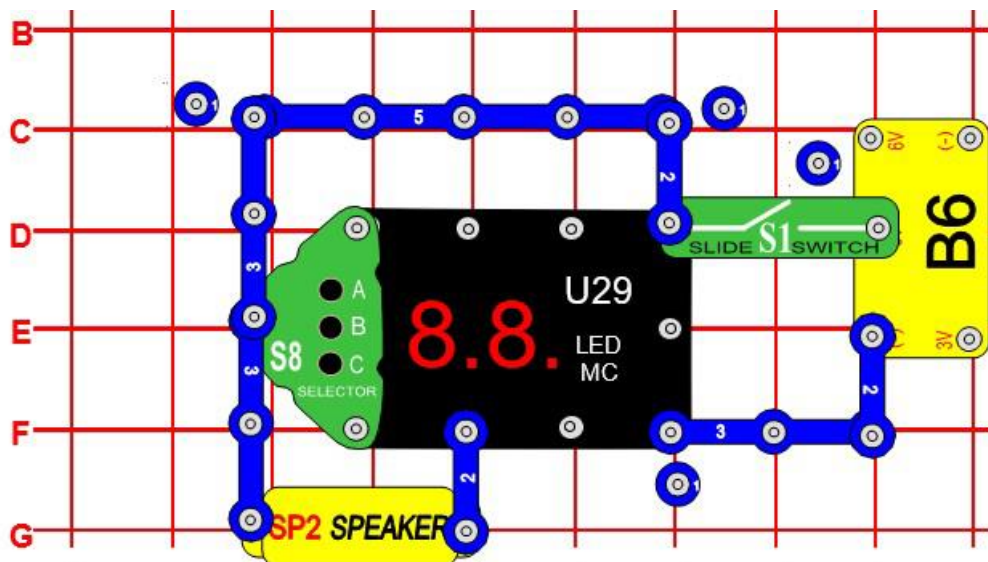
- (+): Battery
- (-): Battery ground
- (S-IN): Selector (S8) input
- (S-OUT): An output, typically connected to an LED
- (1): An output, typically connected to an LED
- (2): An output, typically connected to a speaker
- (3): An input, Selector (S8) input
- (4): An output, typically connected to an LED
- (D1): Turns off the right LED display
- (D2): Turns off the left LED display

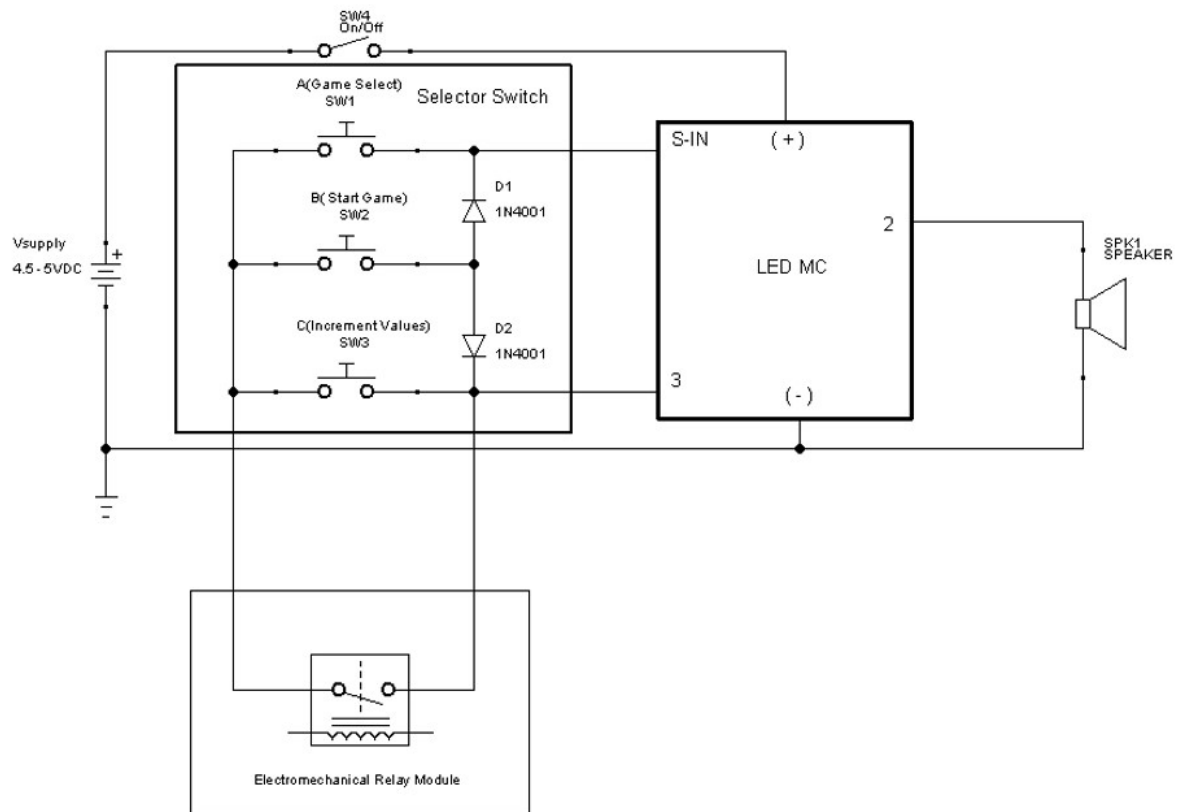
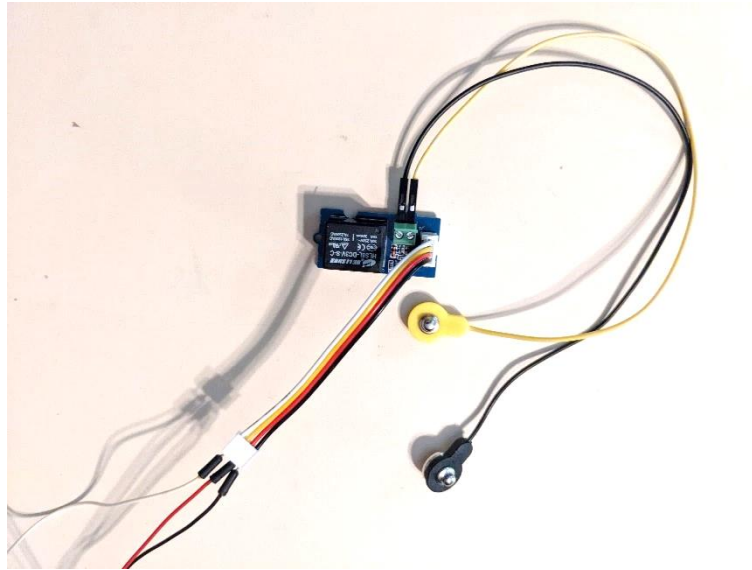


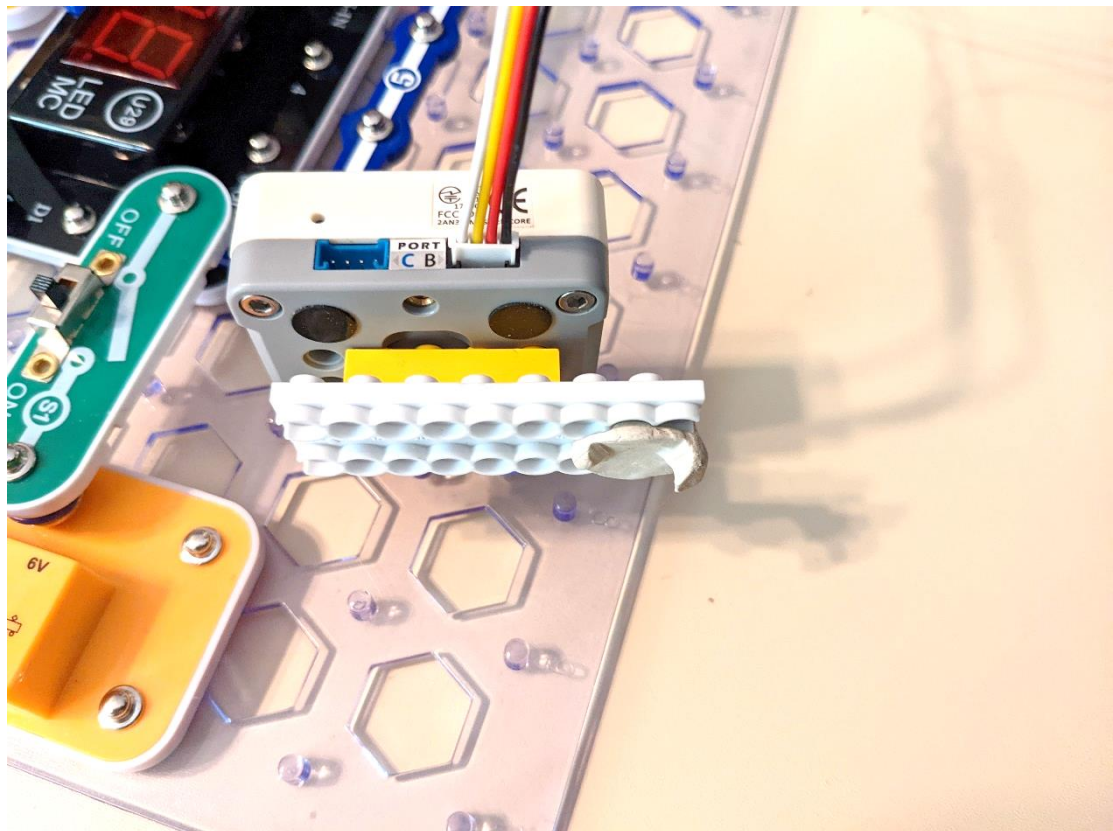
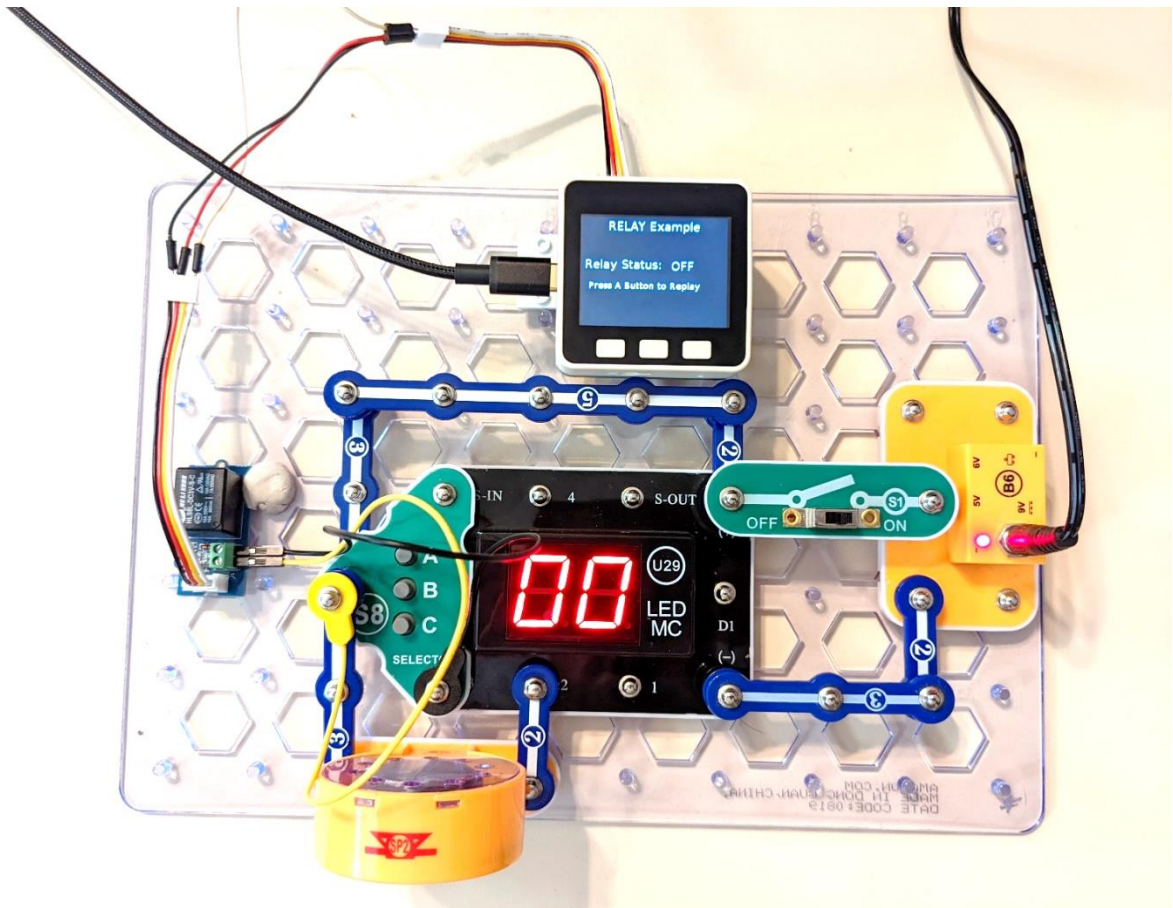


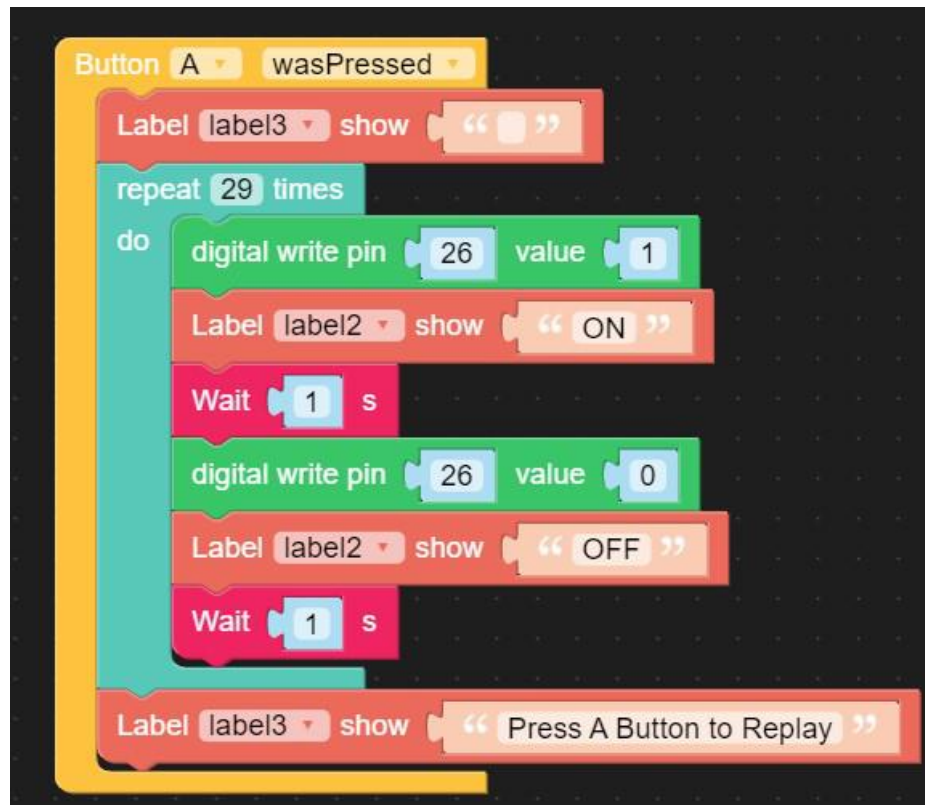












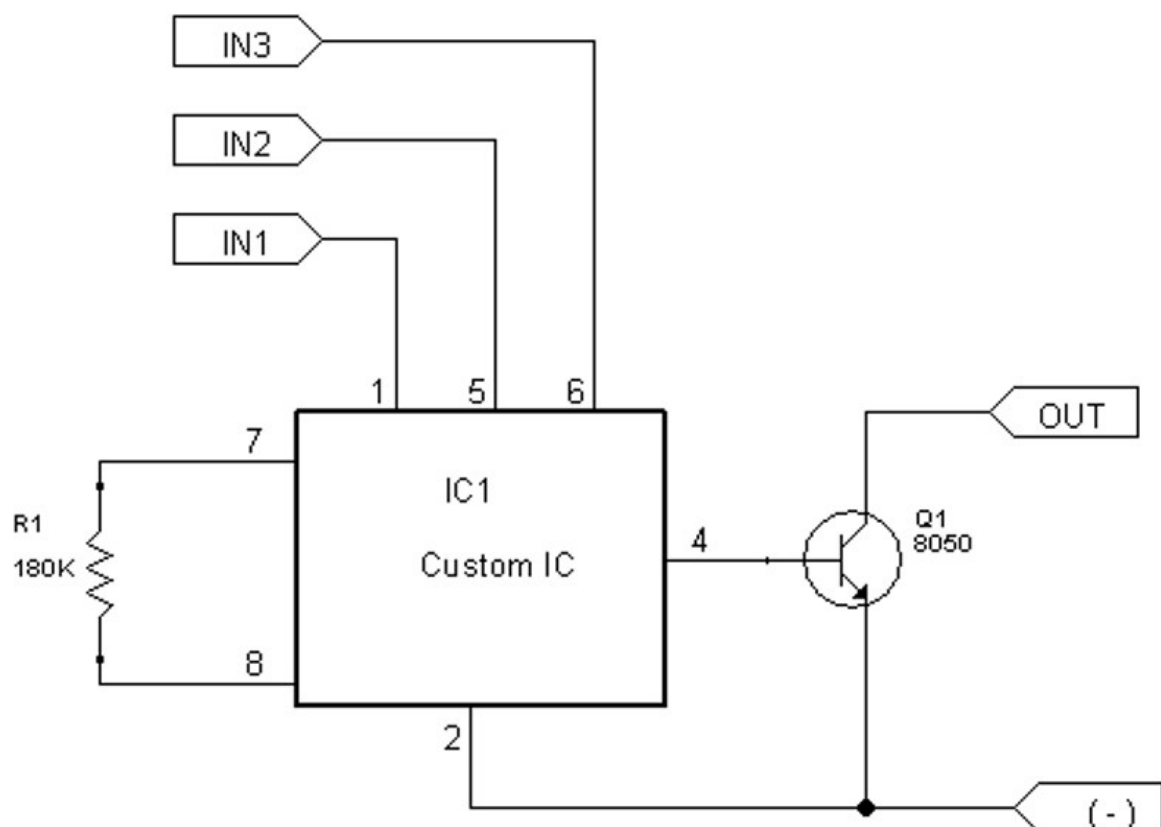


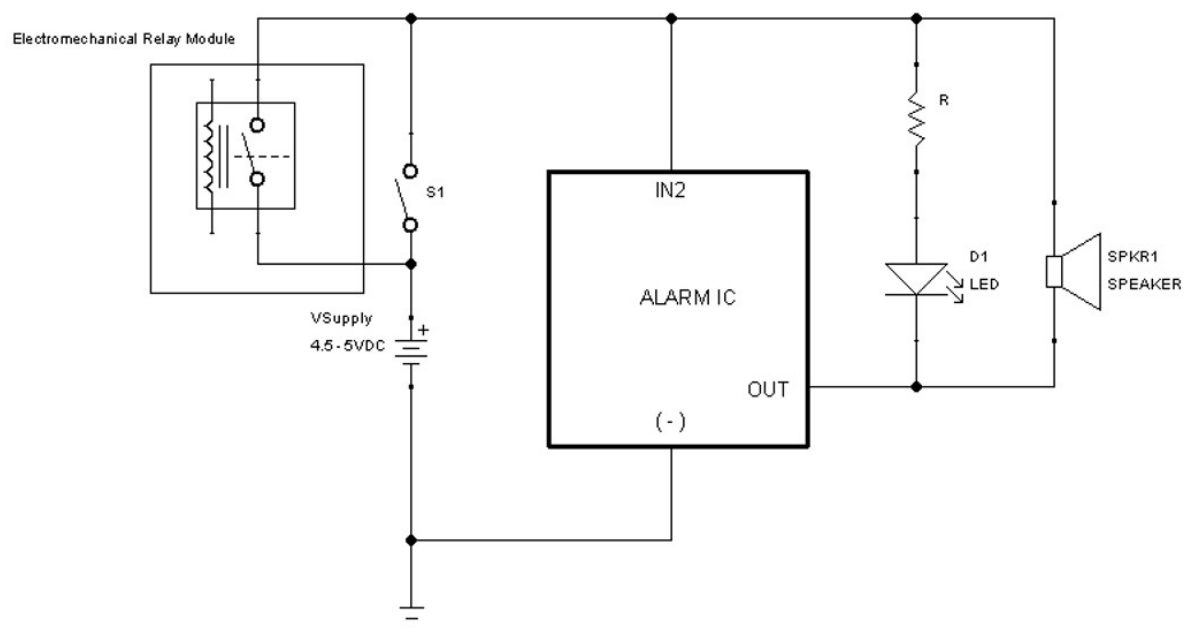
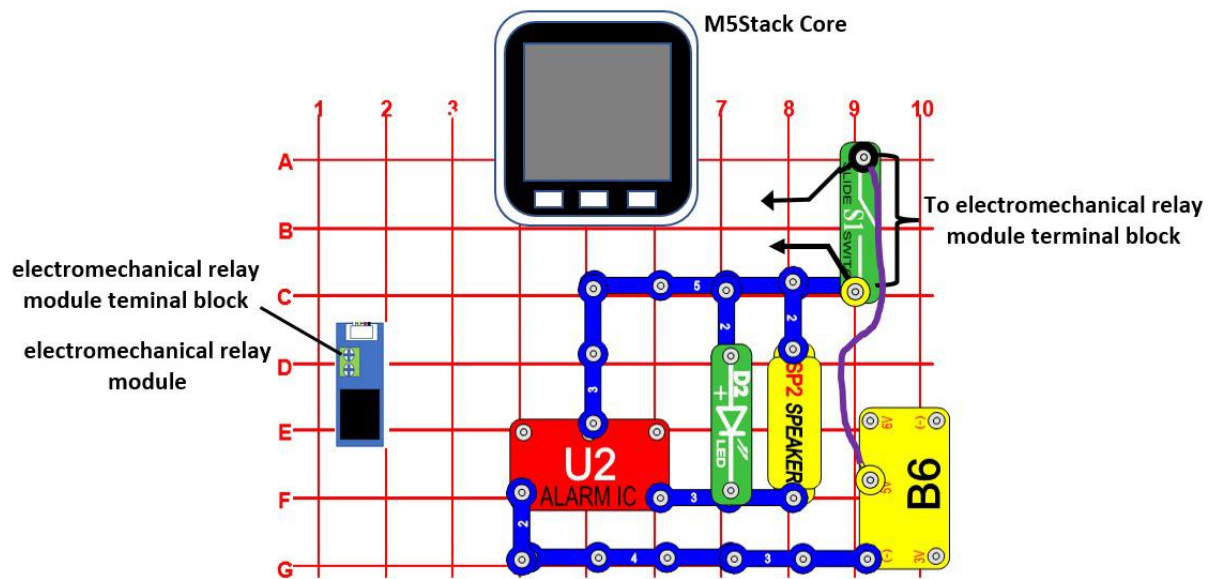
Connections

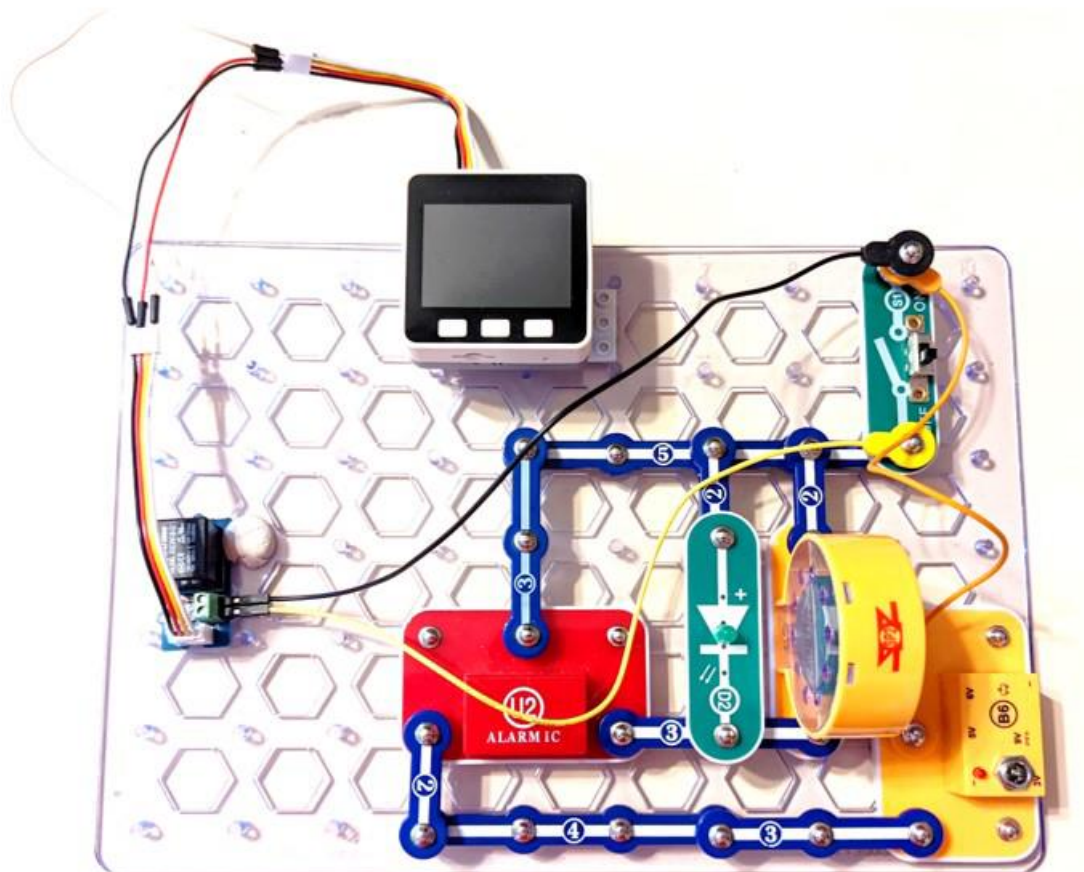
IN1, IN2, IN3 - control inputs

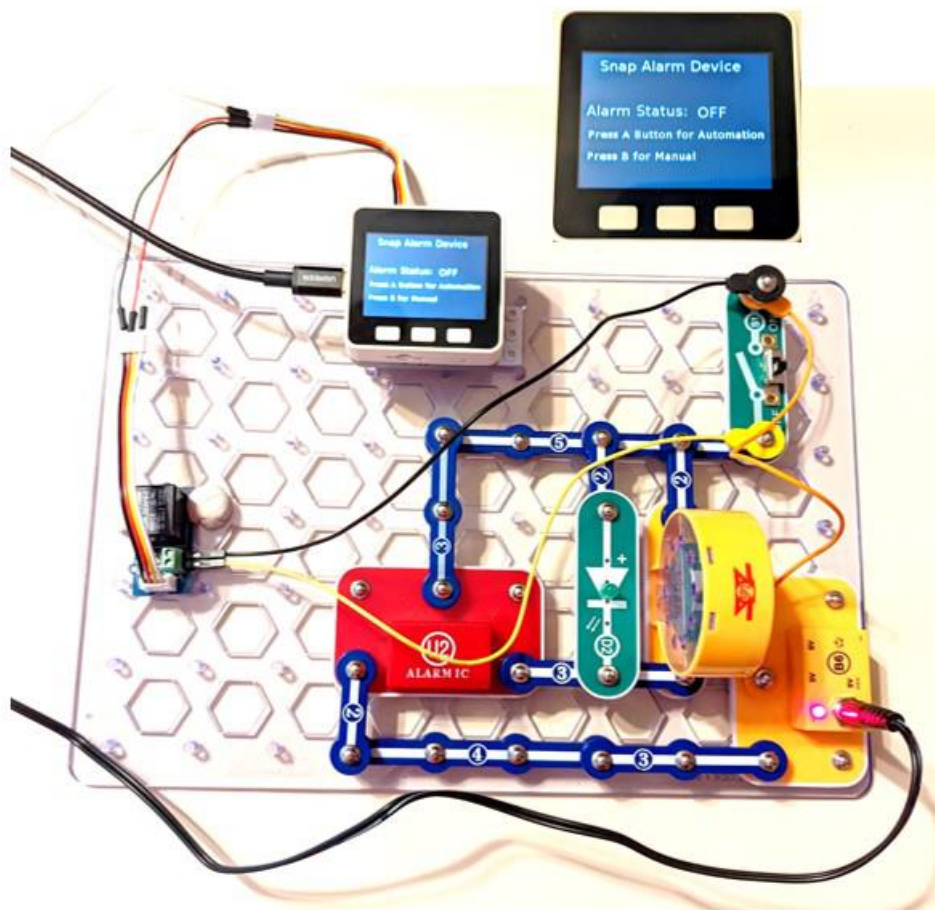
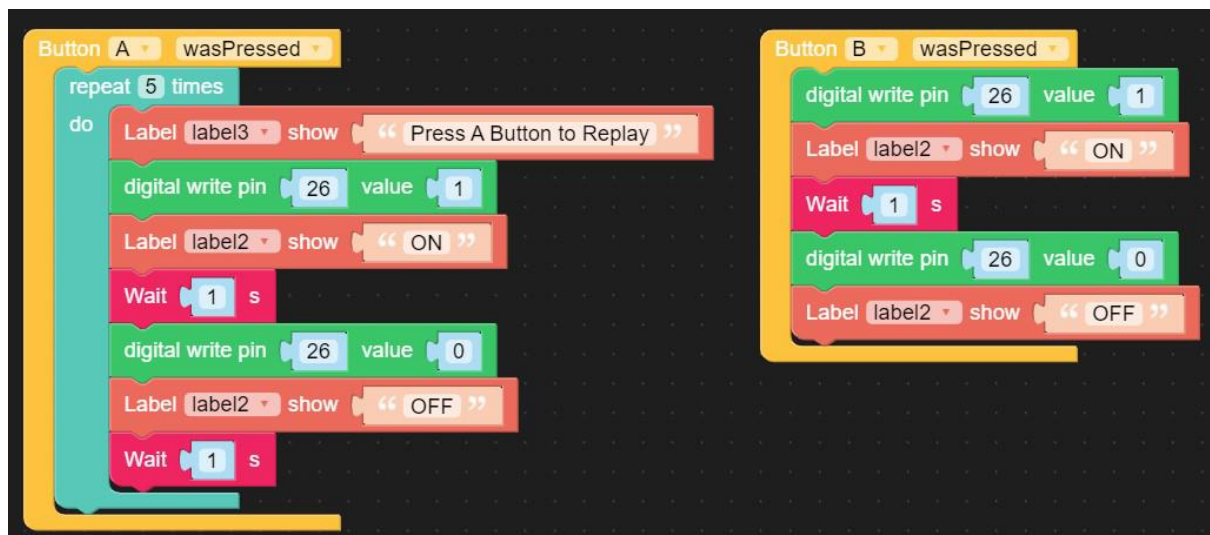
(-) - power return/ground to batteries

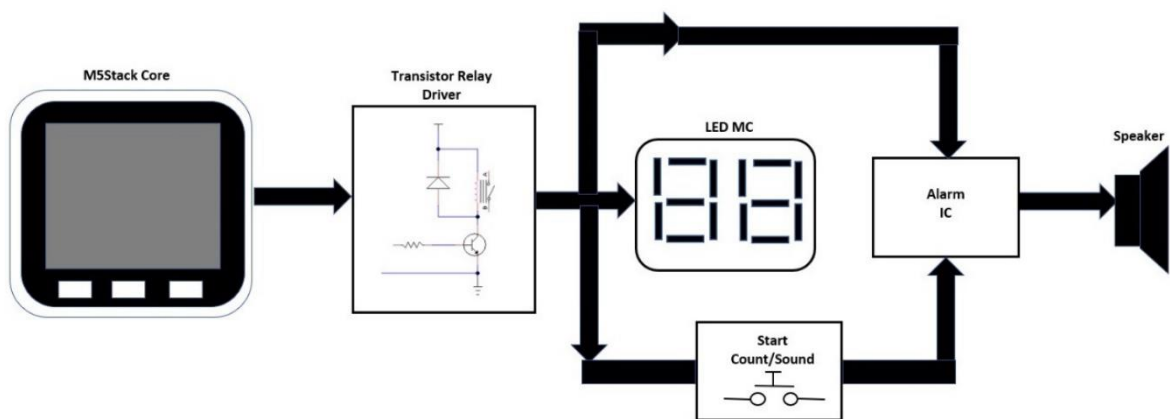
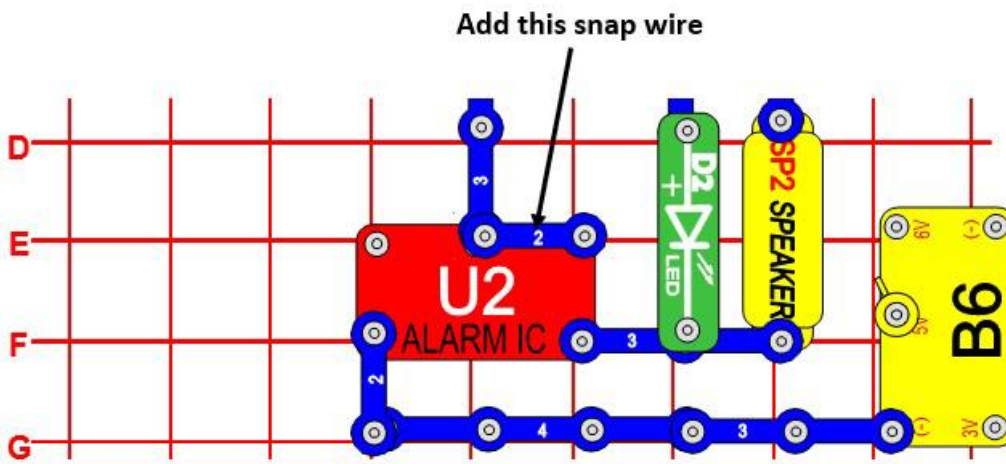
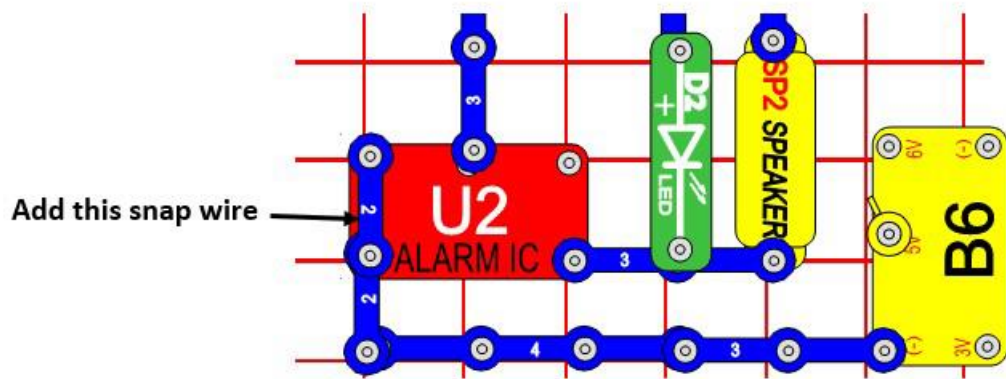
OUT - output connection

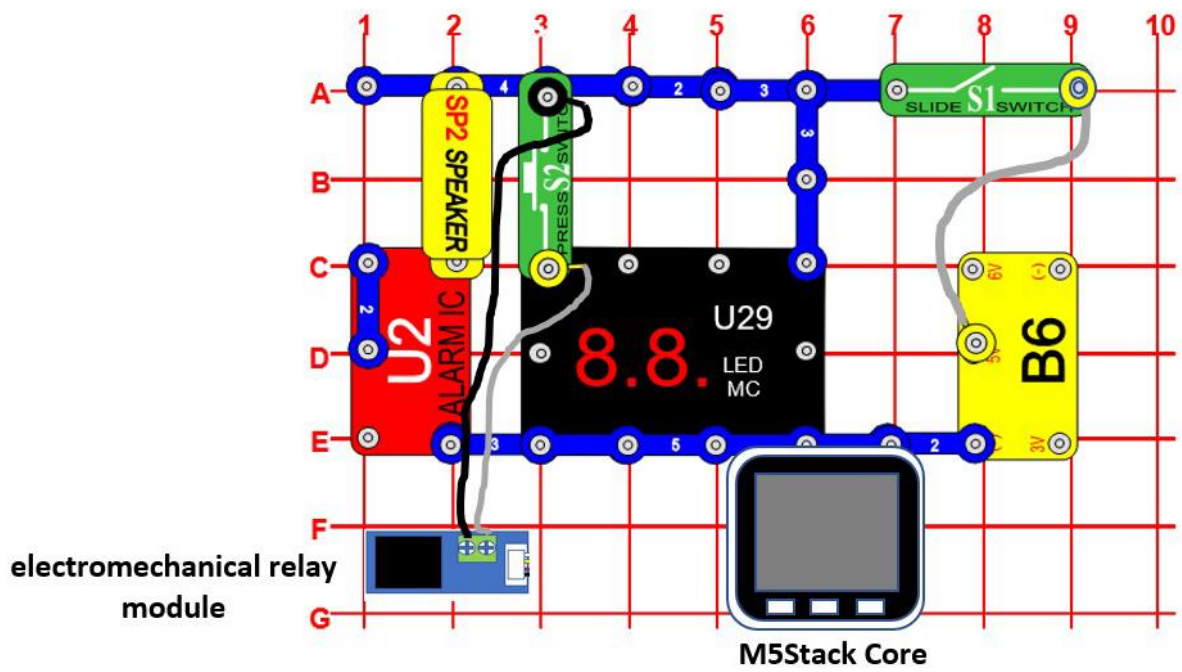
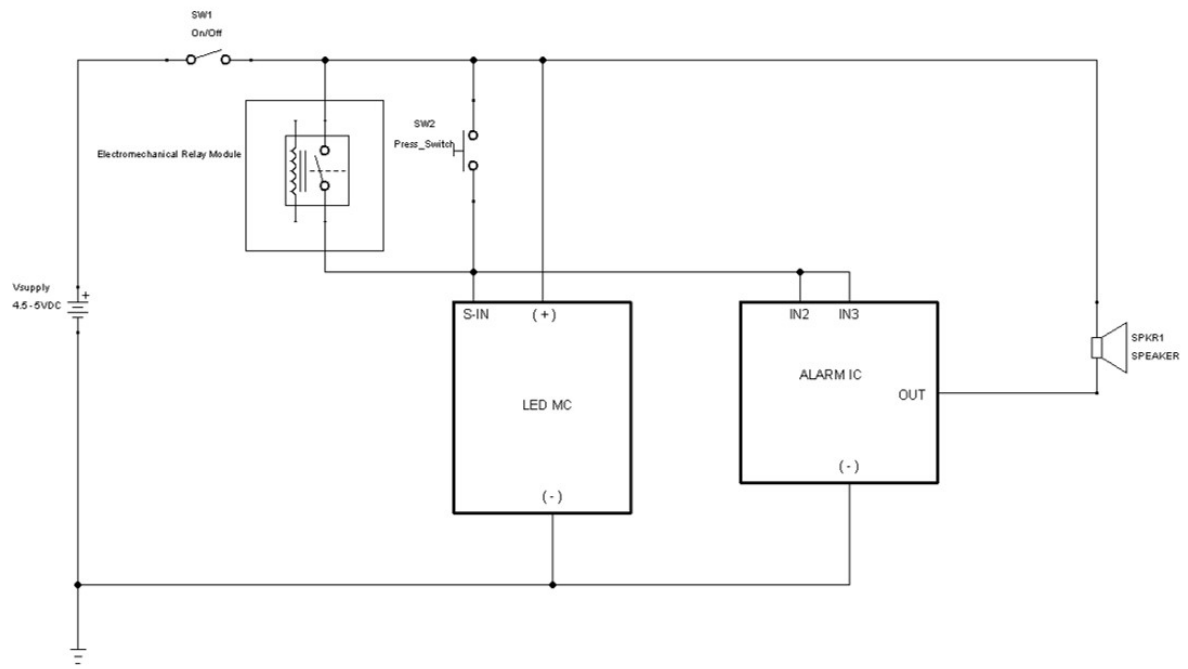


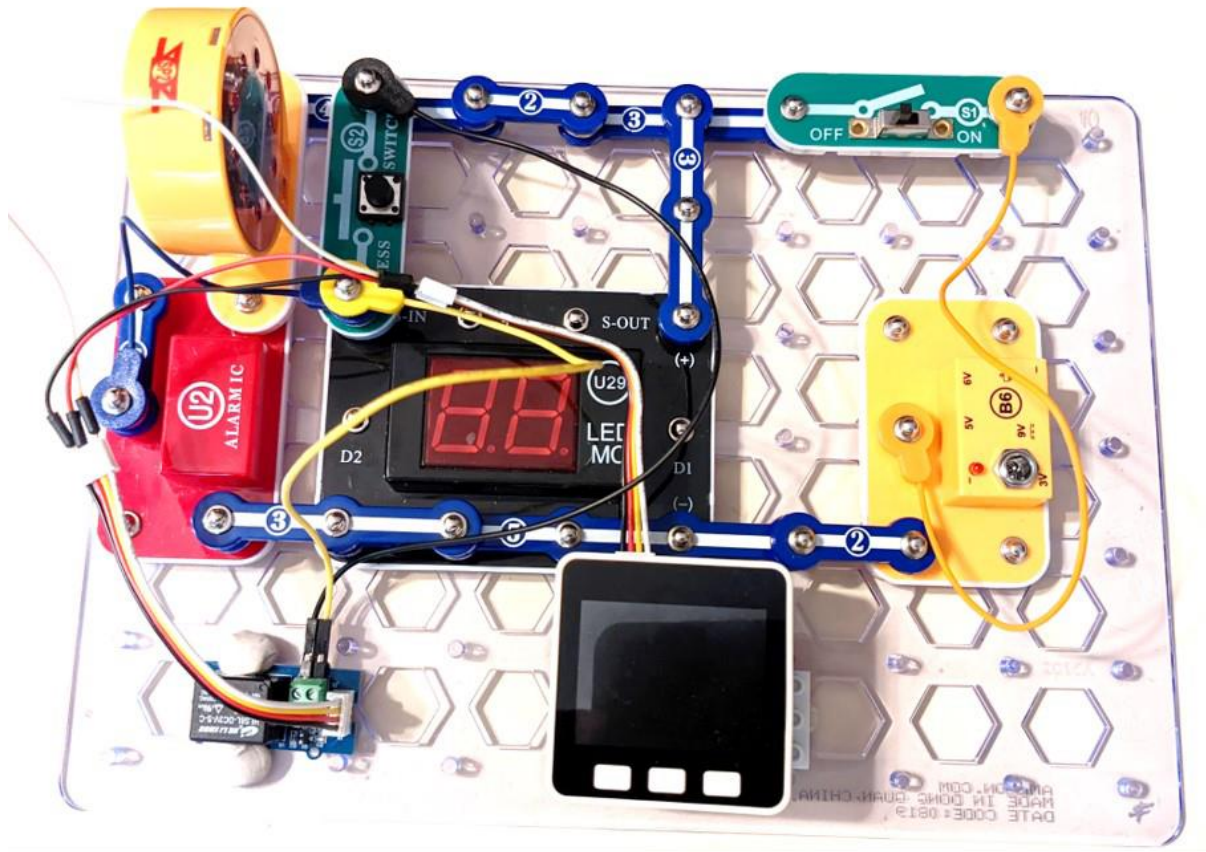






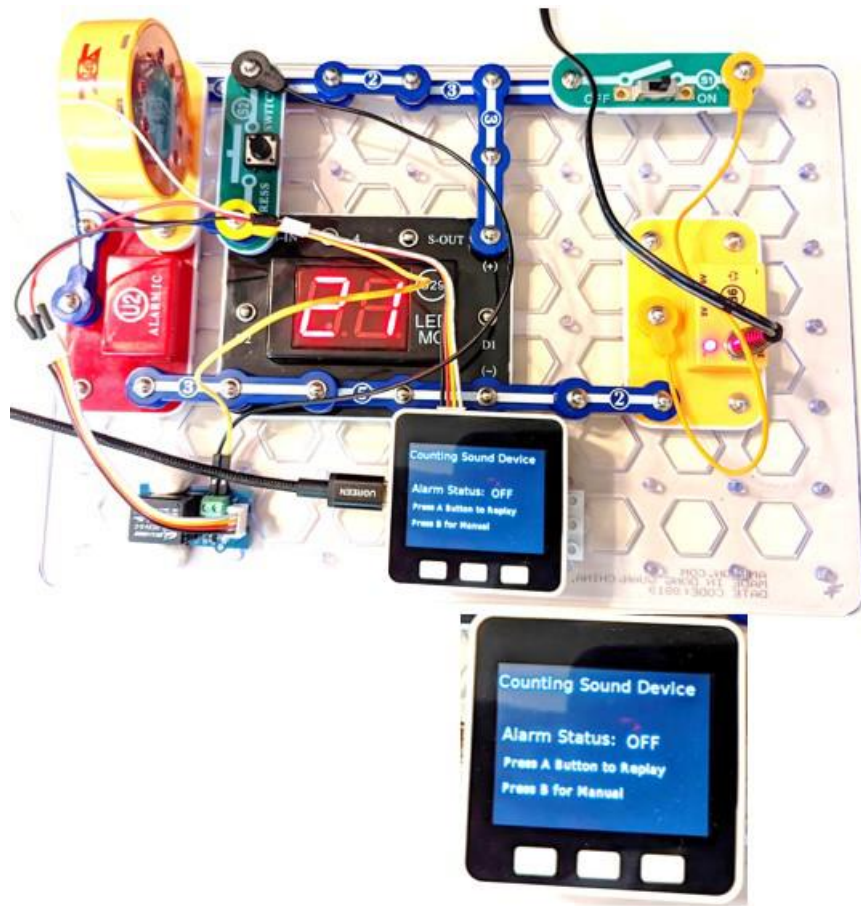
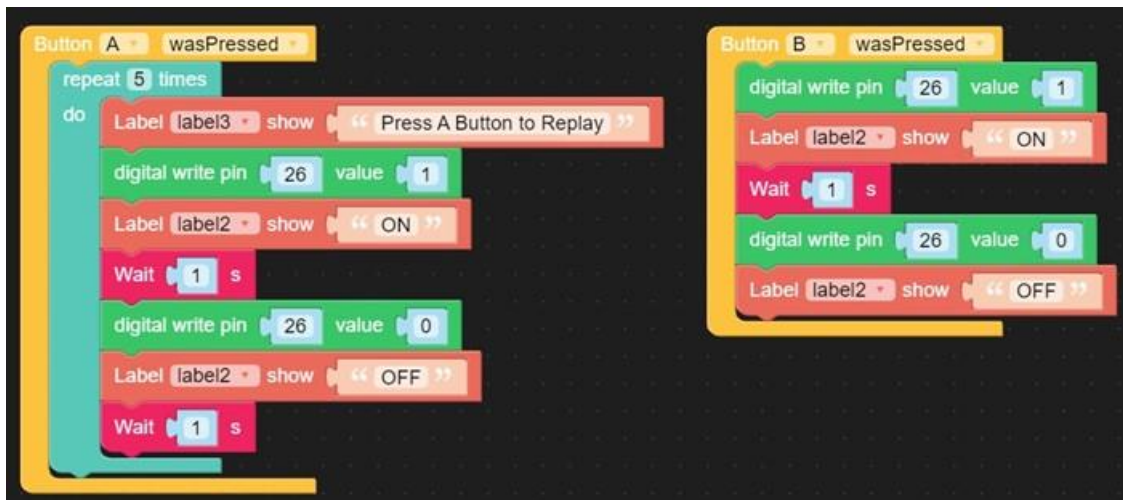




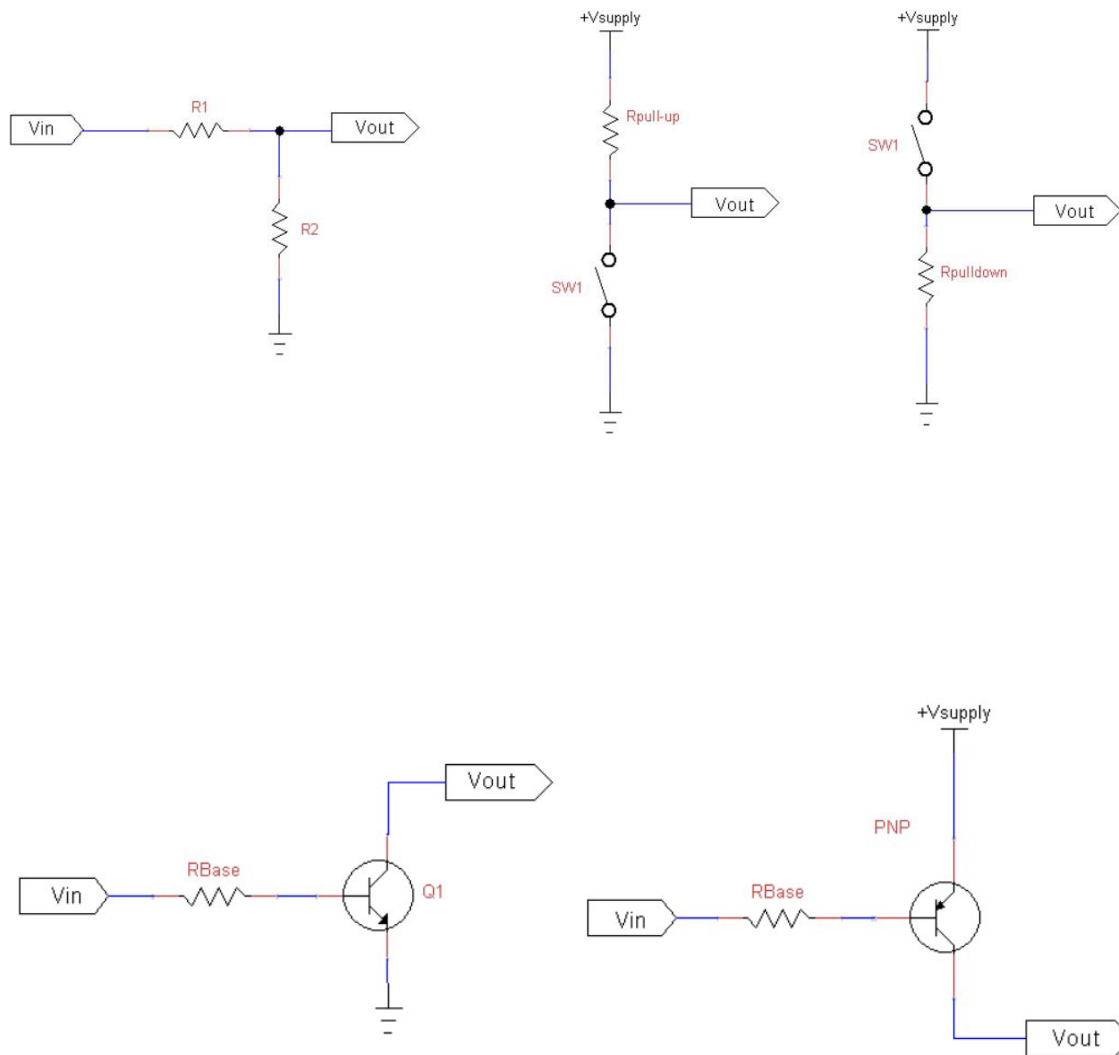


label0 IoTCloud X

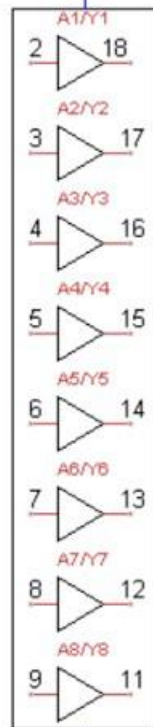
name:	label0
x:	0
y:	13
color:	
text:	Counting Sound Device
font:	DejaVuSans 24 ▼
rotation:	0
layer:	1



Chapter 5: Solderless Breadboarding with the M5Stack



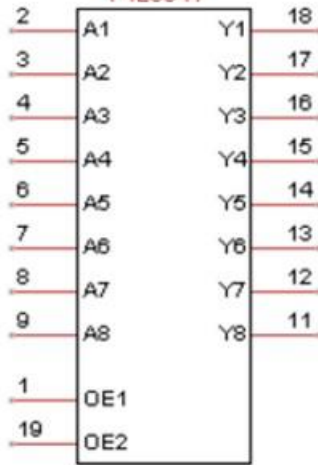
1 ○ OE1 A1/Y1 - A4/Y4

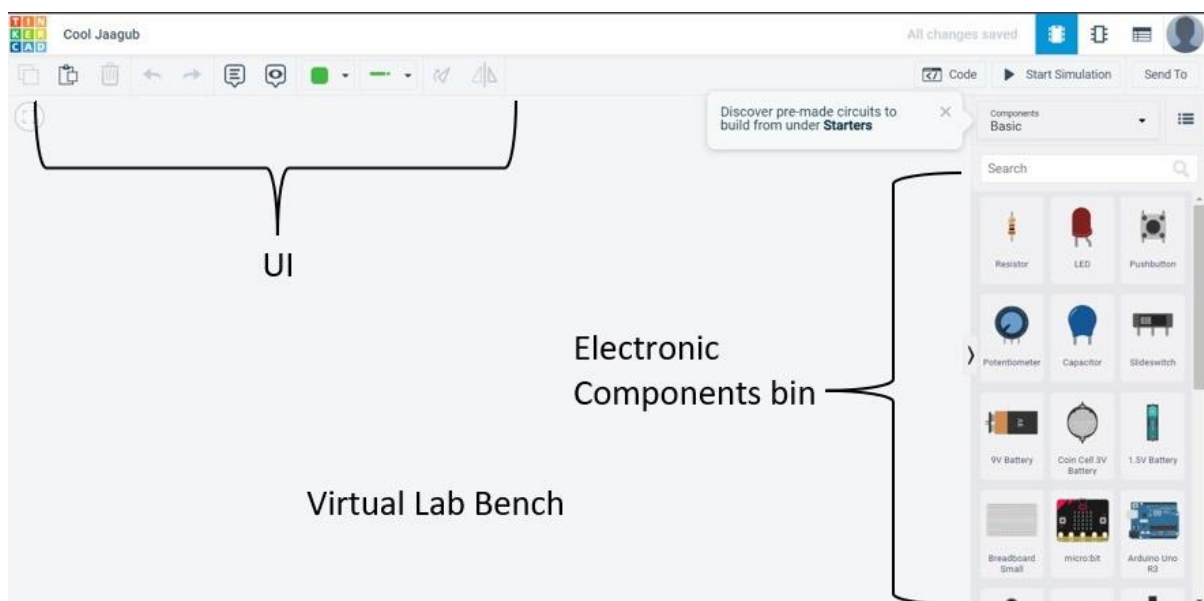
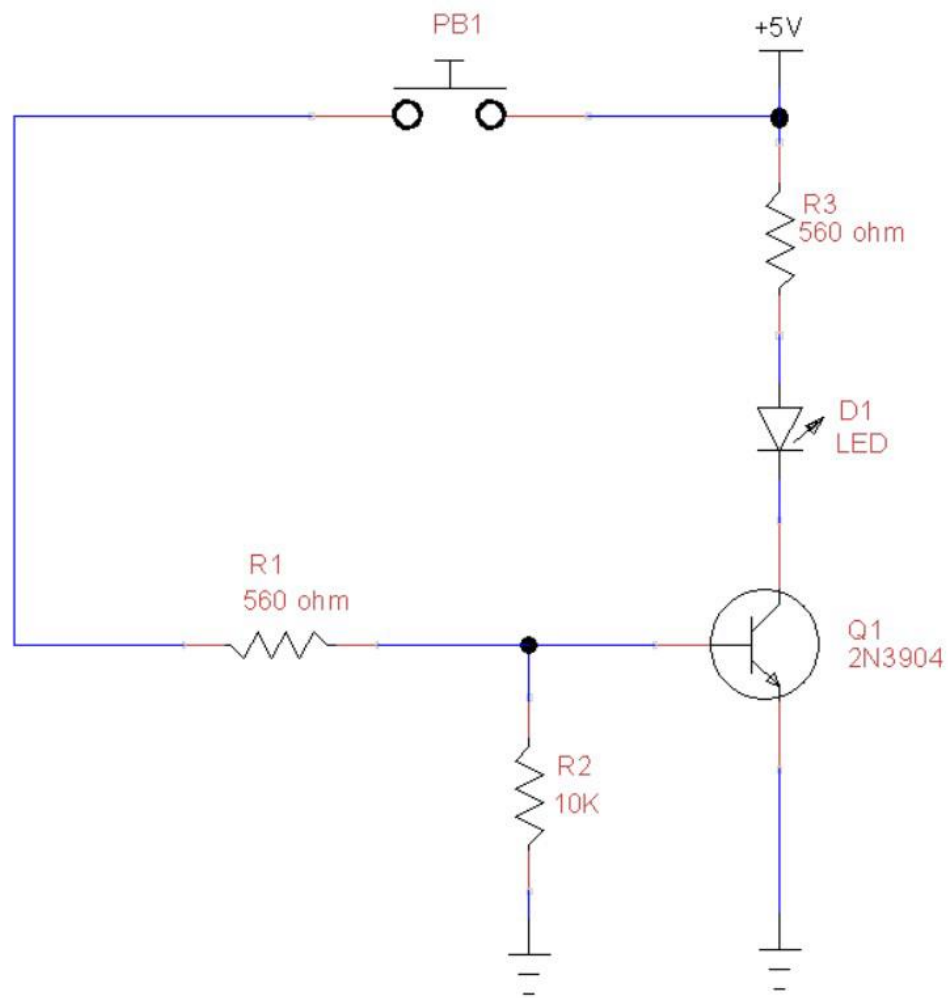


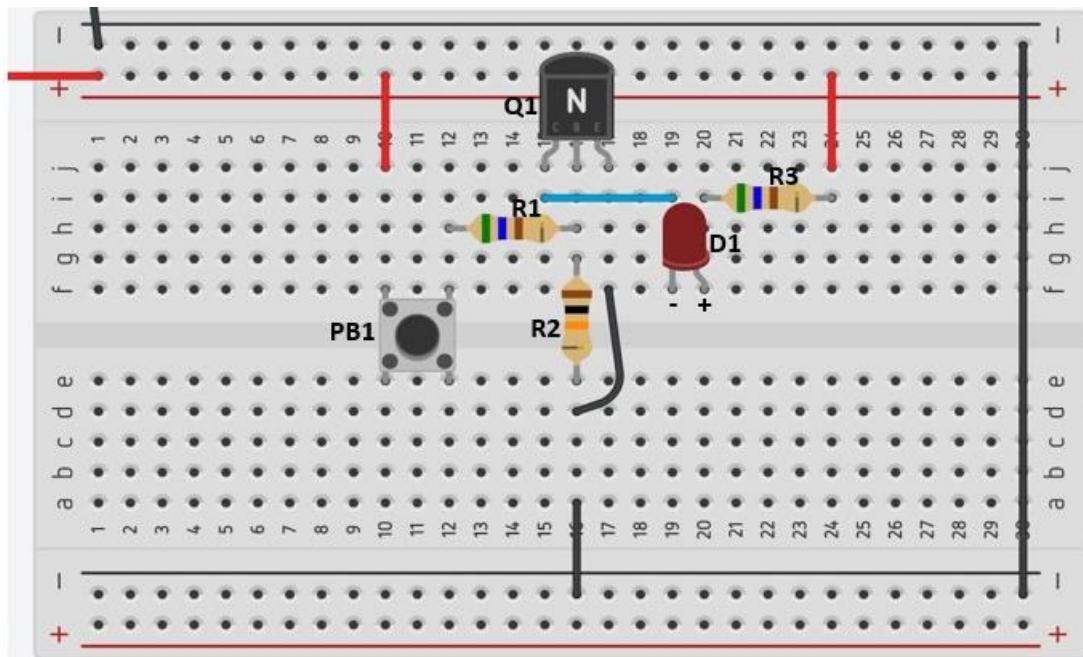
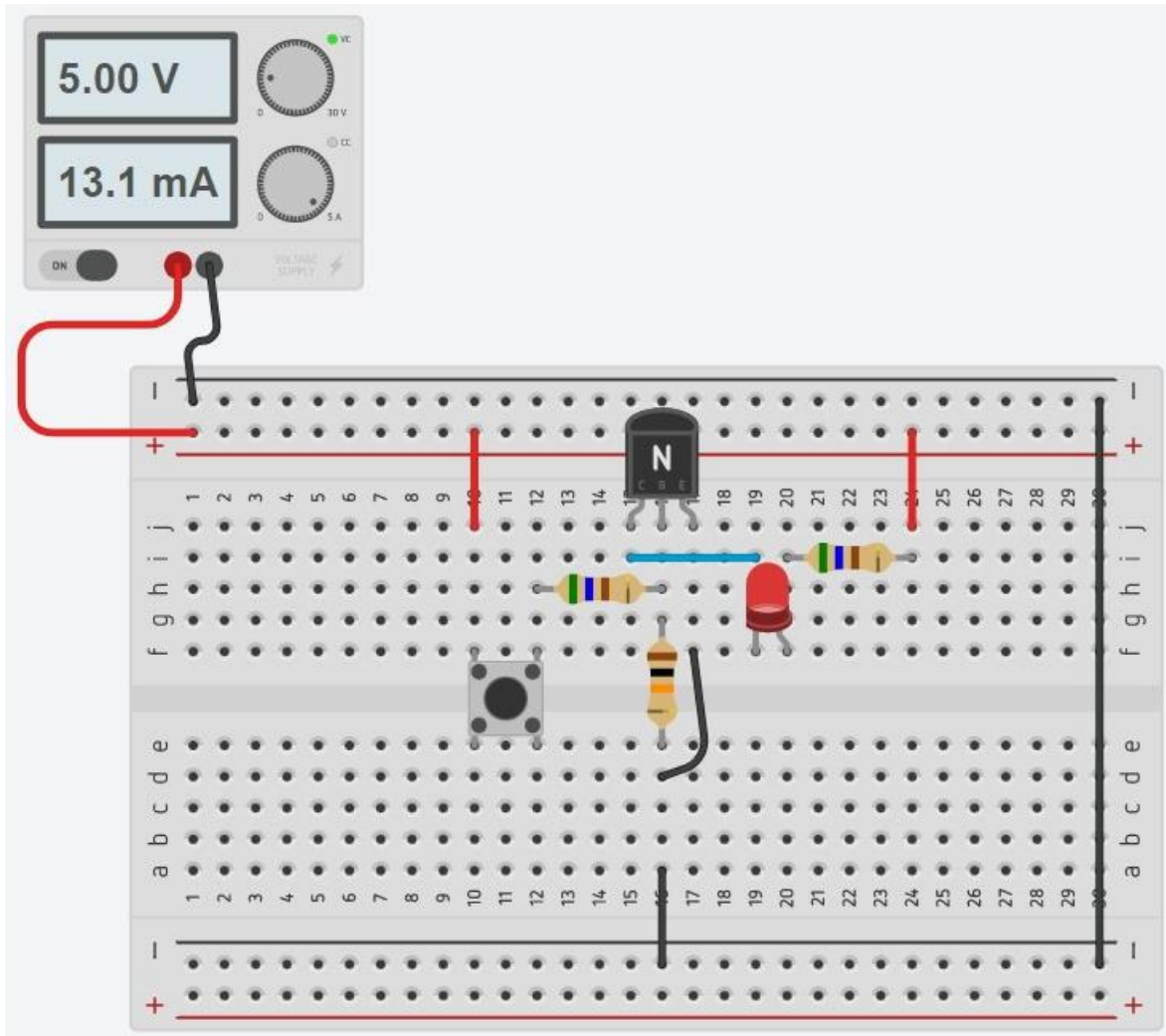
19 ○ OE2 A5/Y5 - A8/Y8

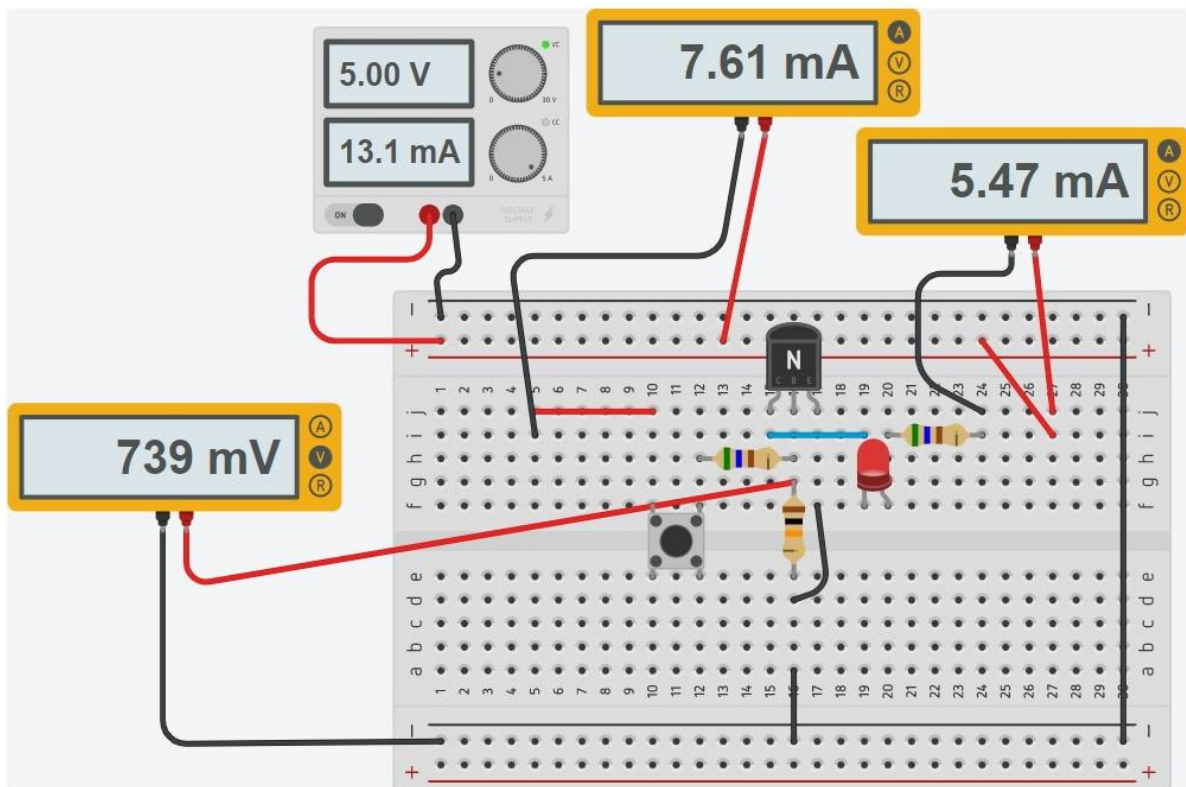
U1

74LS541

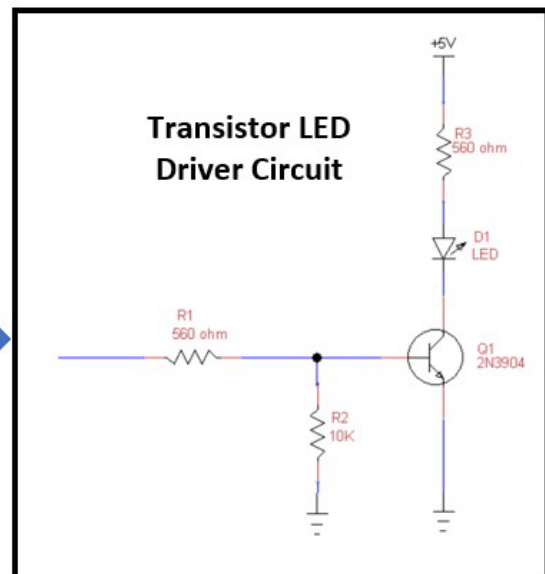
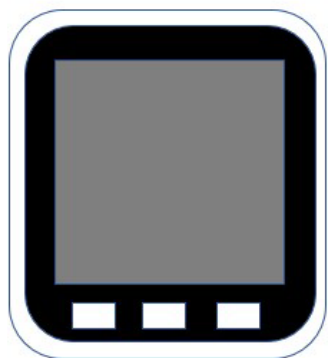


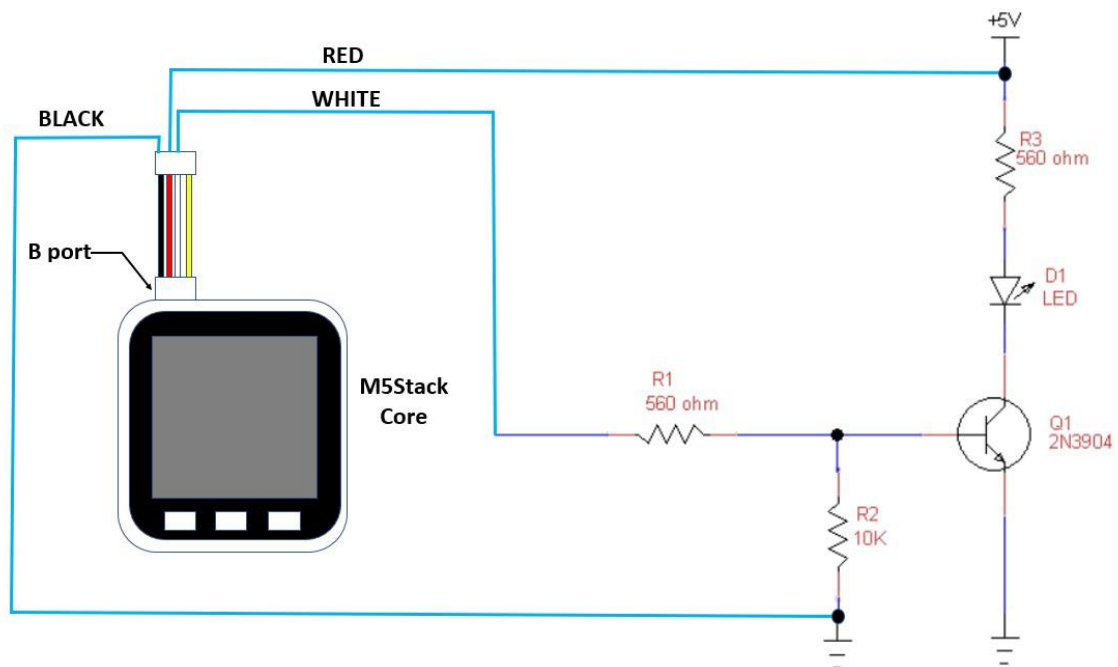
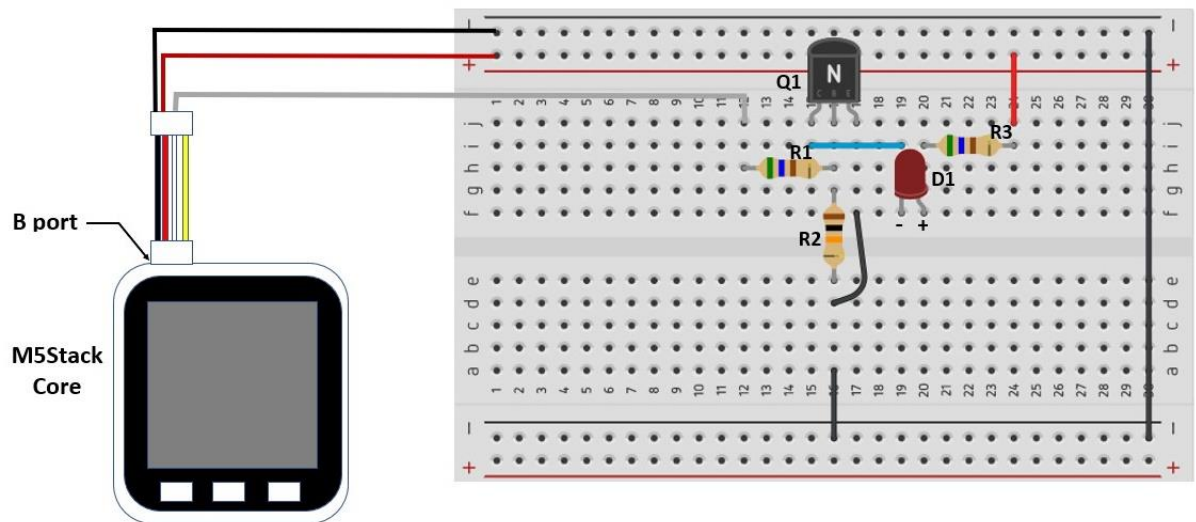


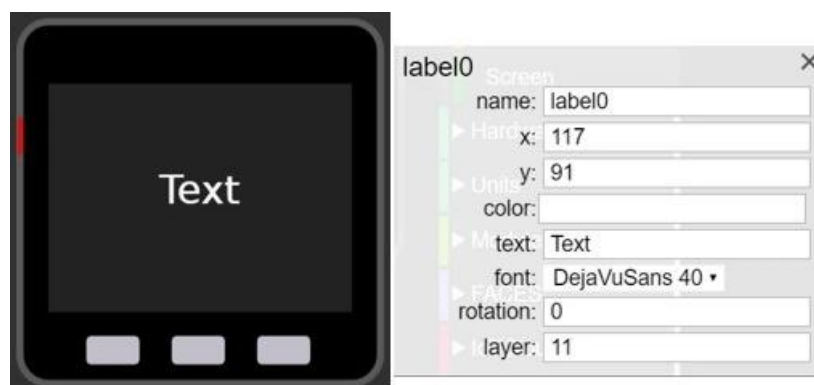
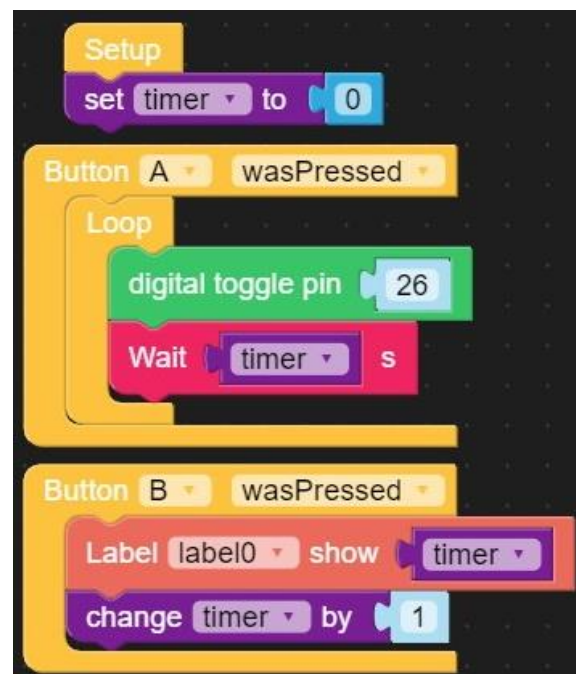
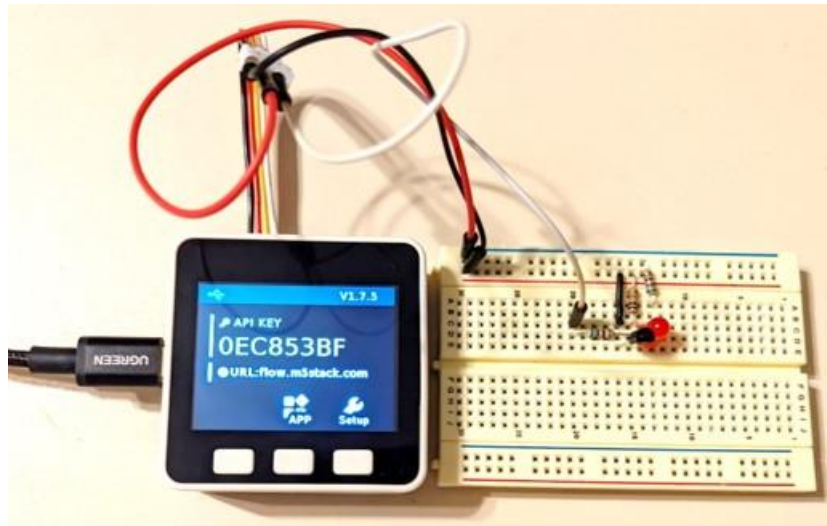


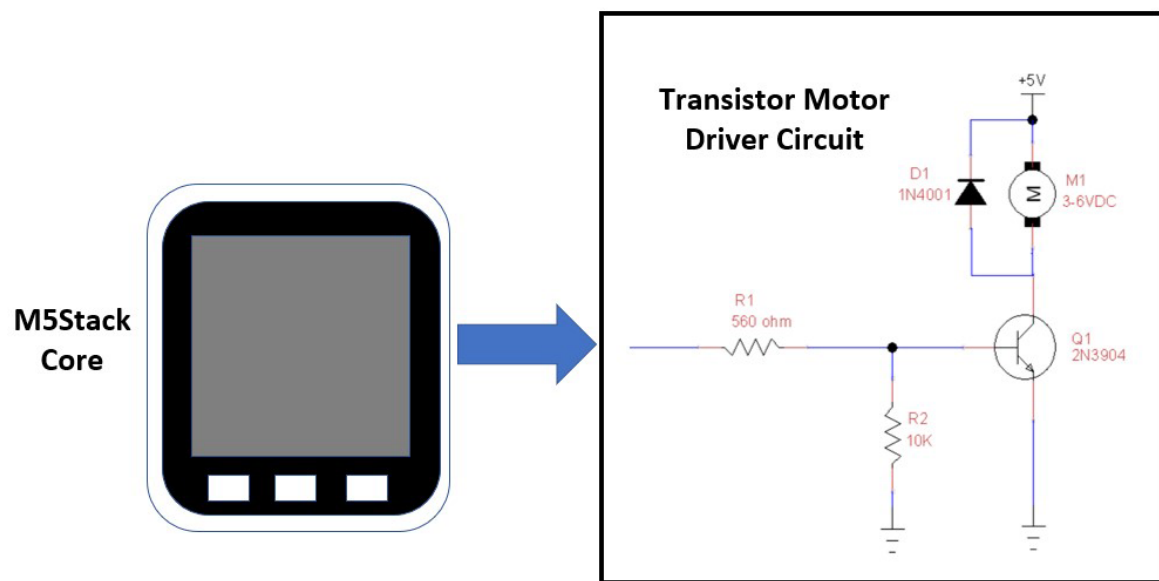
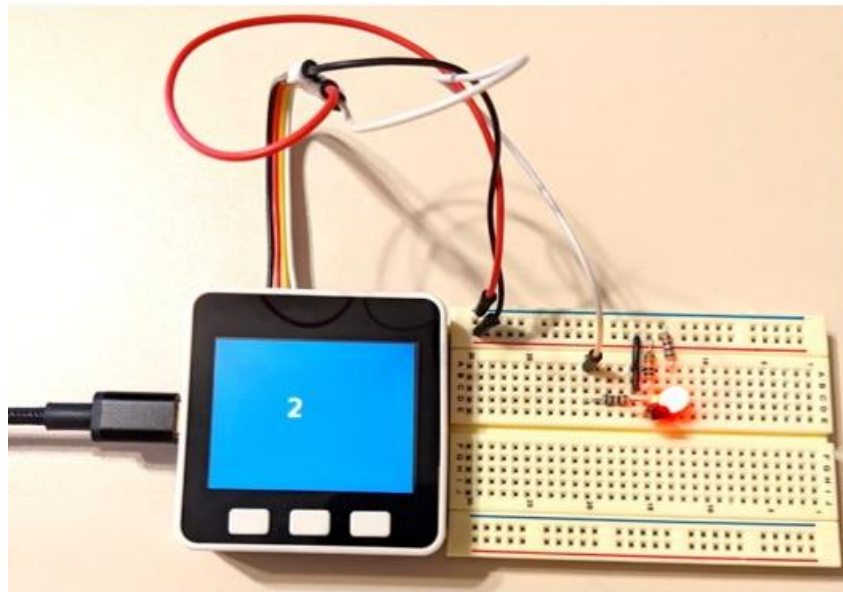


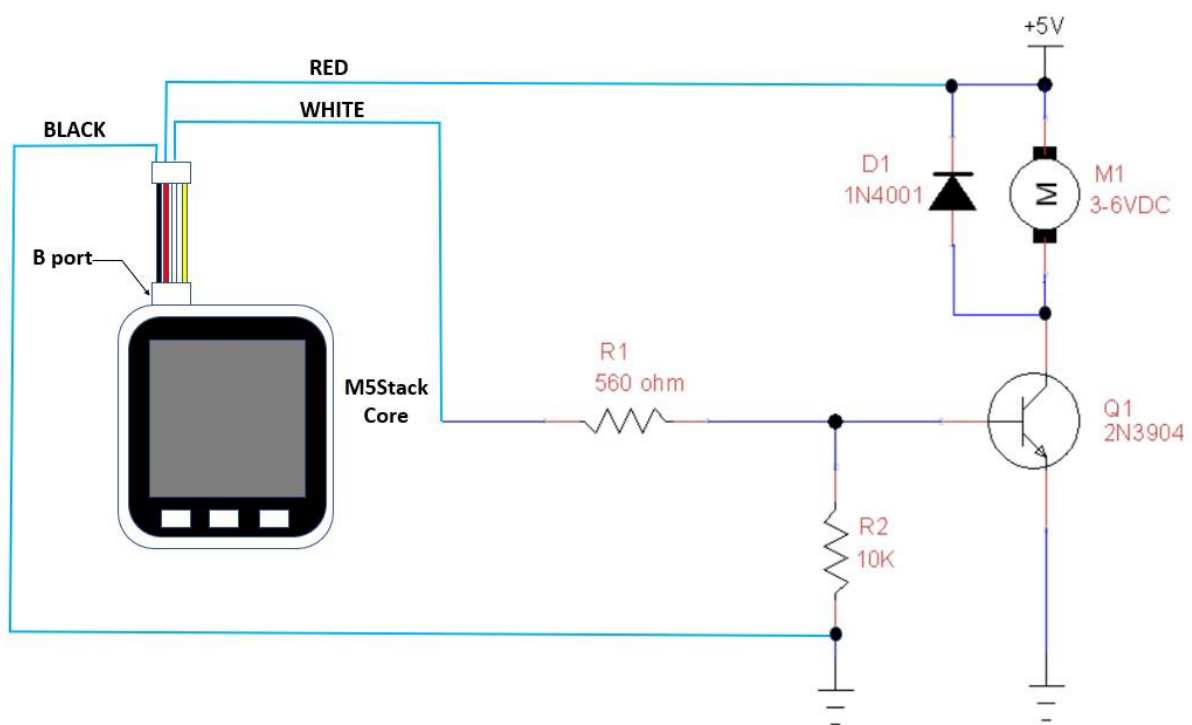
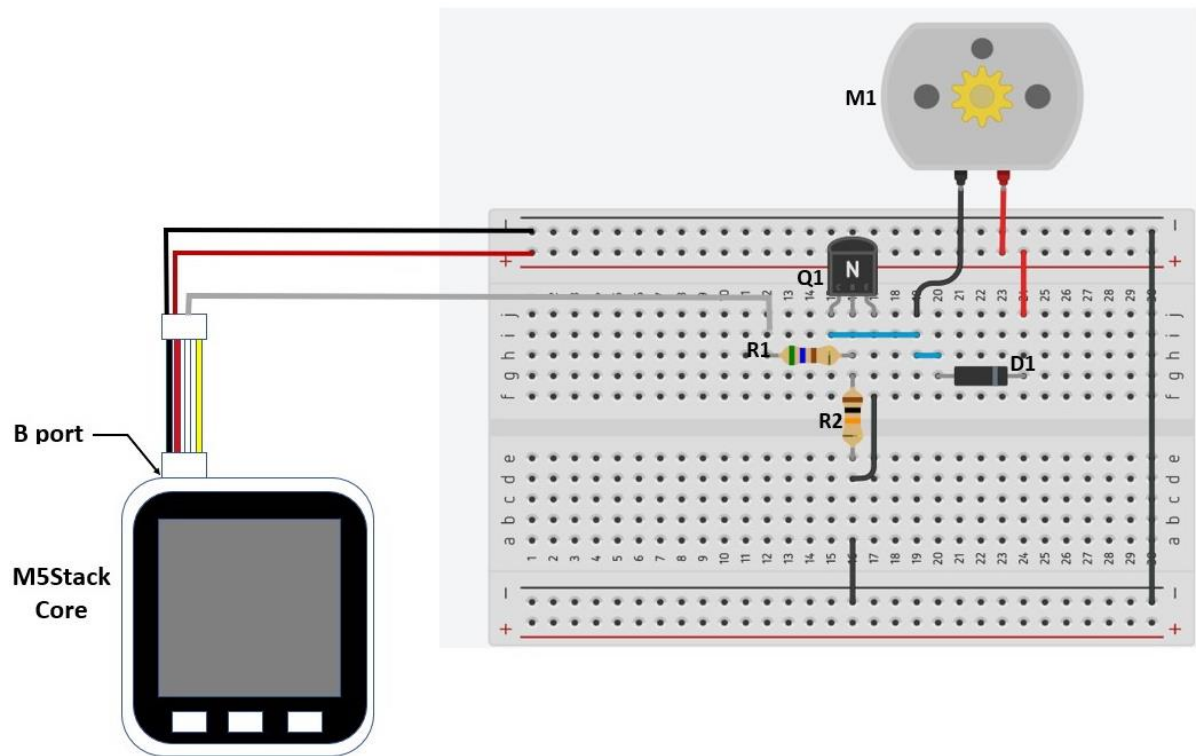
M5Stack
Core

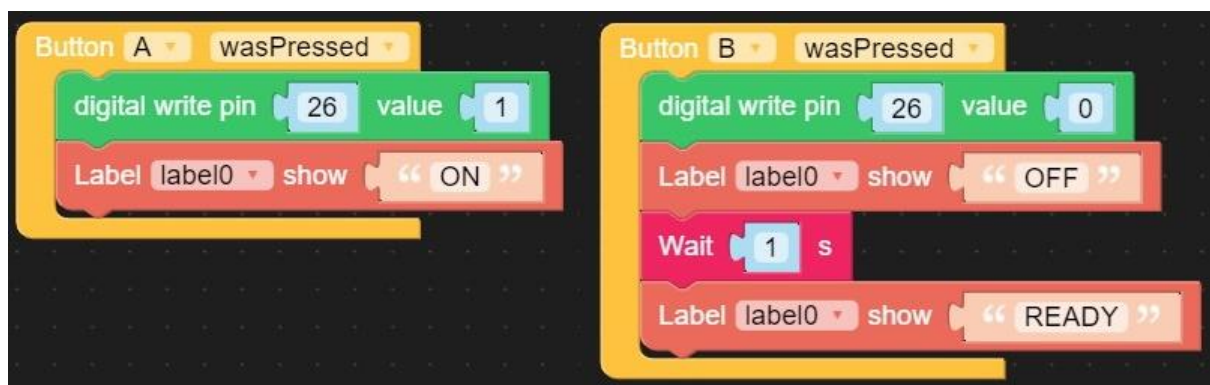
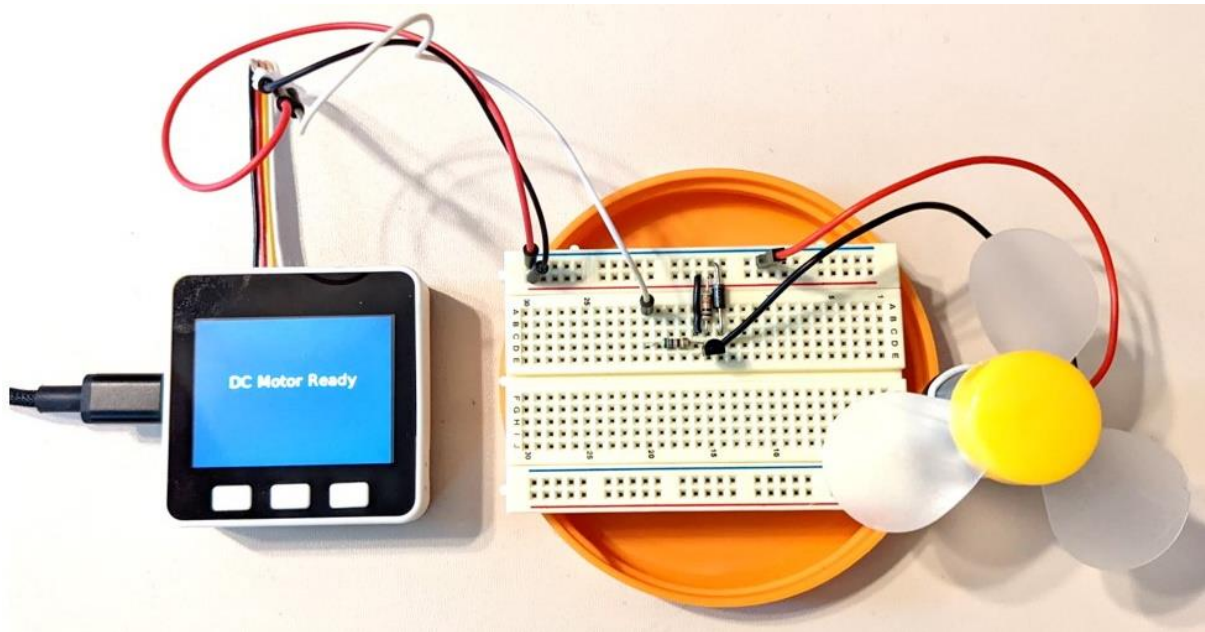














label1 Modules

name: label1

x: 34

y: 27

color:

text: DC Motor Controller

font: DejaVuSans 24 ▾

rotation: 0

layer: 14

label0 IoTCloud

name: label0

x: 124

y: 124

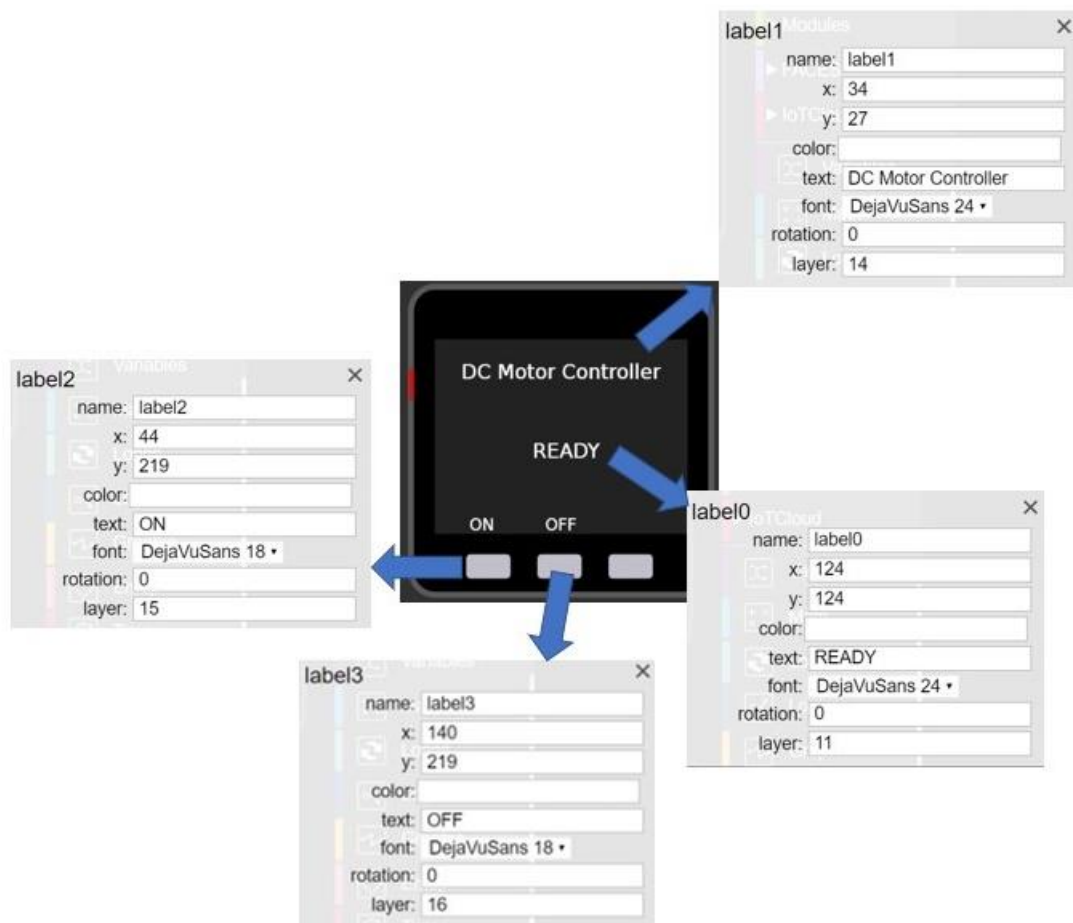
color:

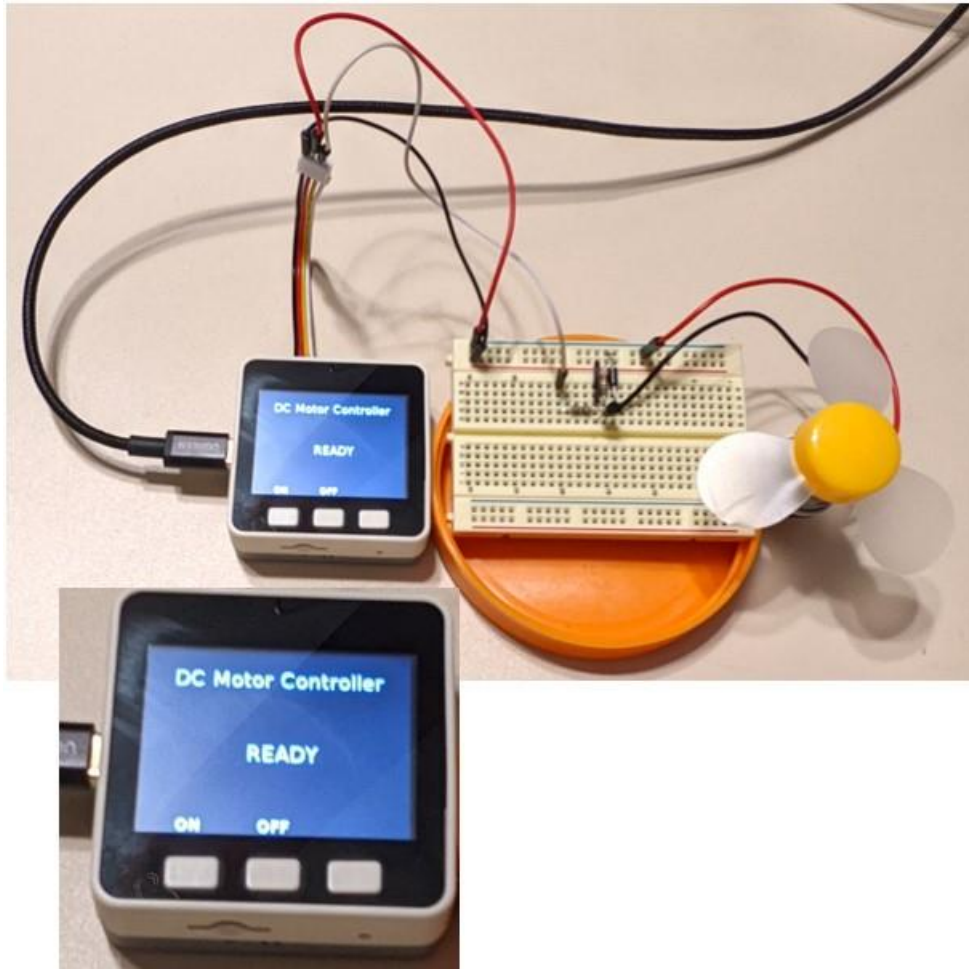
text: READY

font: DejaVuSans 24 ▾

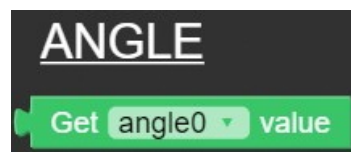
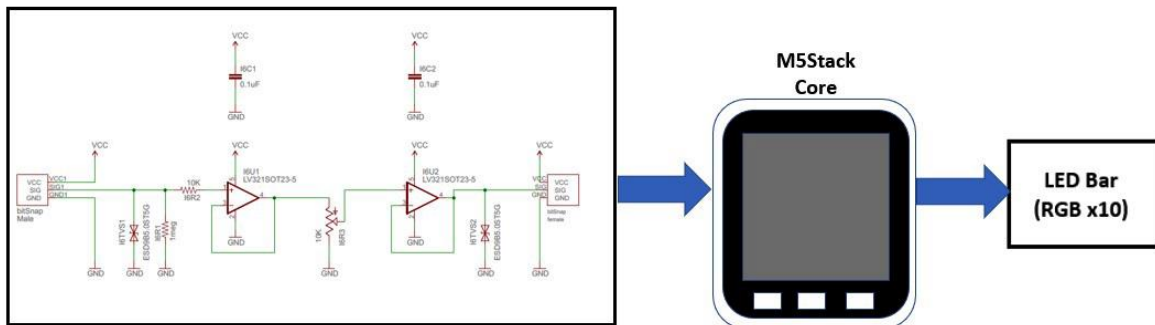
rotation: 0

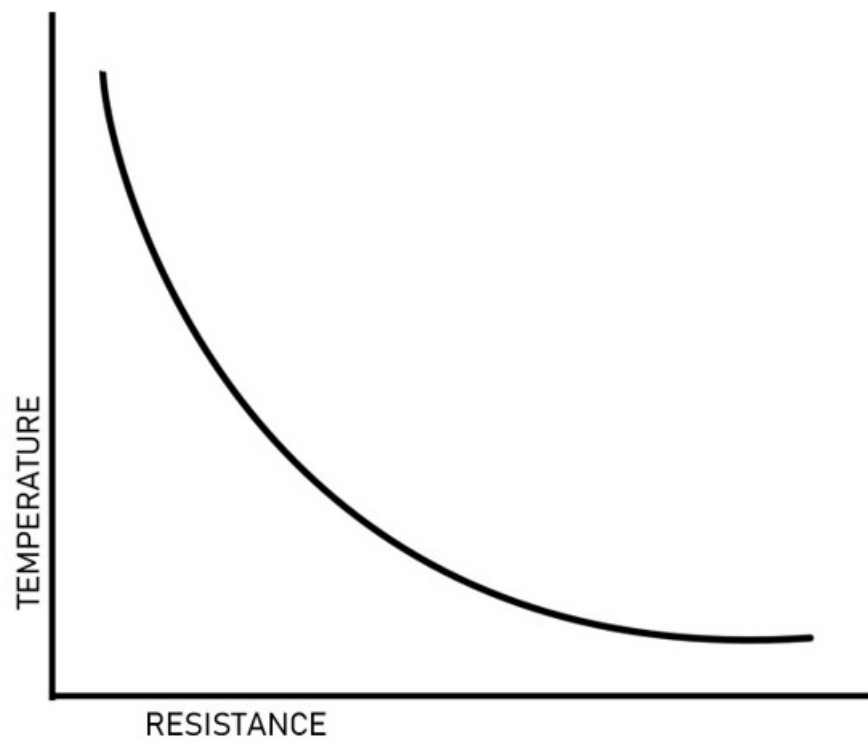
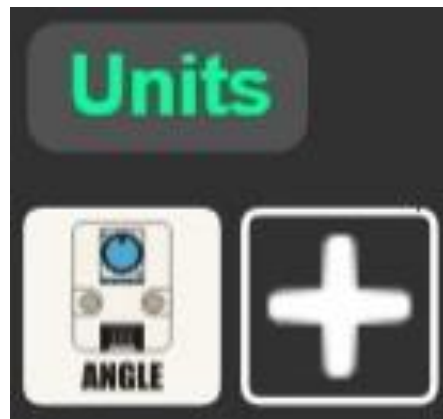
layer: 11

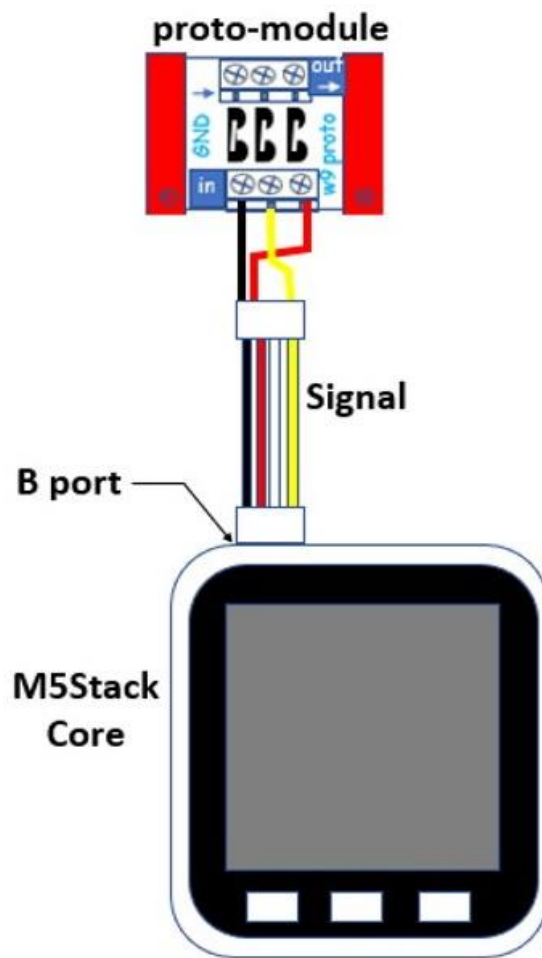


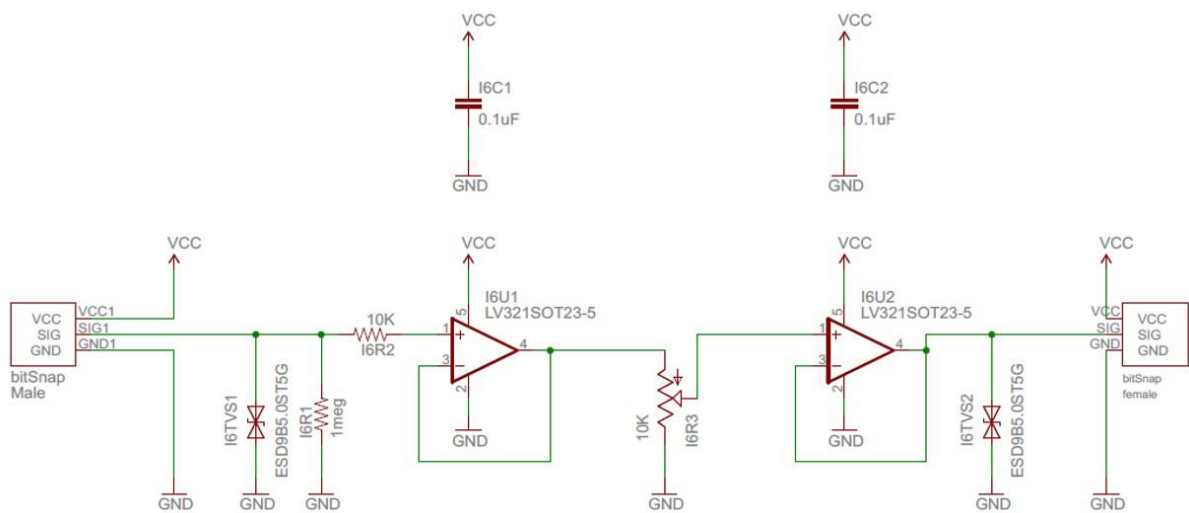
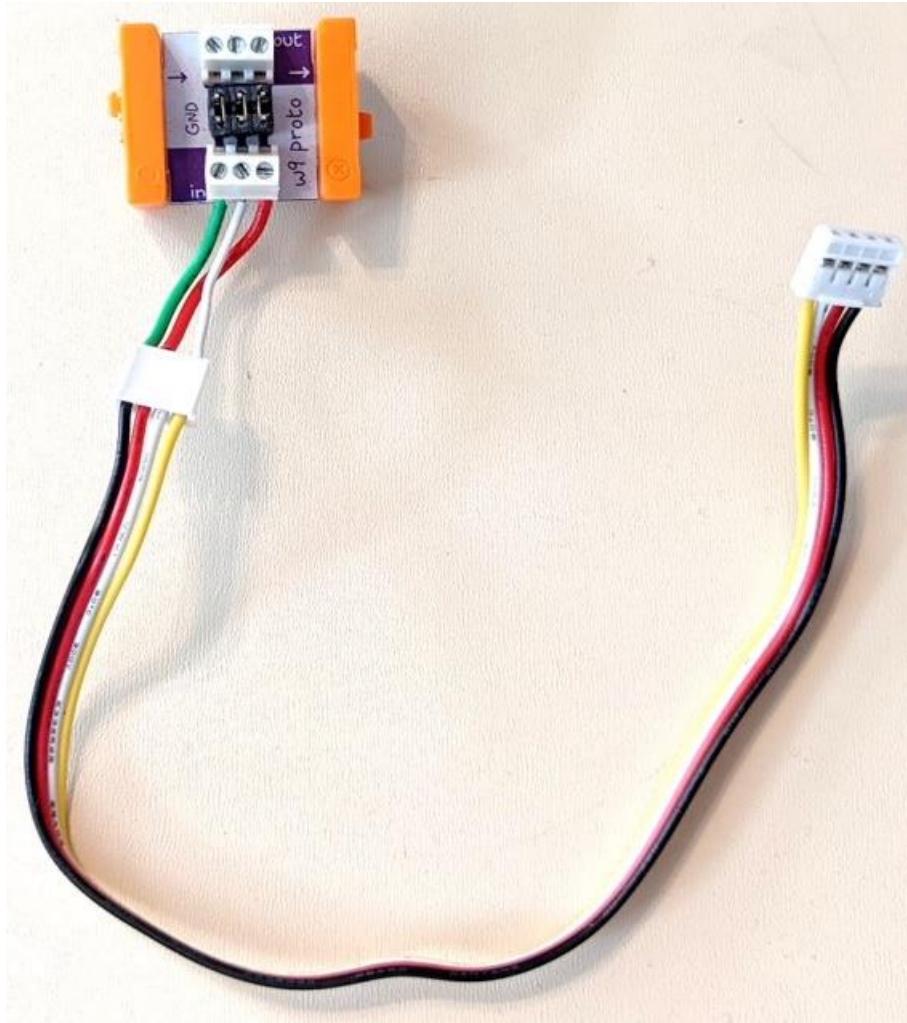


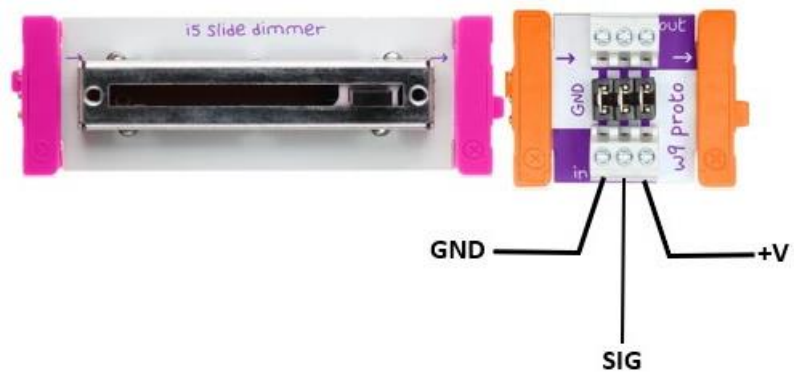
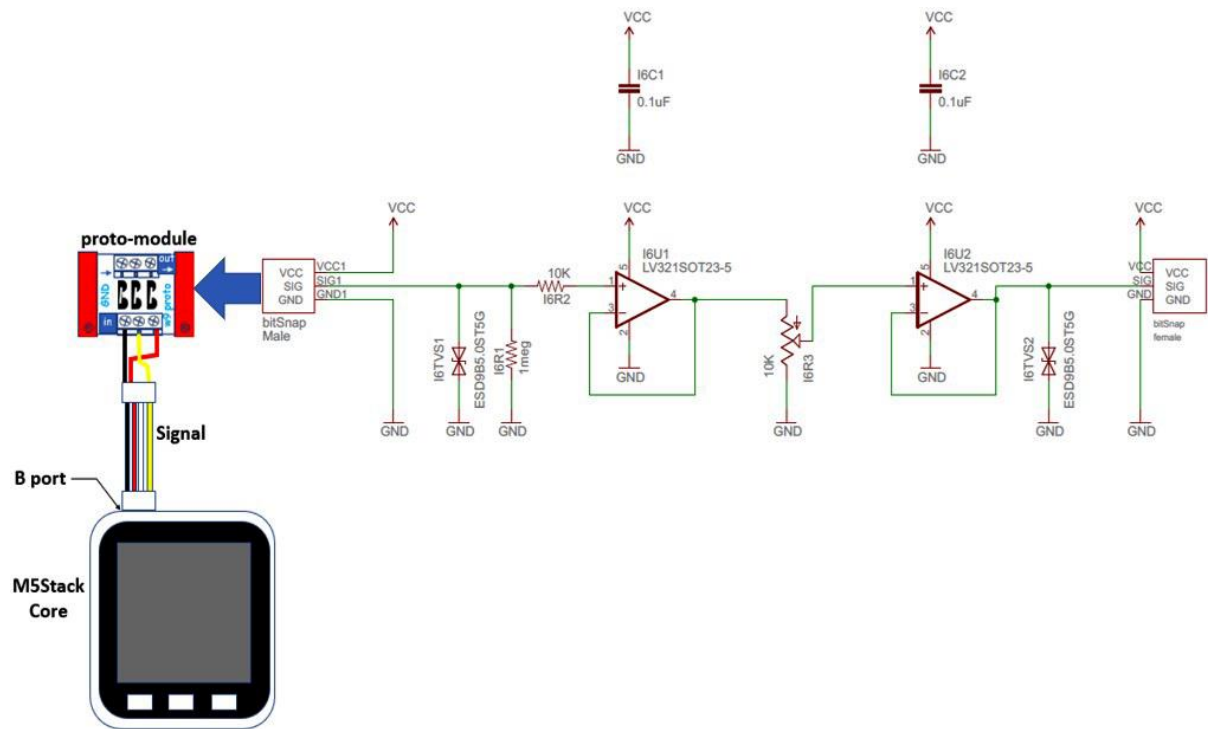
littleBits slide dimmer

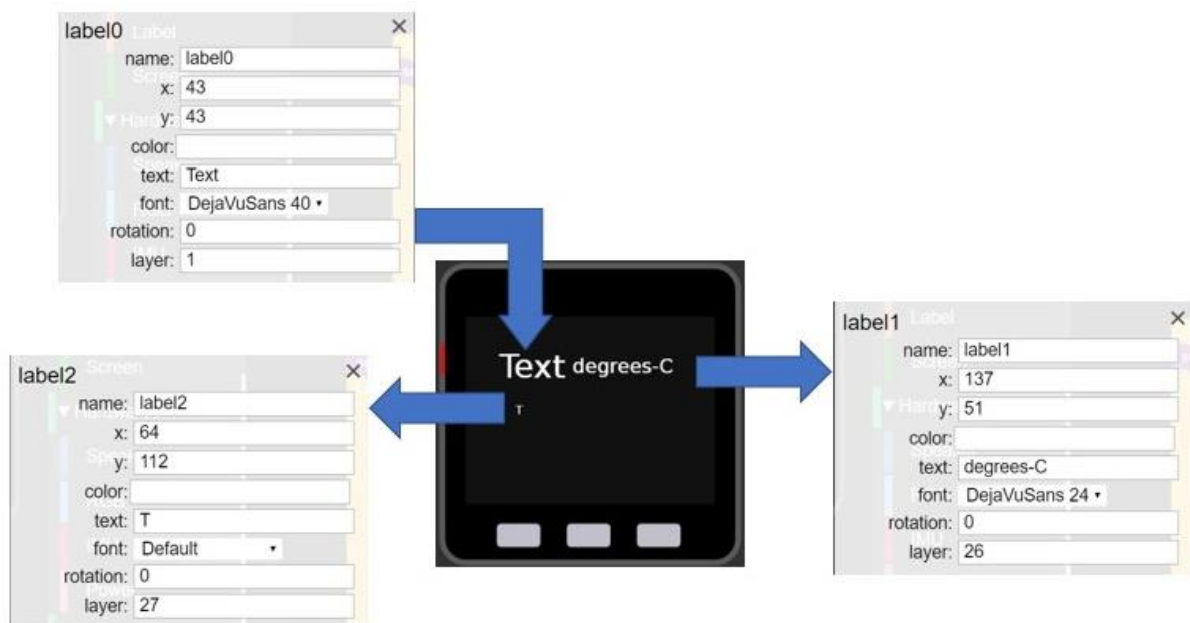
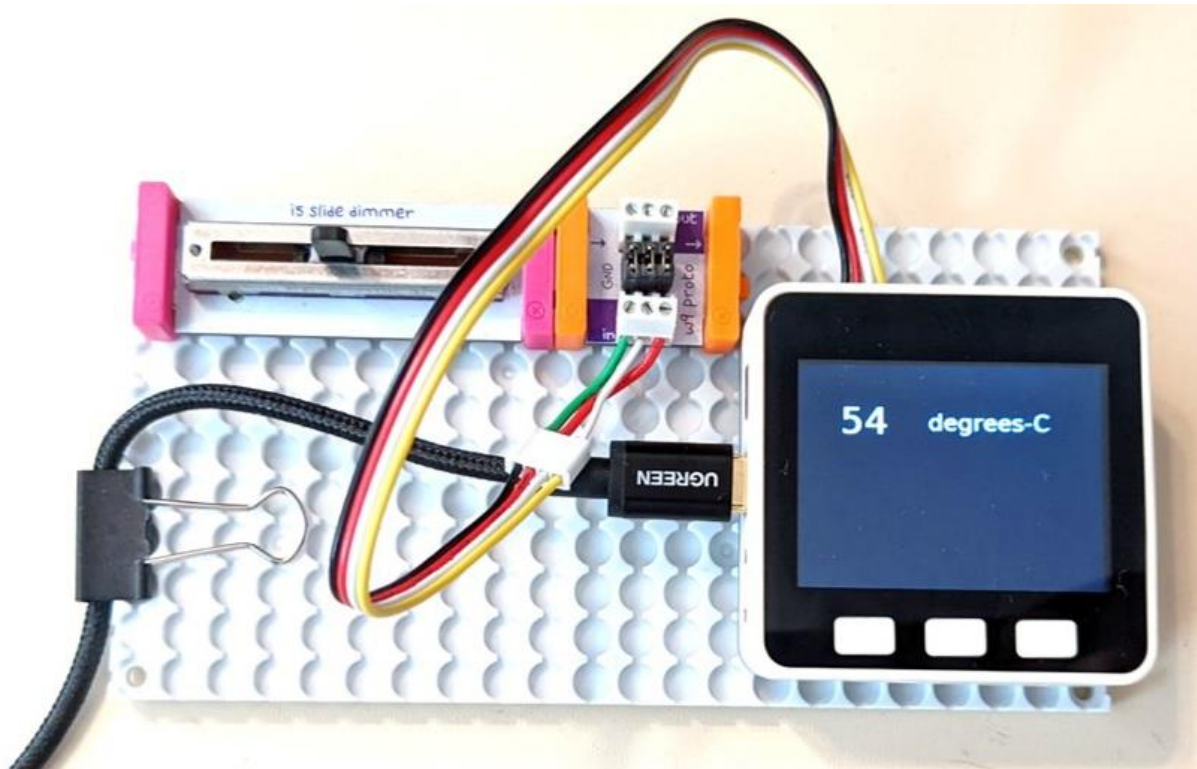


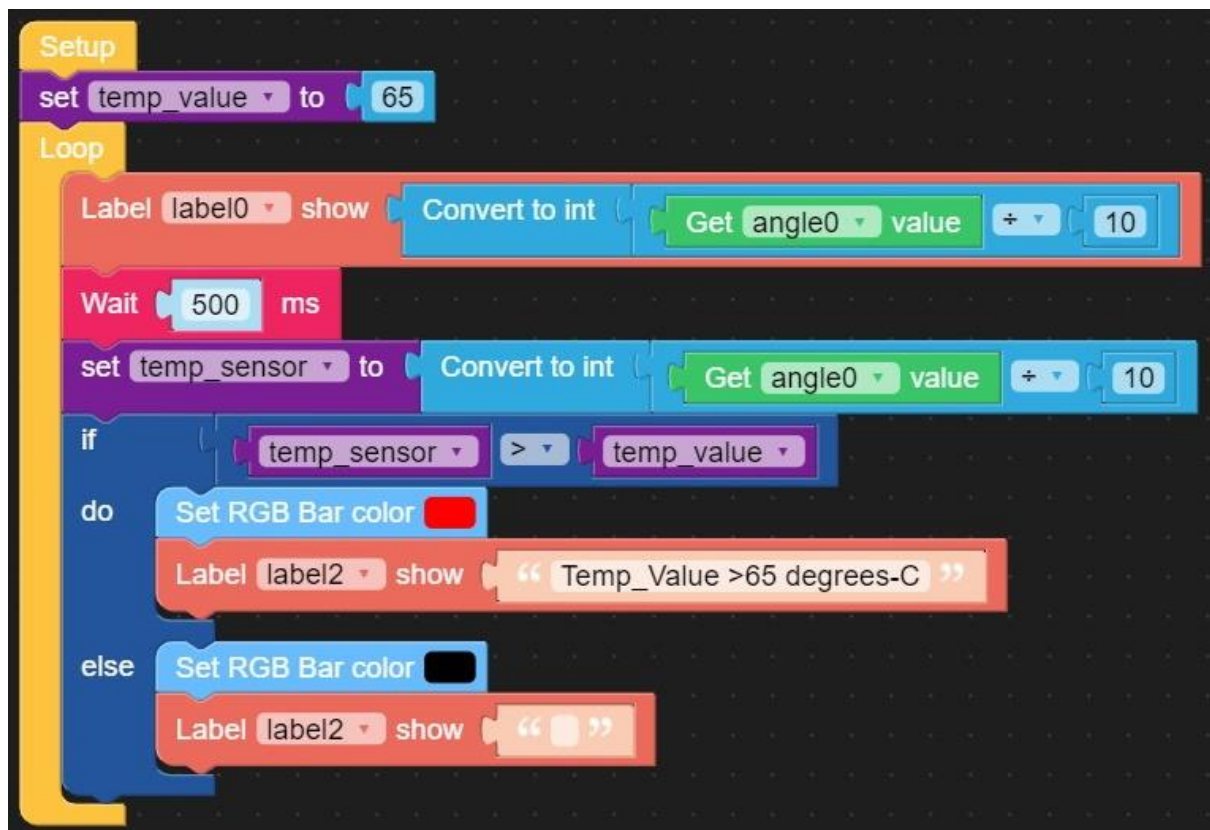


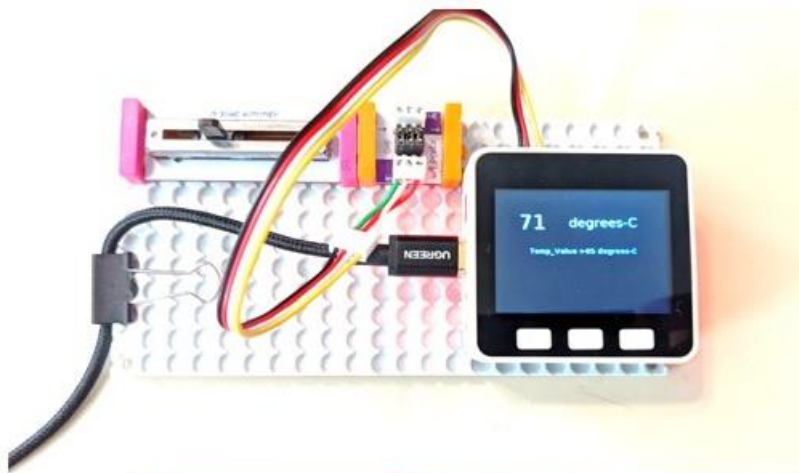


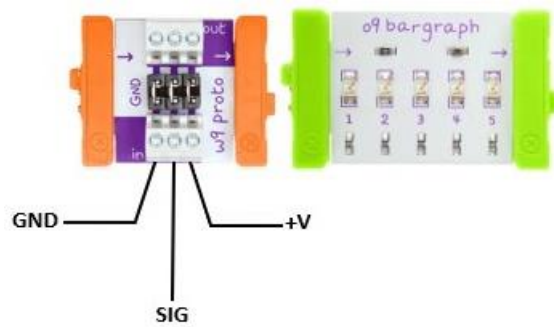
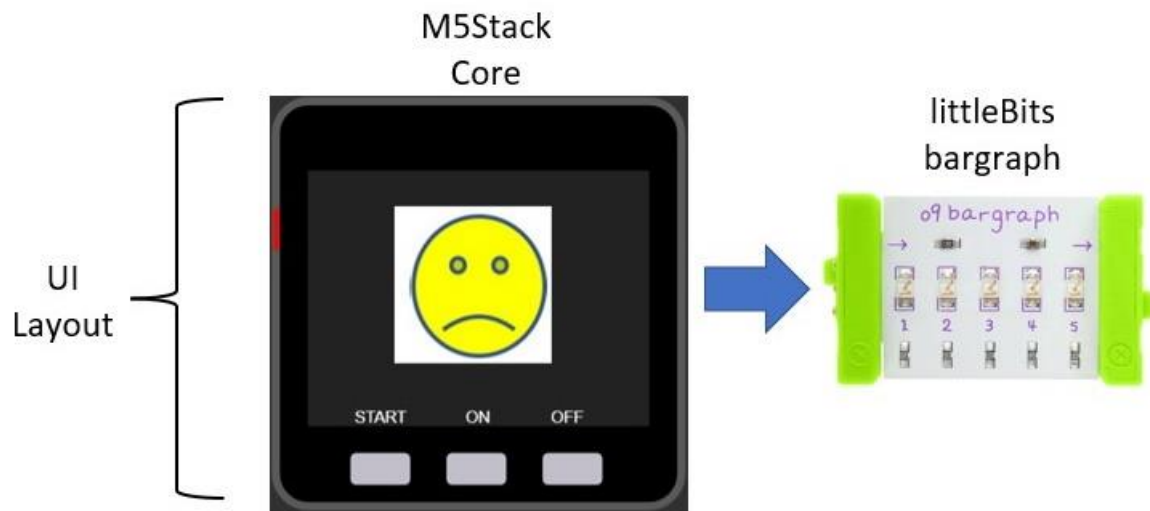


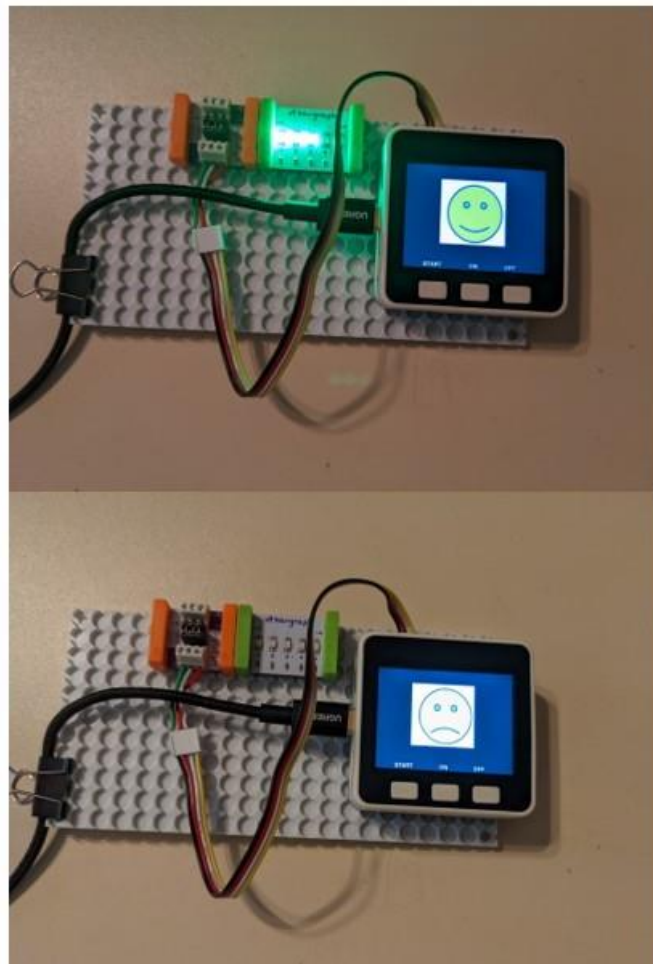
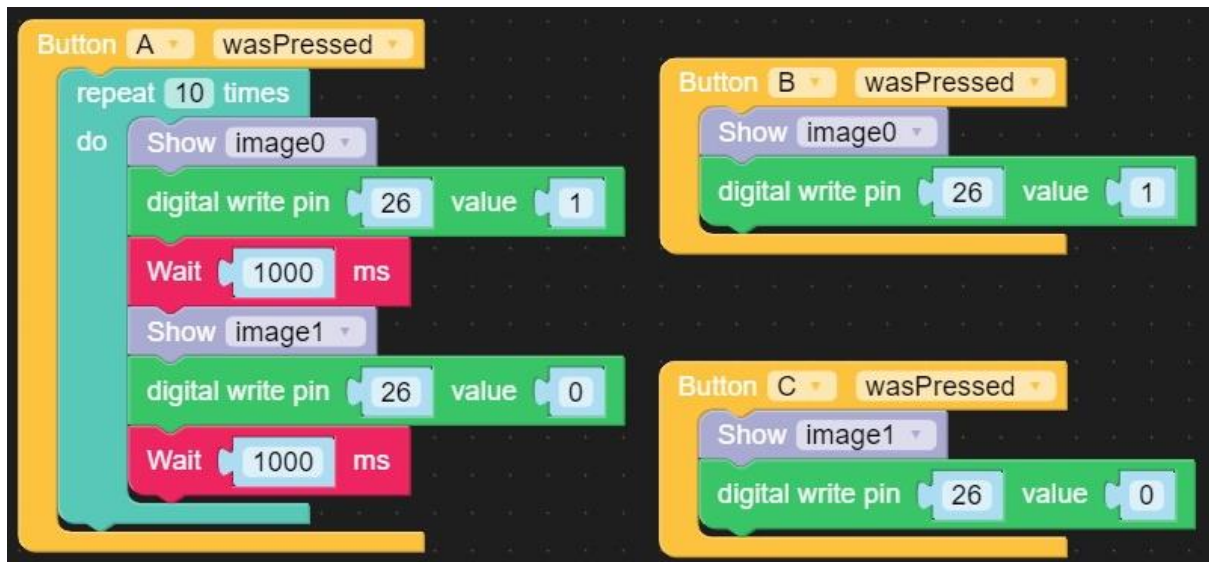


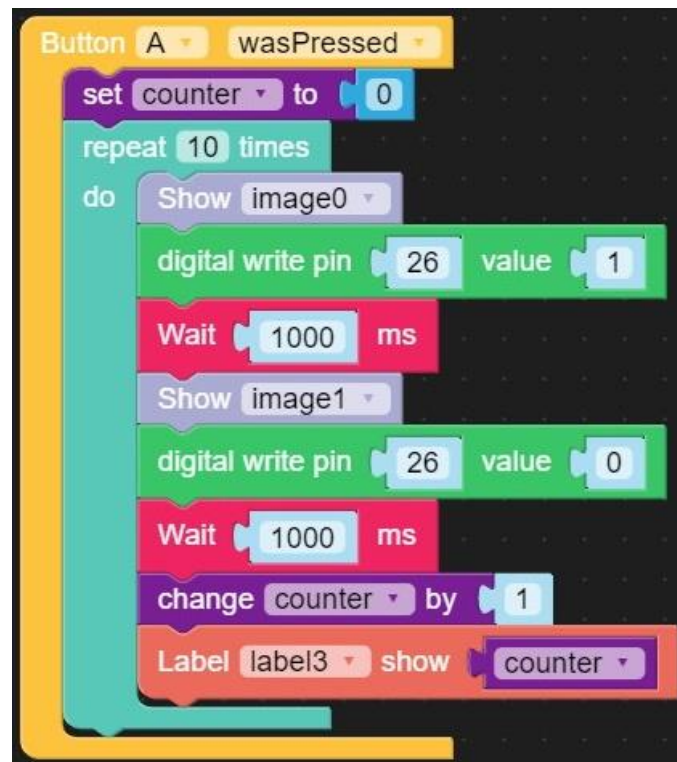
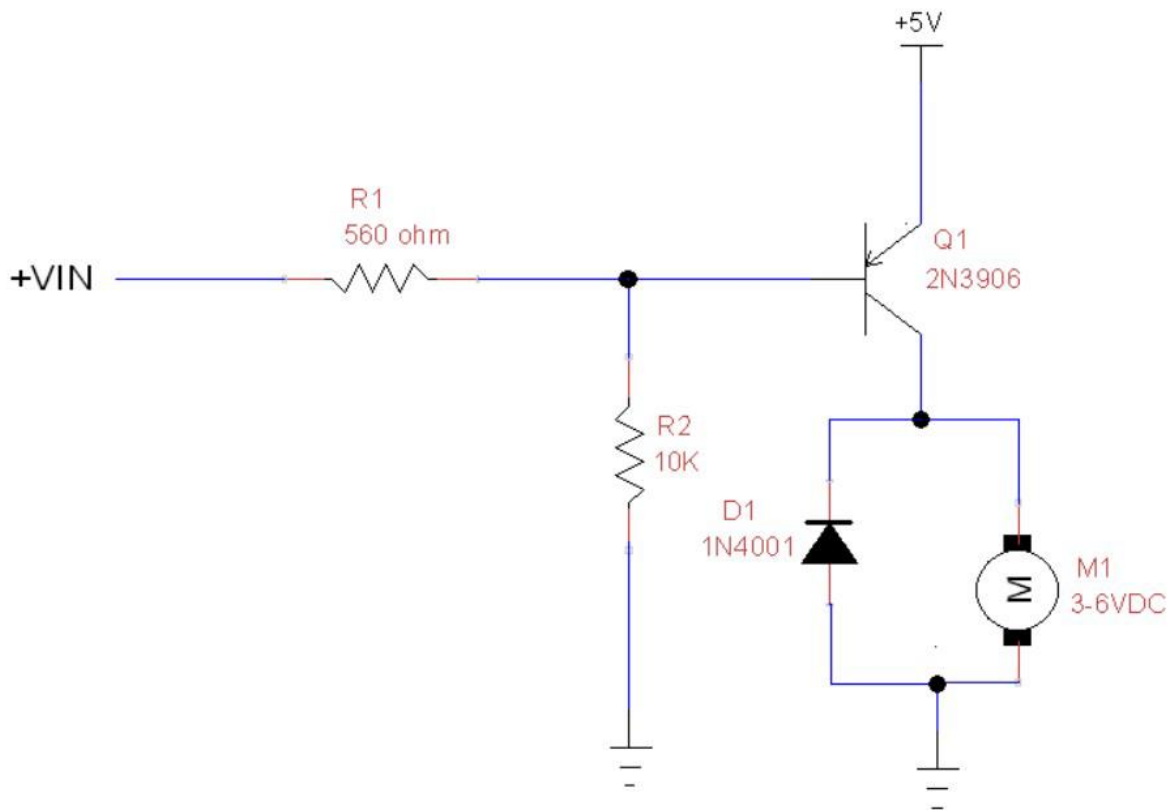




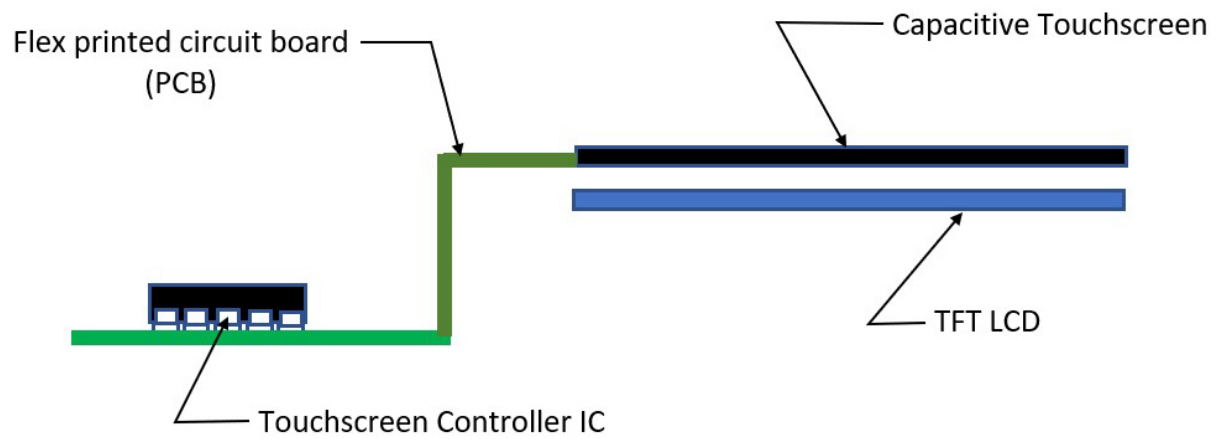


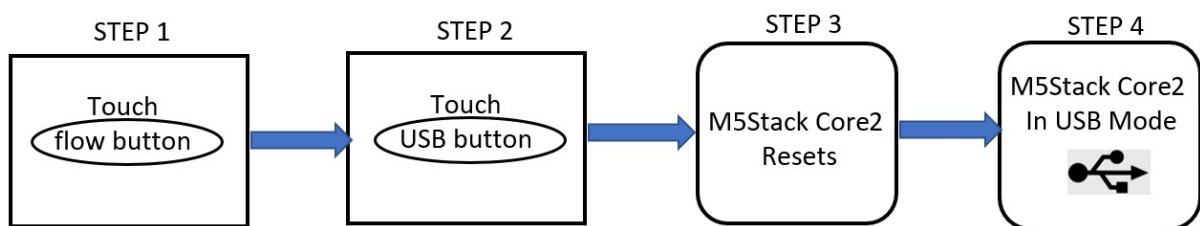






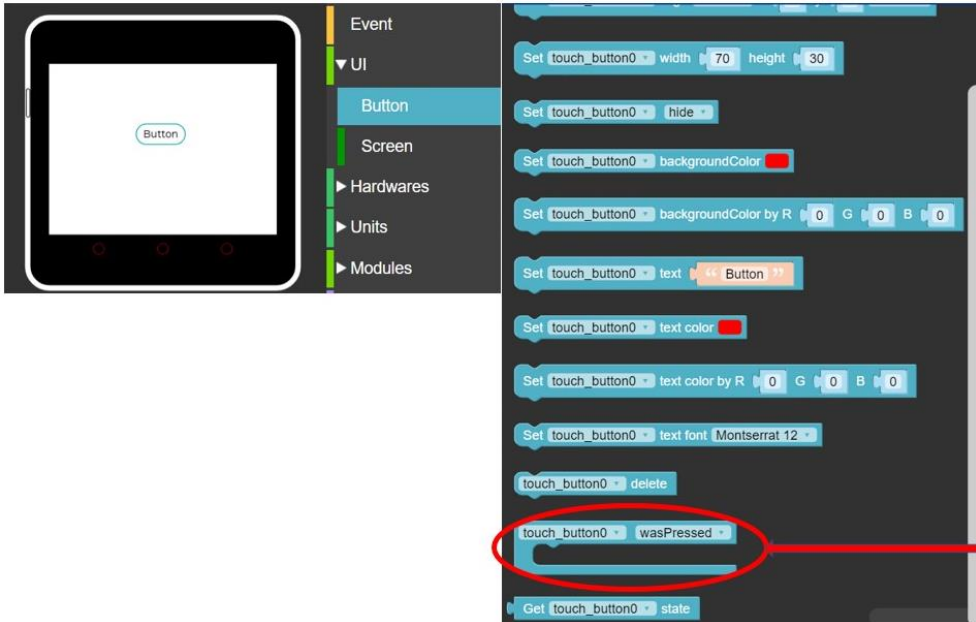
Chapter 6: M5Stack and Arduino





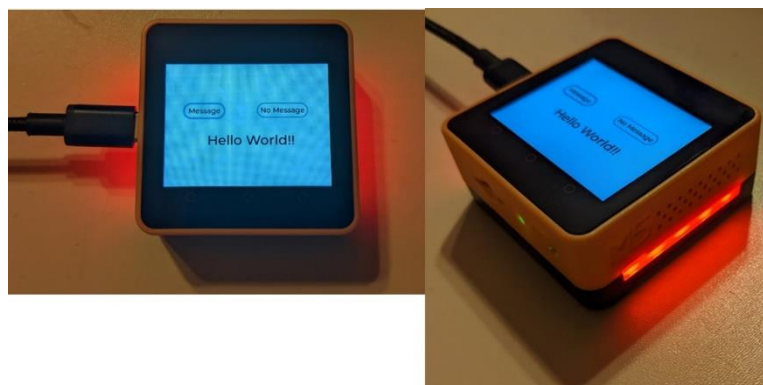
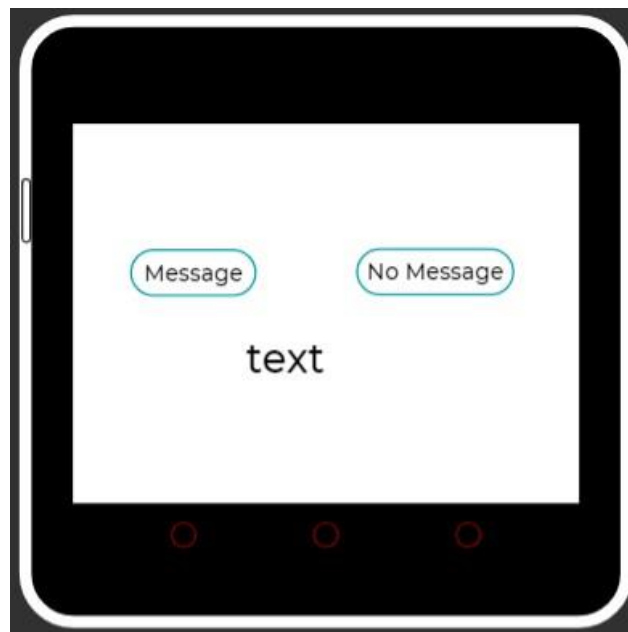


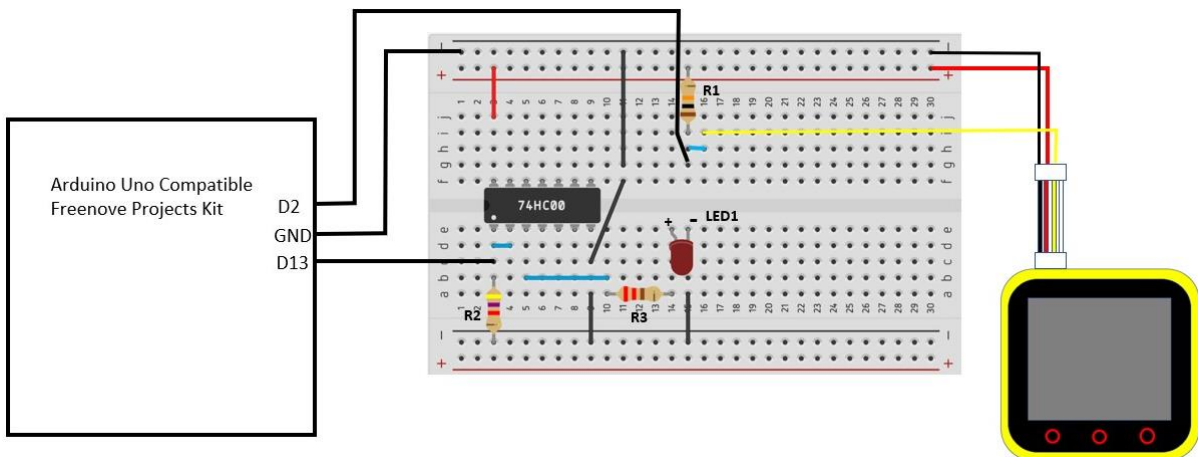
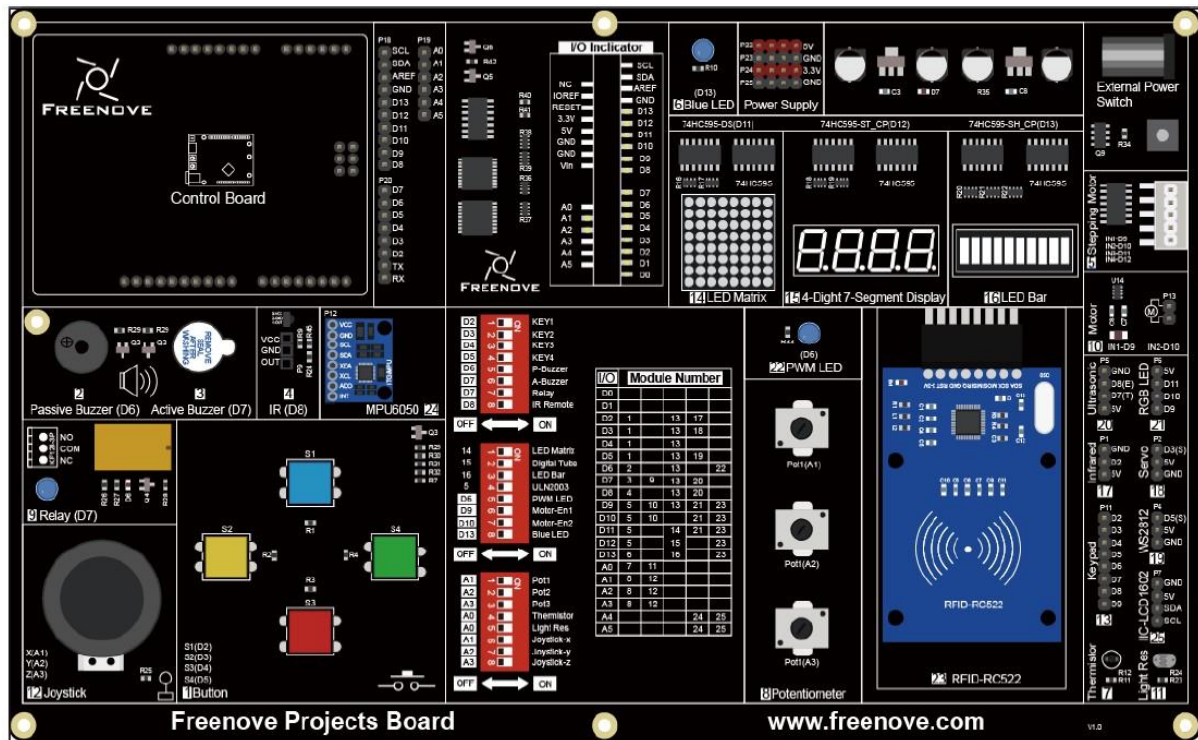
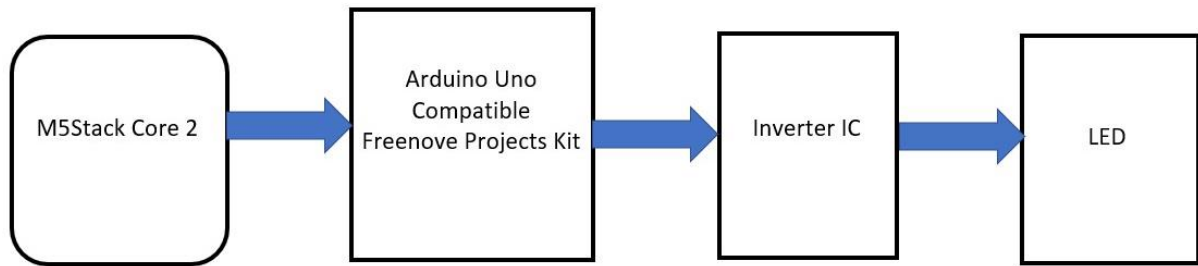
M5Stack Core 2

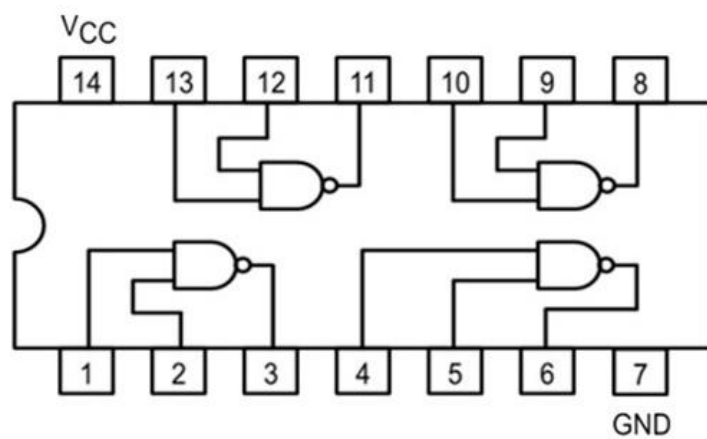
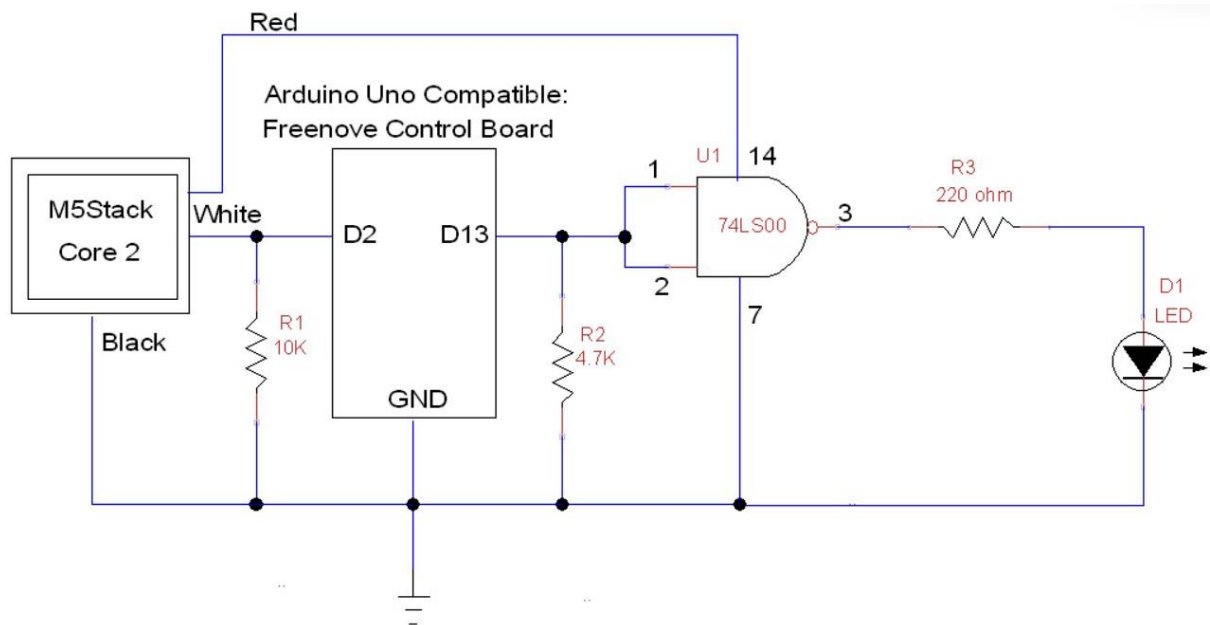


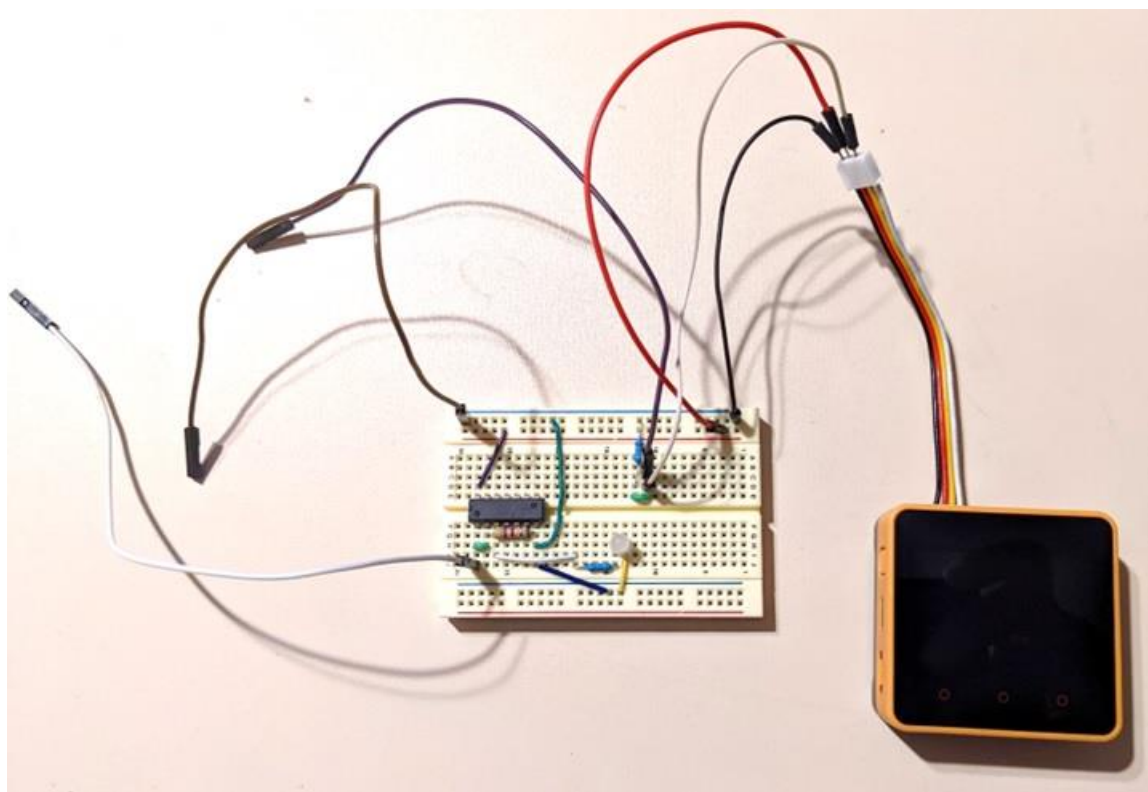
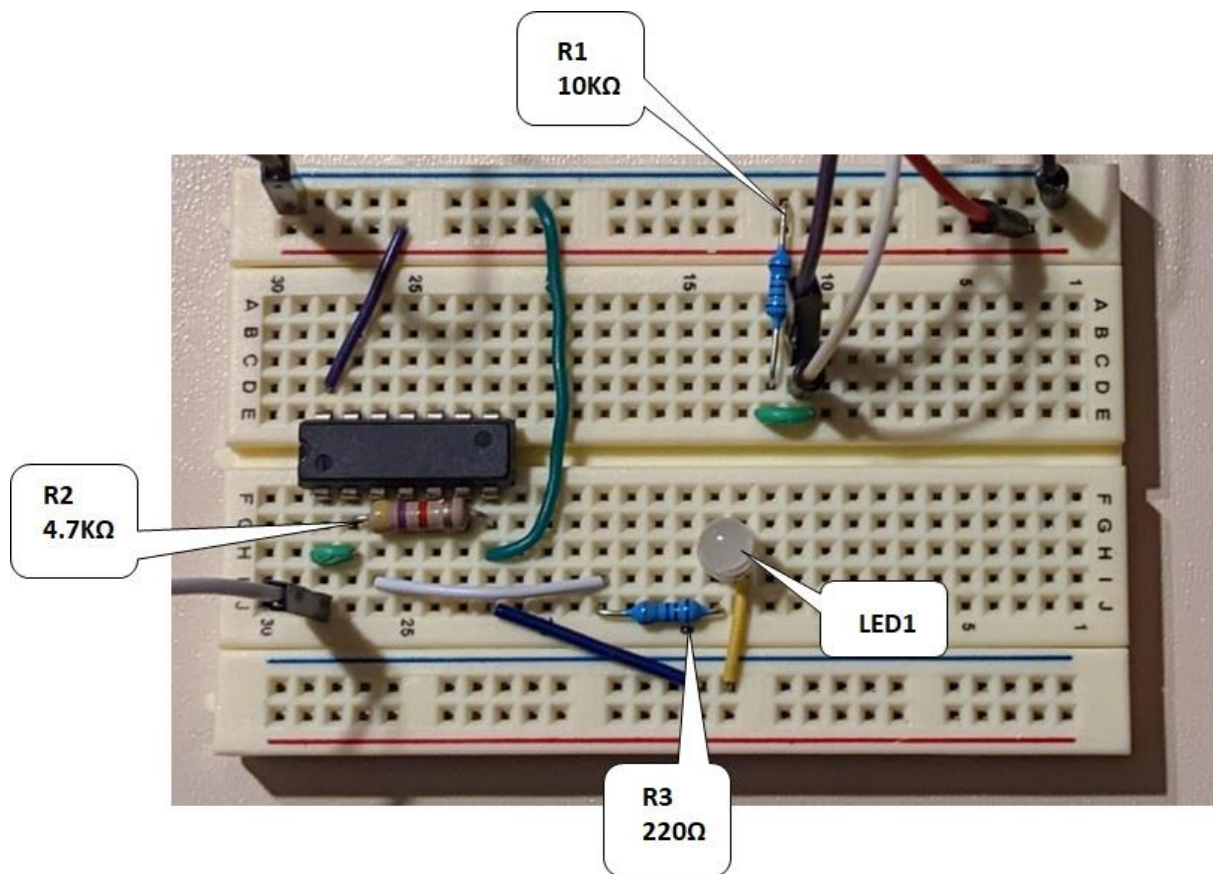
Scroll down

Select this block!







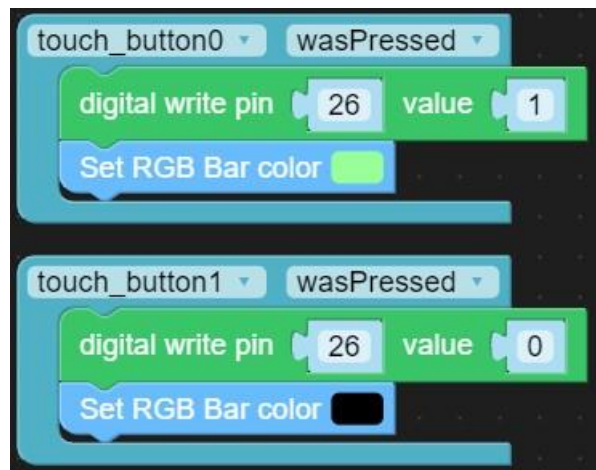


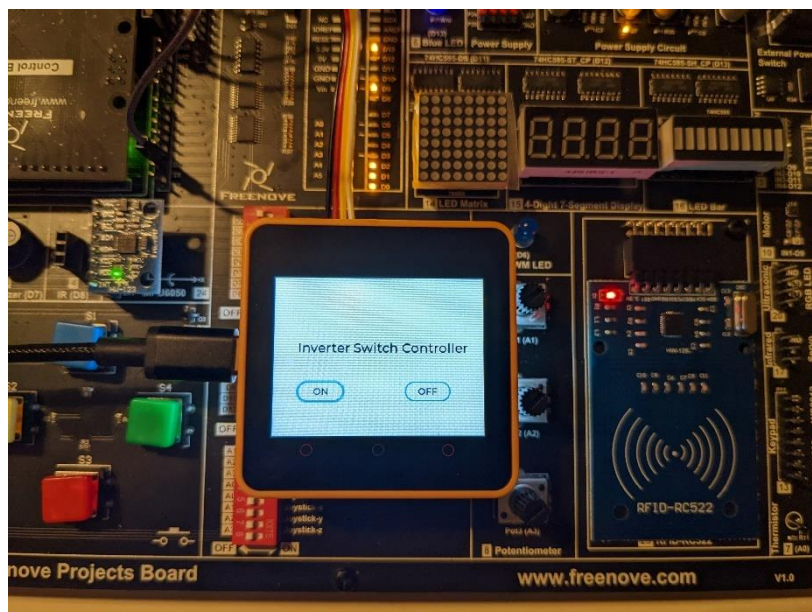
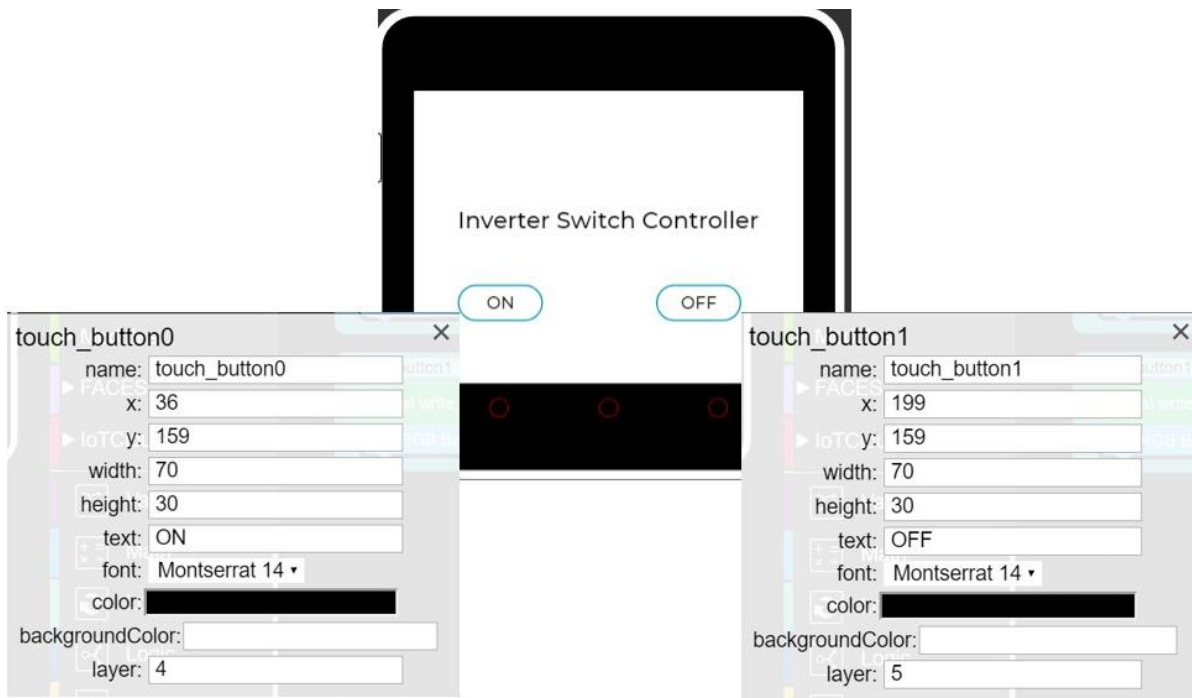
Solderless Breadboard Prototype	Arduino Uno Compatible Freenove Project kit
74LS00/74HC00 Input	D13
ground	ground
R1 10KΩ pulldown	D2

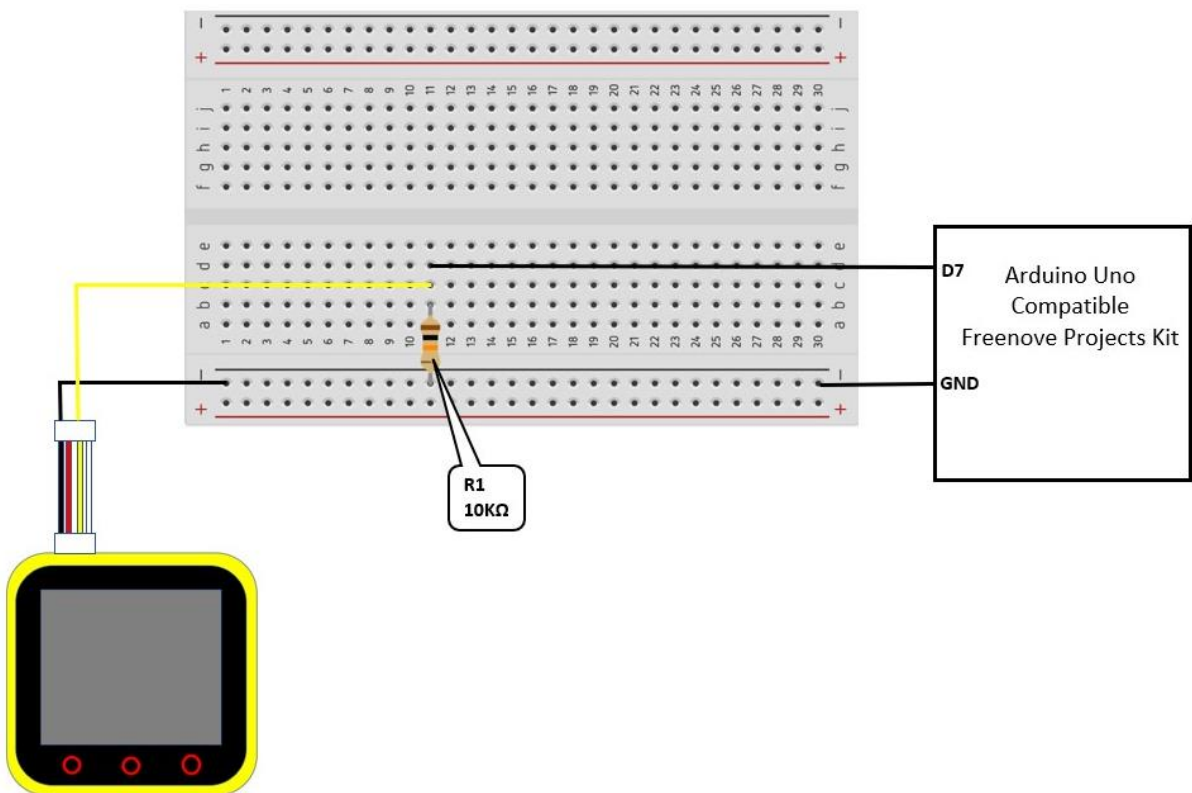
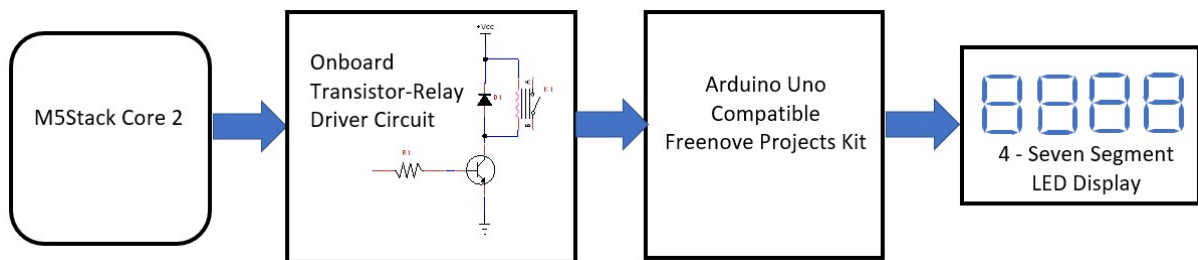
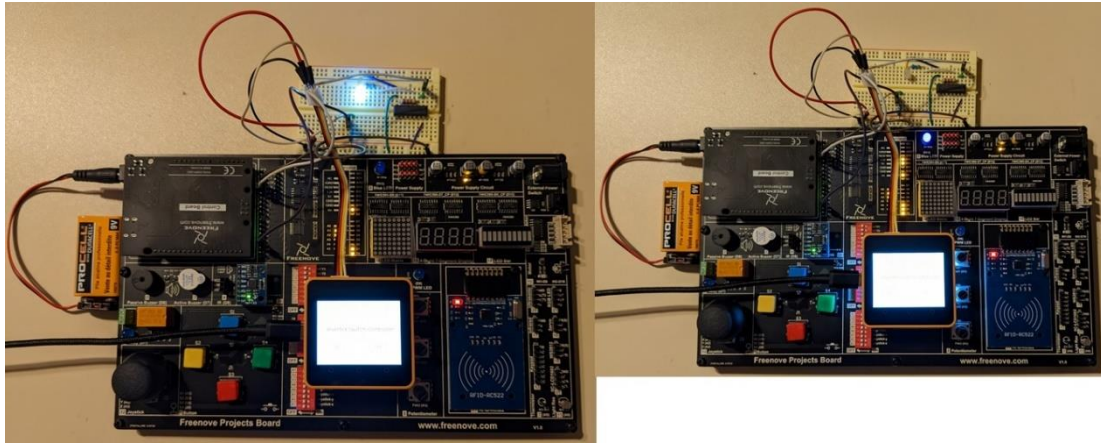
```

1  int M5StackPin = 2; // The M5Stack Core 2 pin
2  int ControlPin = 13; // The Inverter Output Control pin
3
4  void setup() {
5      pinMode(M5StackPin, INPUT); // Set the M5Stack Core 2 pin as an input
6      pinMode(ControlPin, OUTPUT); // Set the ControlPin as an output
7  }
8
9  void loop() {
10     if (digitalRead(M5StackPin) == HIGH) // if the M5Stack Core 2 UI ON button is pressed
11         digitalWrite(ControlPin, HIGH); // Switch ON the Inverter Output ControlPin
12     else // if the M5Stack Core 2 UI OFF button is pressed
13         digitalWrite(ControlPin, LOW); // Switch OFF the Inverter Output ControlPin
14 }

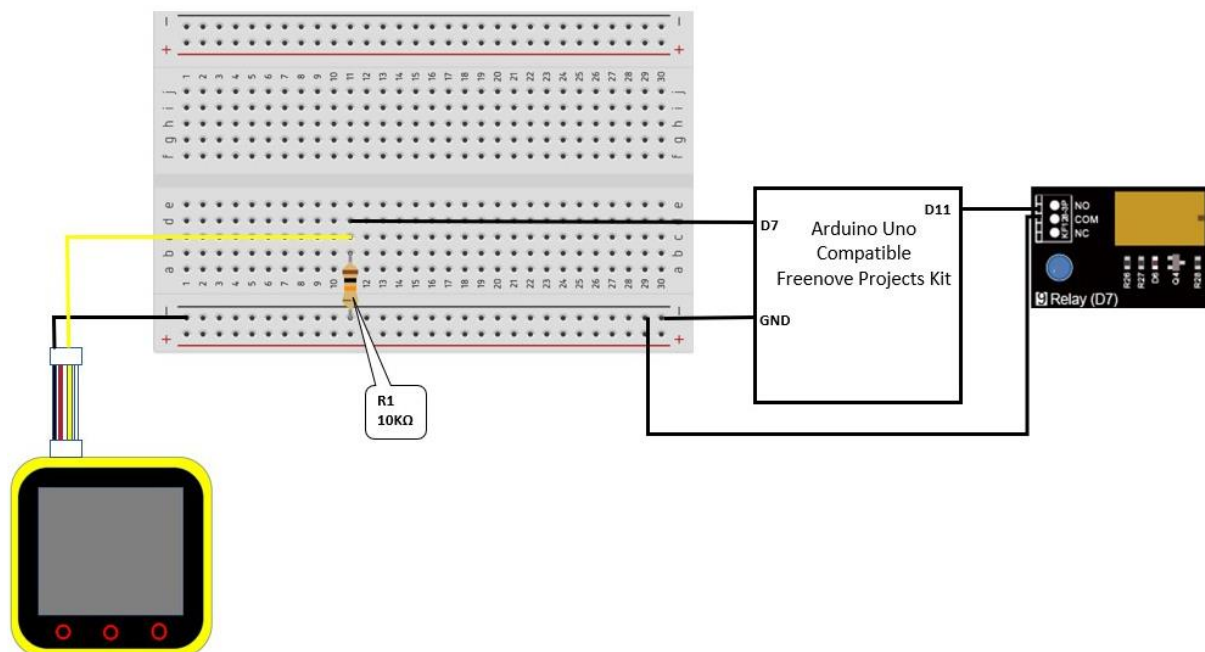
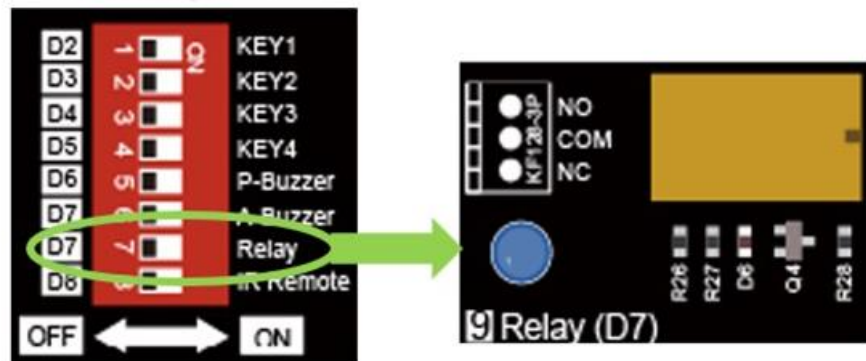
```

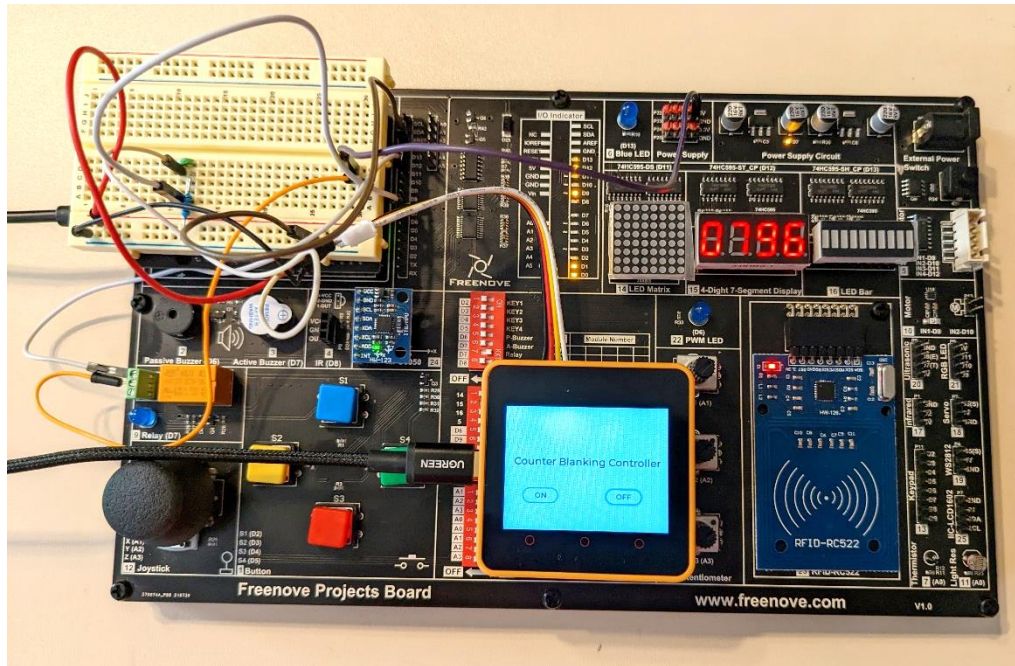






Slide the DIP switch to the ON position

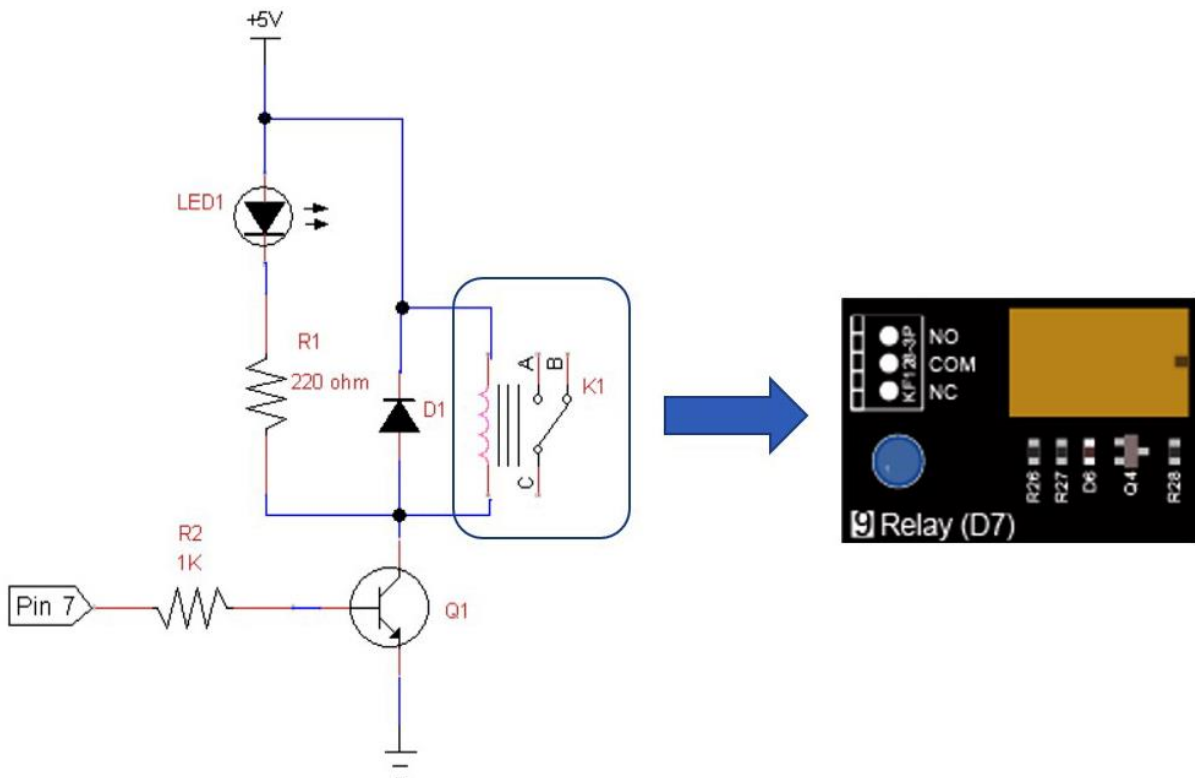
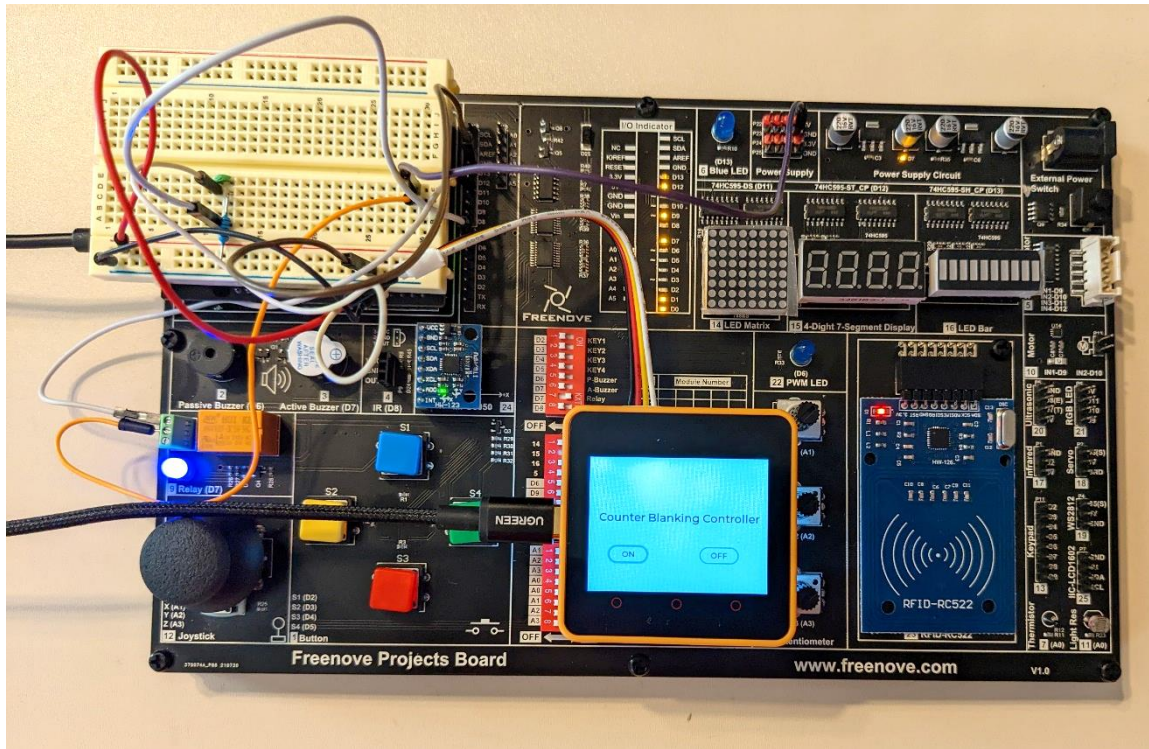


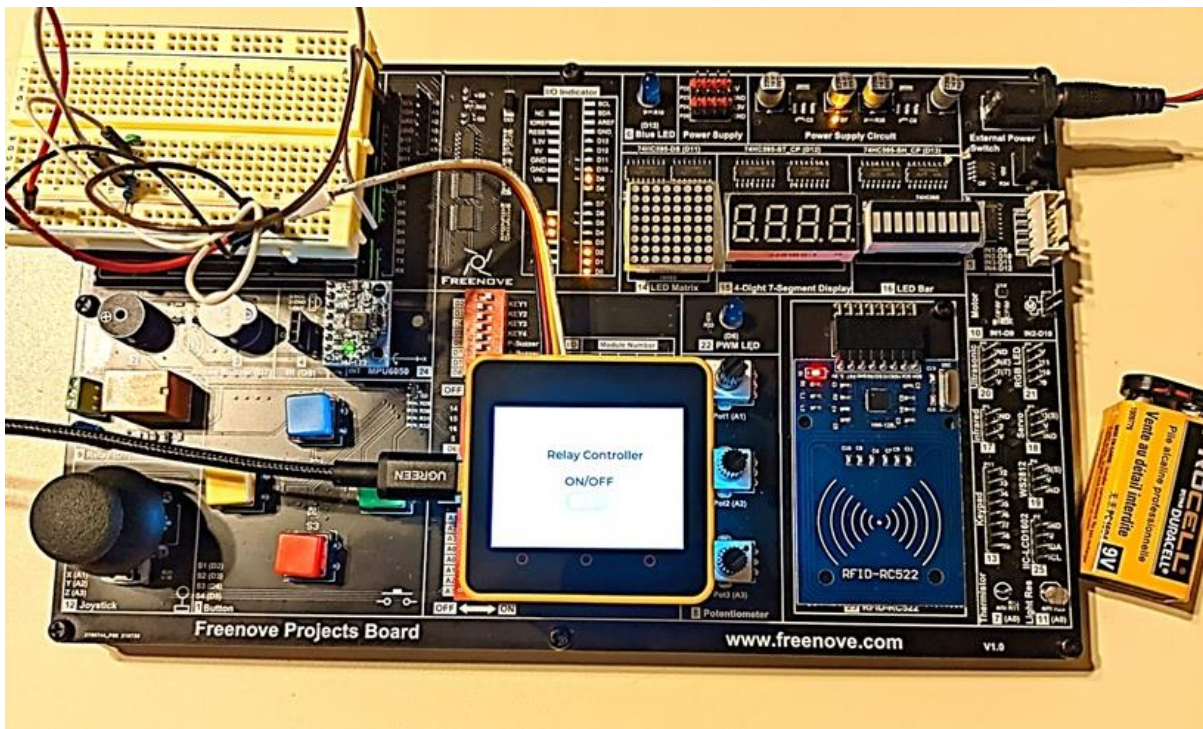


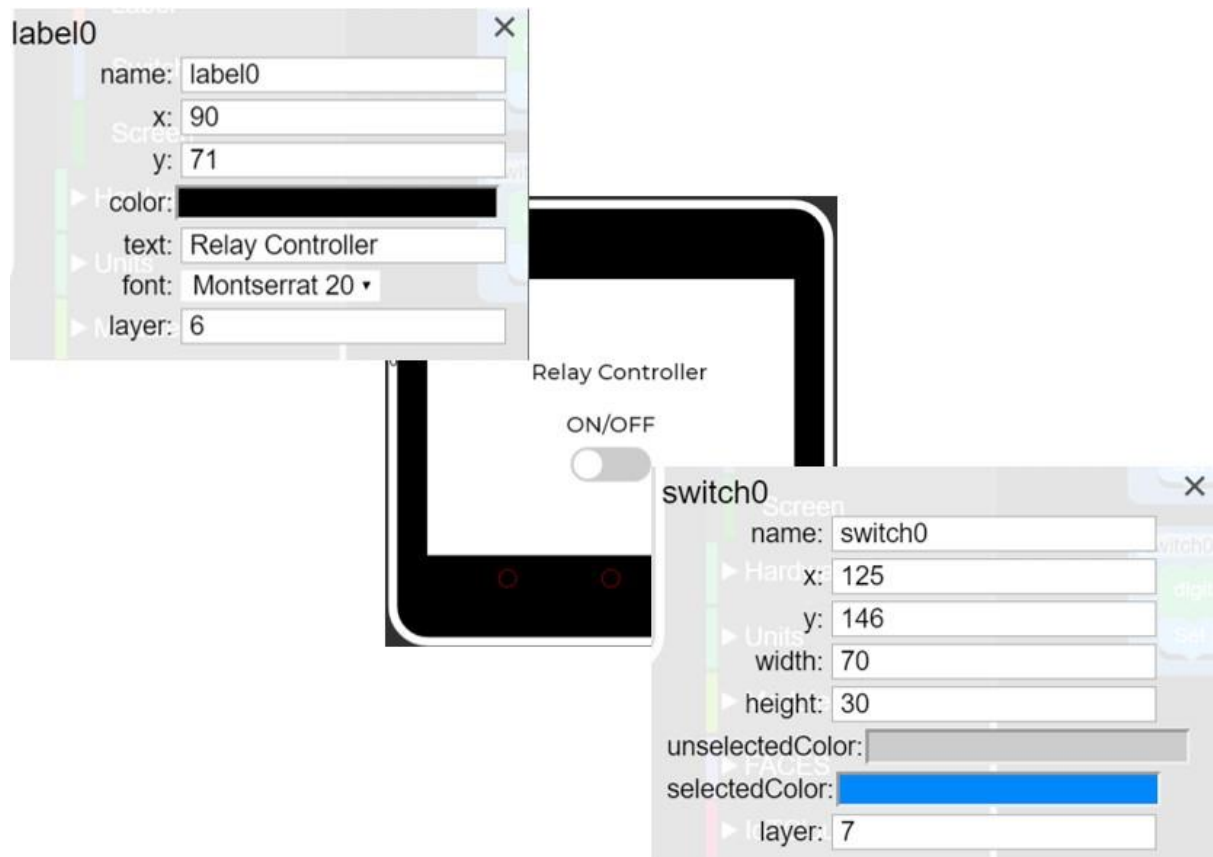
```

1  int latchPin = 12;           // Pin connected to ST_CP of 74HC595(Pin12)
2  int clockPin = 13;          // Pin connected to SH_CP of 74HC595(Pin11)
3  int dataPin = 11;           // Pin connected to DS of 74HC595(Pin14)
4  int countValue = 0;         // Digital tube display value
5
6  byte num[] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, //0-7
7                0x80, 0x90, 0x88, 0x83, 0xc6, 0xa1, 0x86, 0x8e
8                }; //8-F
9
10 void setup() {
11     // set pins to output
12     pinMode(latchPin, OUTPUT);
13     pinMode(clockPin, OUTPUT);
14     pinMode(dataPin, OUTPUT);
15 }
16
17 void loop() {
18     for (int j = 0; j < 250; j++) {
19         ShowCount(countValue);
20     }
21     countValue++;
22 }
23
24 void ShowCount(int value) {
25     DigitalTube_MSBFIRST(0, num[value % 10000 / 1000]); delay(1);
26     DigitalTube_MSBFIRST(1, num[value % 1000 / 100]); delay(1);
27     DigitalTube_MSBFIRST(2, num[value % 100 / 10]); delay(1);
28     DigitalTube_MSBFIRST(3, num[value % 10]); delay(1);
29 }
30
31 void DigitalTube_MSBFIRST(int number, byte value) {
32     digitalWrite(latchPin, LOW);
33     shiftOut(dataPin, clockPin, MSBFIRST, 0x01 << number);
34     shiftOut(dataPin, clockPin, MSBFIRST, value);
35     digitalWrite(latchPin, HIGH);
36 }

```



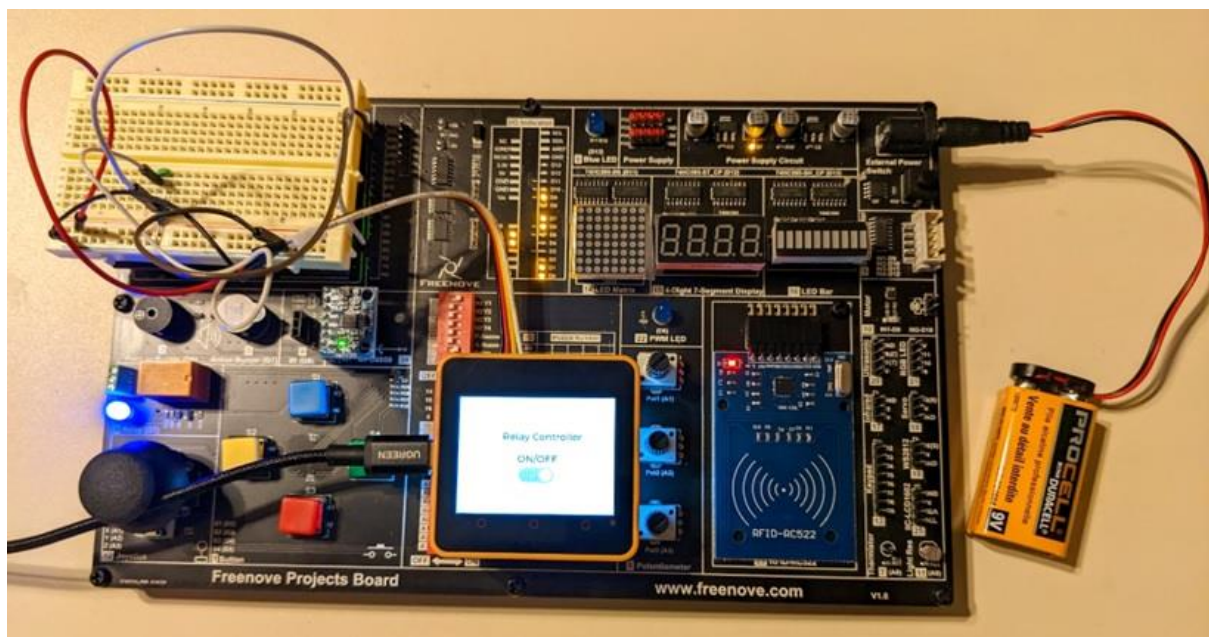


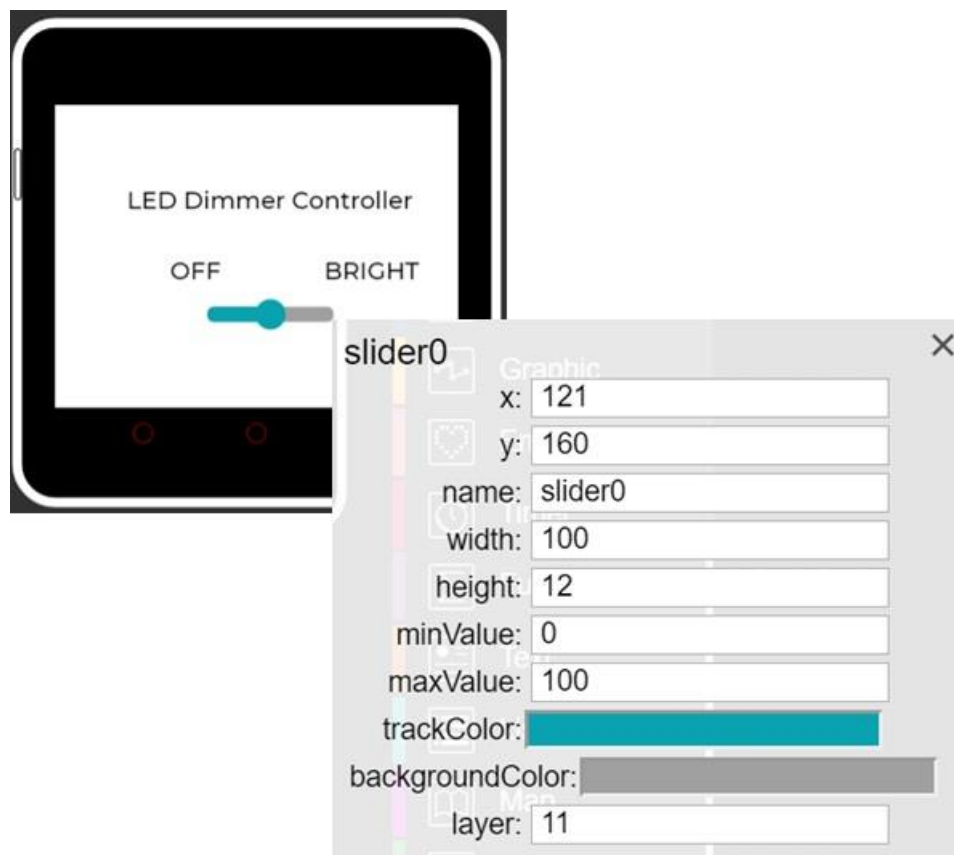


```

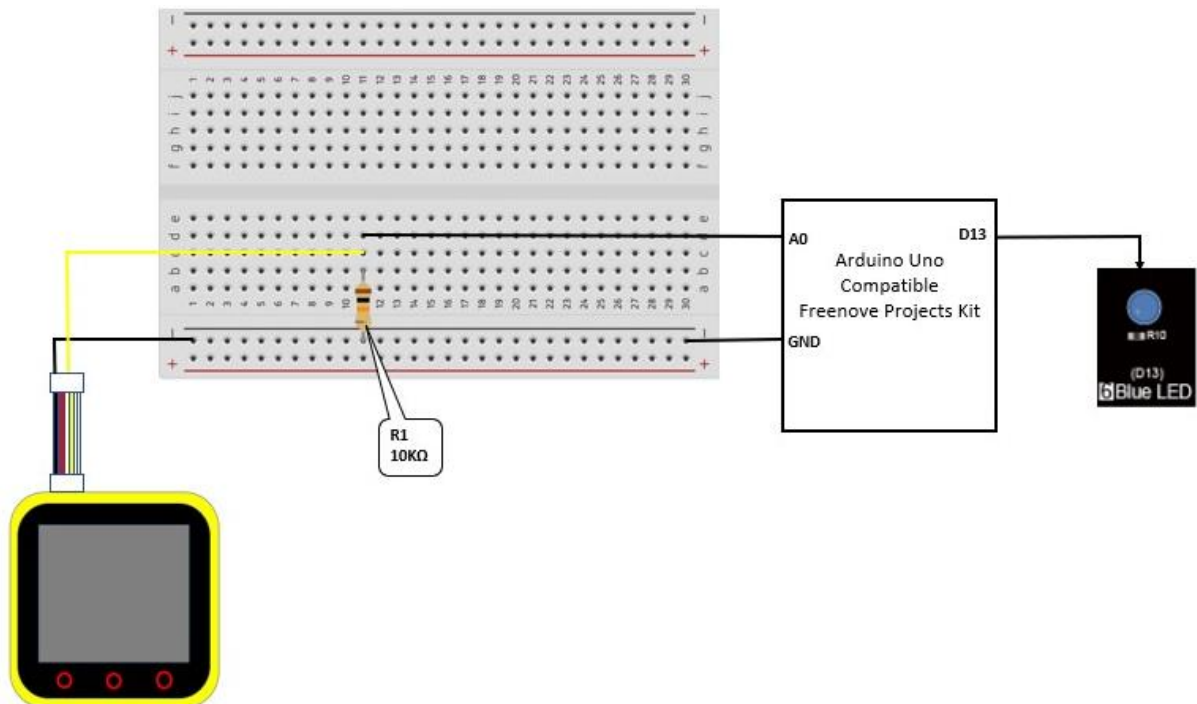
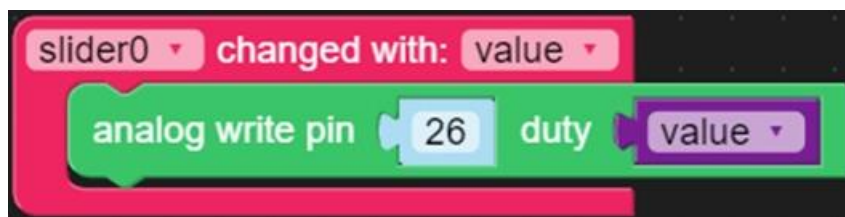
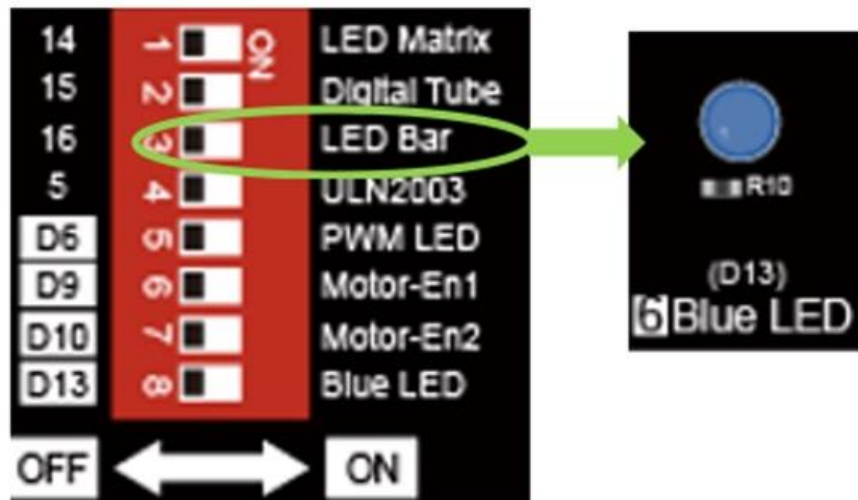
1  int relayPin = 7;    // the number of the relay pin
2  int buttonPin = 2;   // the number of the push button pin
3
4  int buttonState = HIGH;    // Record button state, and initial the state to high level
5  int relayState = LOW;     // Record relay state, and initial the state to low level
6  int lastButtonState = HIGH; // Record the button state of last detection
7  long lastChangeTime = 0;  // Record the time point for button state change
8
9  void setup() {
10     pinMode(buttonPin, INPUT); // Set push button pin into input mode
11     pinMode(relayPin, OUTPUT); // Set relay pin into output mode
12     digitalWrite(relayPin, relayState); // Set the initial state of relay into "off"
13     Serial.begin(9600);          // Initialize serial port, and set baud rate to 9600
14 }
15
16 void loop() {
17     int nowButtonState = digitalRead(buttonPin); // Read current state of button pin
18     // If button pin state has changed, record the time point
19     if (nowButtonState != lastButtonState) {
20         lastChangeTime = millis();
21     }
22     // If button state changes, and stays stable for a while, then it should have skipped the bounce area
23     if (millis() - lastChangeTime > 10) {
24         if (buttonState != nowButtonState) { // Confirm button state has changed
25             buttonState = nowButtonState;
26             if (buttonState == LOW) {        // Low level indicates the button is pressed
27                 relayState = !relayState;    // Reverse relay state
28                 digitalWrite(relayPin, relayState); // Update relay state
29                 Serial.println("Button is Pressed!");
30             }
31             else {                          // High level indicates the button is released
32                 Serial.println("Button is Released!");
33             }
34         }
35     }
36     lastButtonState = nowButtonState; // Save the state of last button
37 }

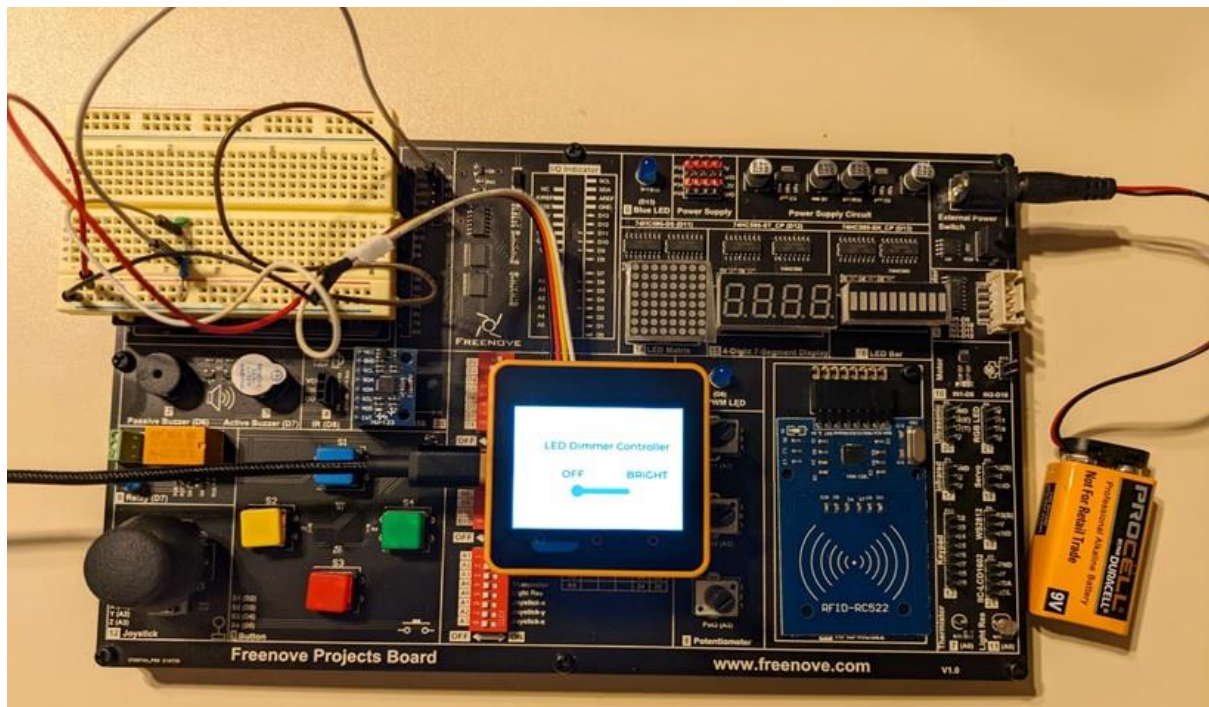
```





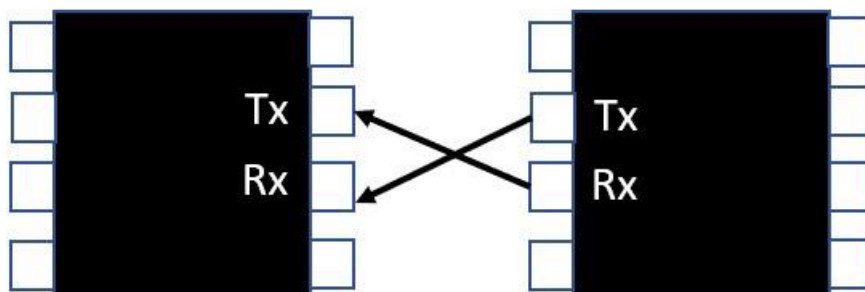
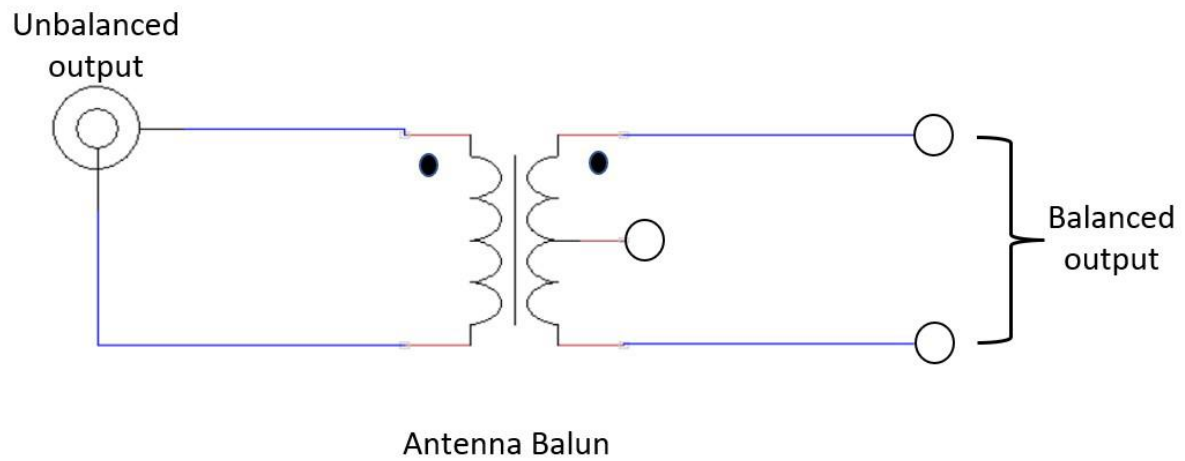
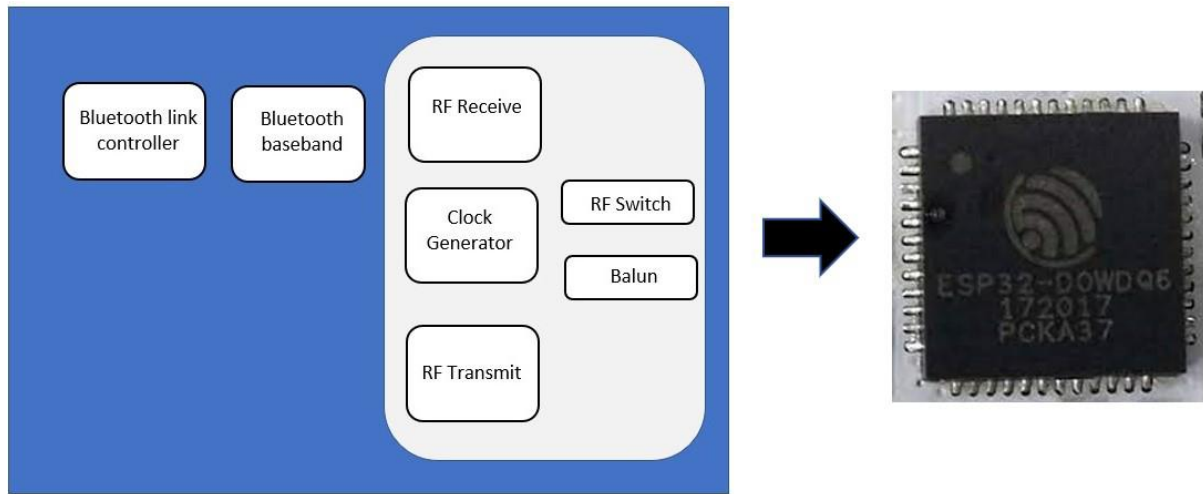
Slide the DIP switch to the ON position





```
1  int adcValue;           // Define a variable to save the ADC value
2  int ledPin = 13;        // Use D9 on Freenove UNO to control the LED
3
4  void setup() {
5      pinMode(ledPin, OUTPUT);           // Initialize the LED pin as an output
6  }
7
8  void loop() {
9      adcValue = analogRead(A1);         // Convert the analog of A0 port to digital
10     // Map analog to the 0-255 range, and works as PWM duty cycle of ledPin port
11     analogWrite(ledPin, map(adcValue, 0, 1023, 0, 255));
12 }
```

Chapter 7: Working with M5Stack and Bluetooth





Execute

EspNow

MQTT

Http

► Socket

Modbus Master

Modbus Slave

CAN

Blynk BLE

BLE UART

BLE UART

Init ble UART name

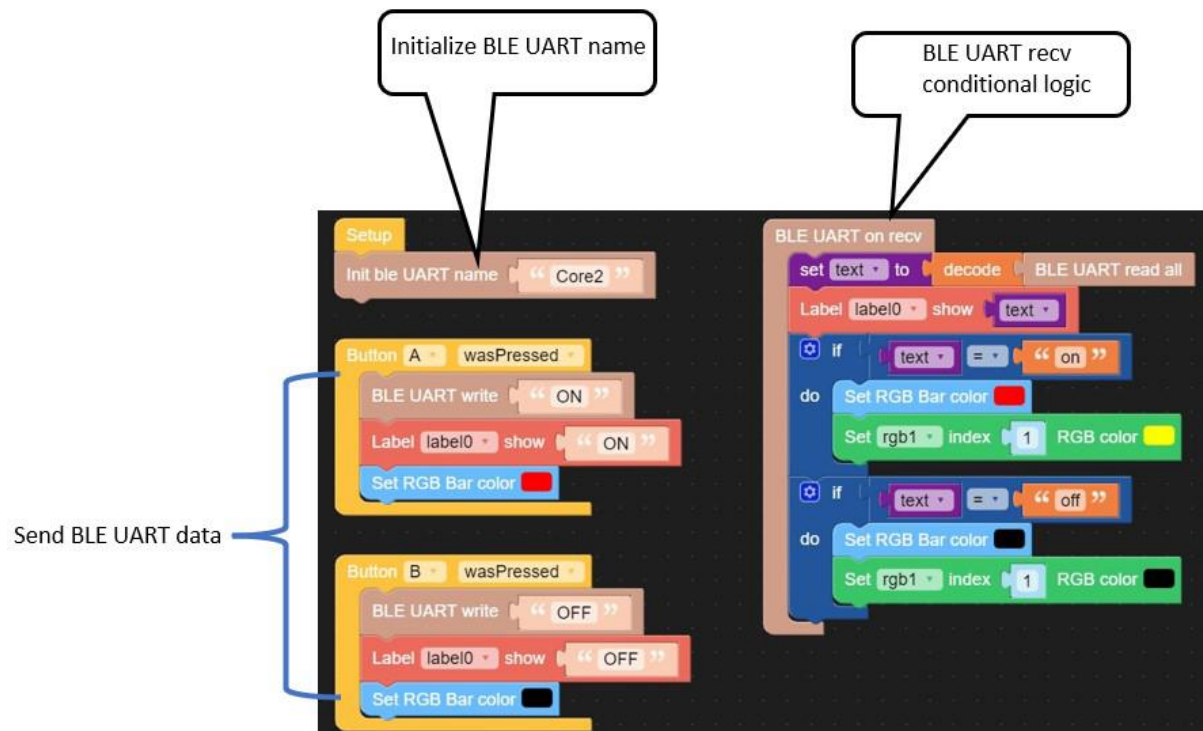
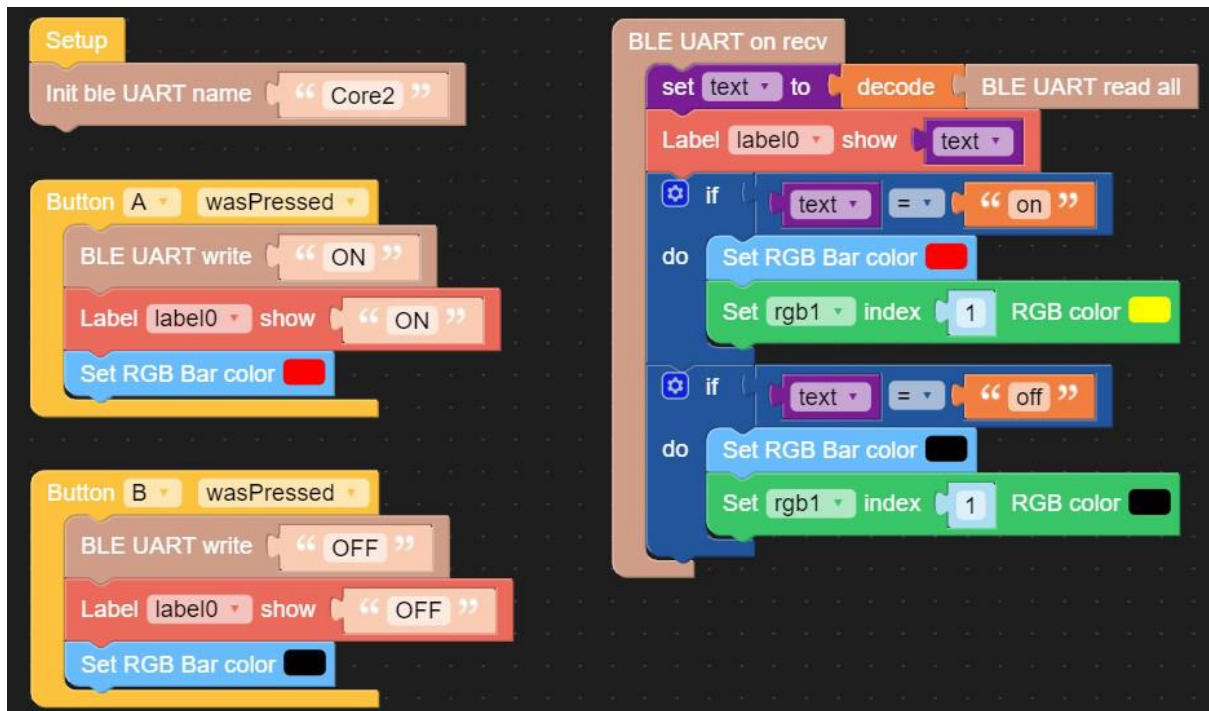
BLE UART write

BLE UART remain cache

BLE UART read all

BLE UART read1 characters

BLE UART on recv





ViewModel profiles



Glucose



Blood Pressure

Service profiles



Cyclic Speed and Cadence



Heart Rate



Health Thermometer

Utils services



Universal Asynchronous
Receiver/Transmitter (UART)

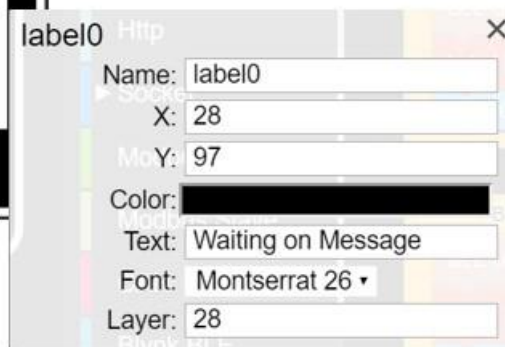


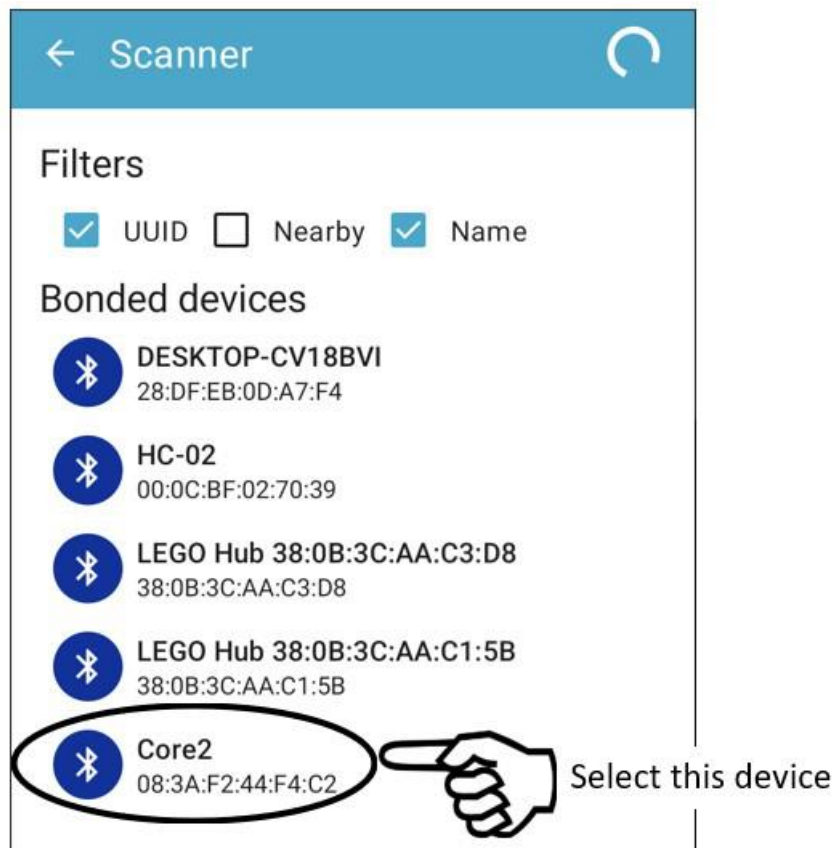
Device Firmware Update
Download from Google Play.



nRF Logger
Download from Google Play.

3.1.3





← Core2

Disconnect



Input

Macros



Output



The incoming messages will be displayed here.

Text to send

Send

Type Text Here



← Core2 Disconnect

Input Macros

Output

The incoming messages will be displayed here.

Text to send

Hello World!!

Send

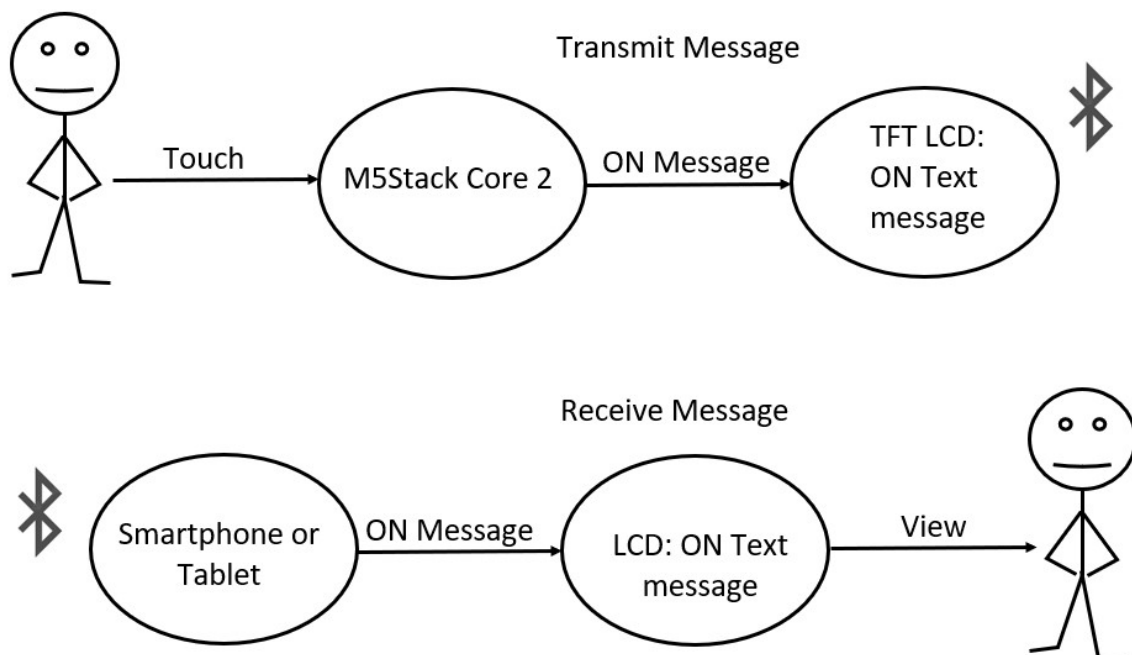
< GIF

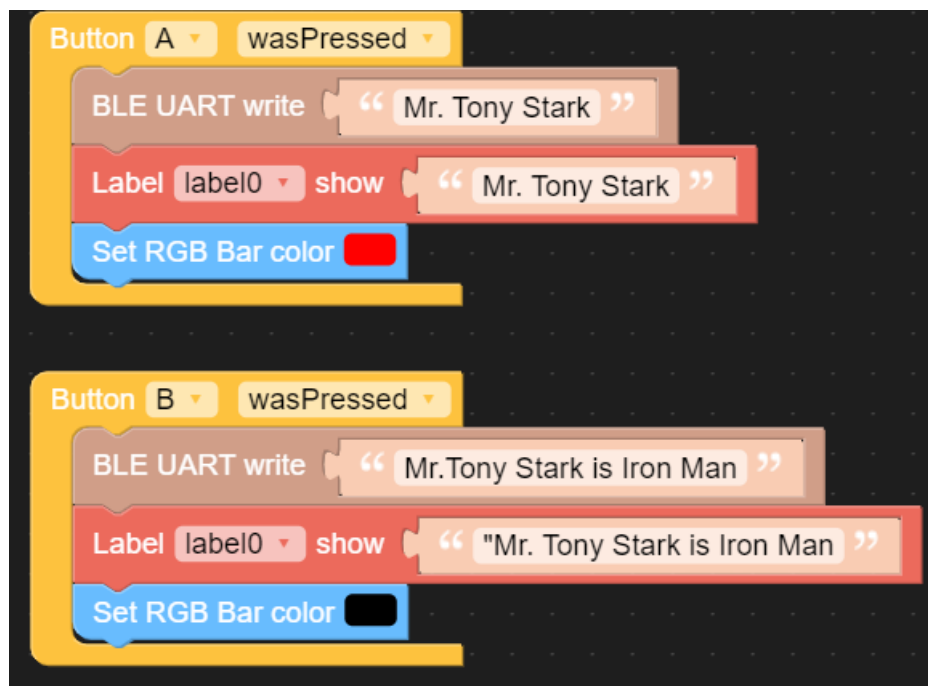
1 2 3 4 5 6 7 8 9 0

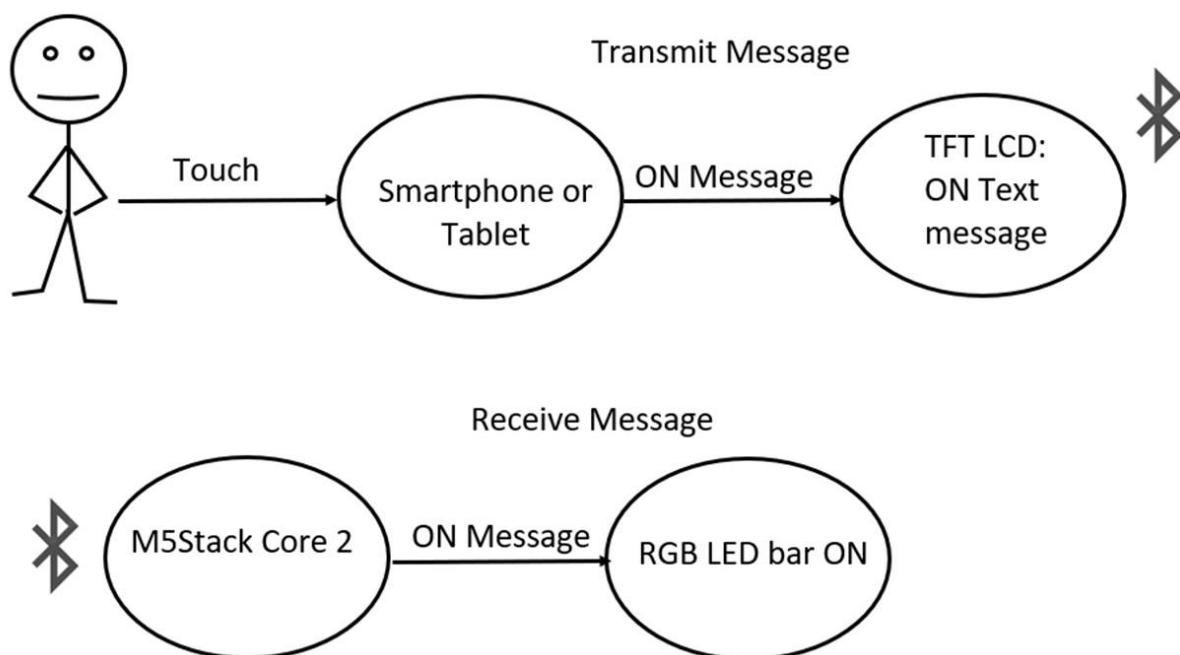
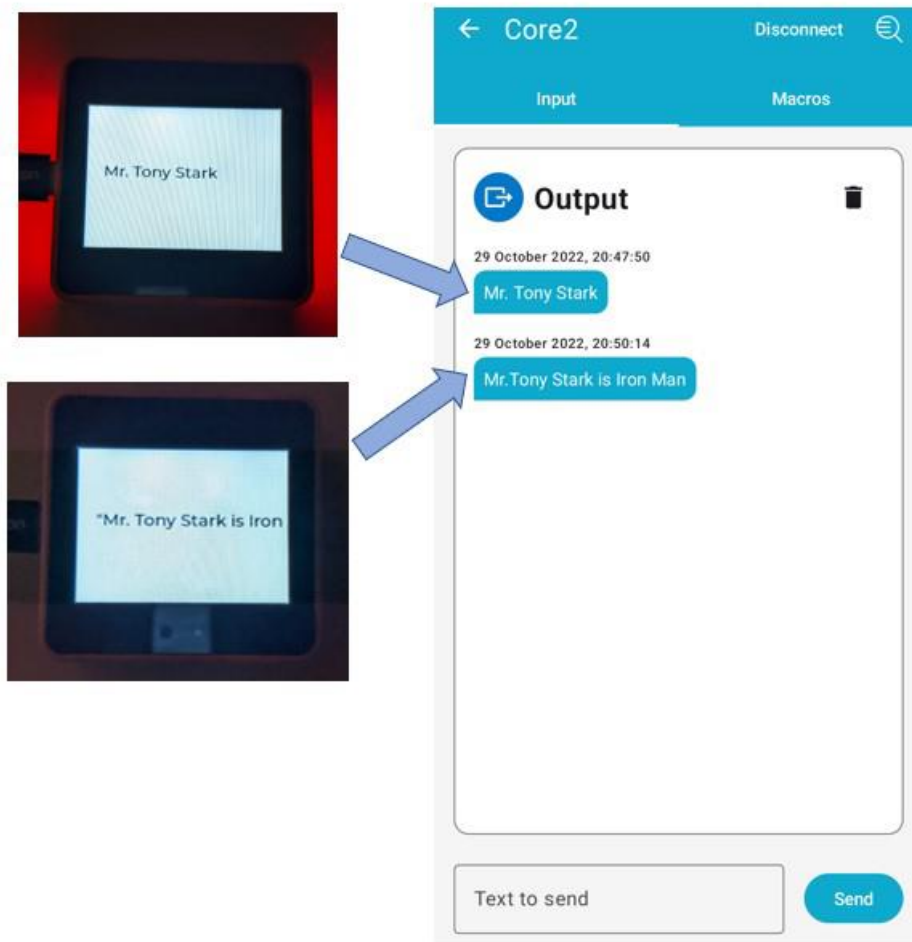
@ # \$ % & * + () /

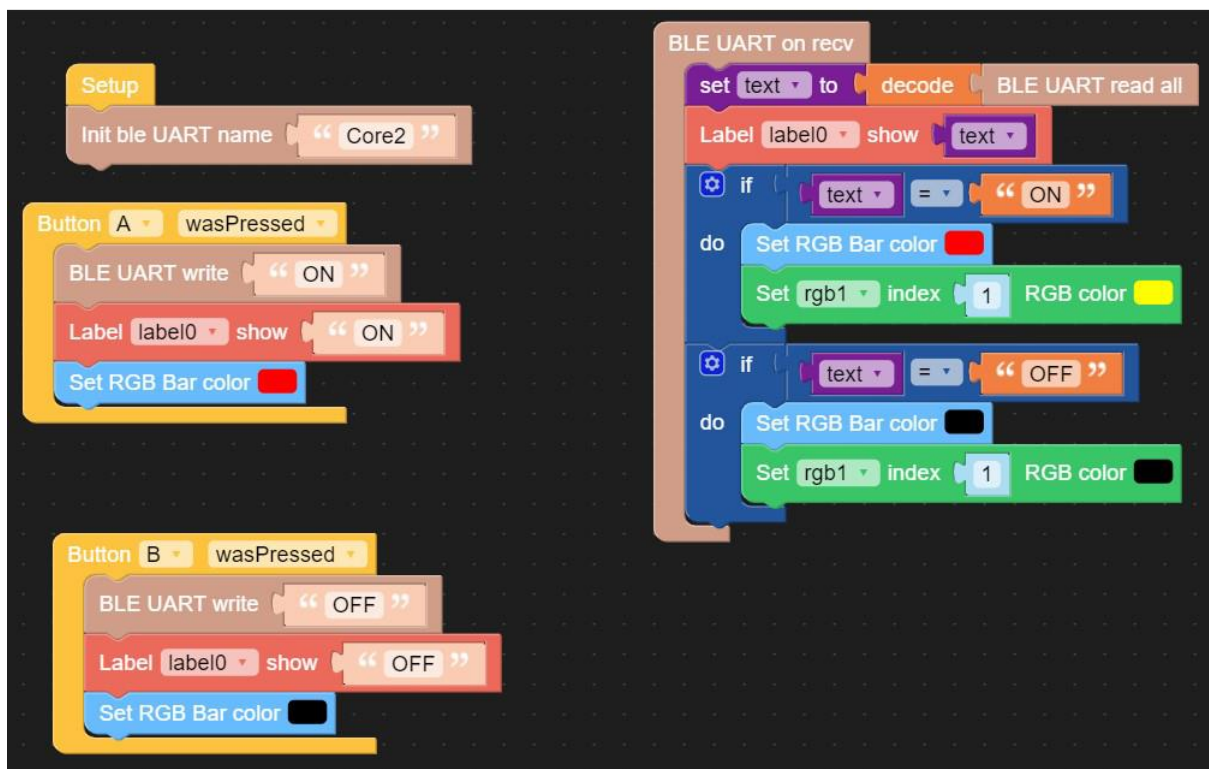
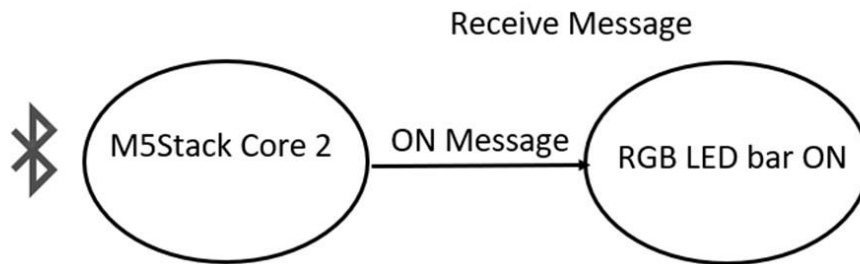
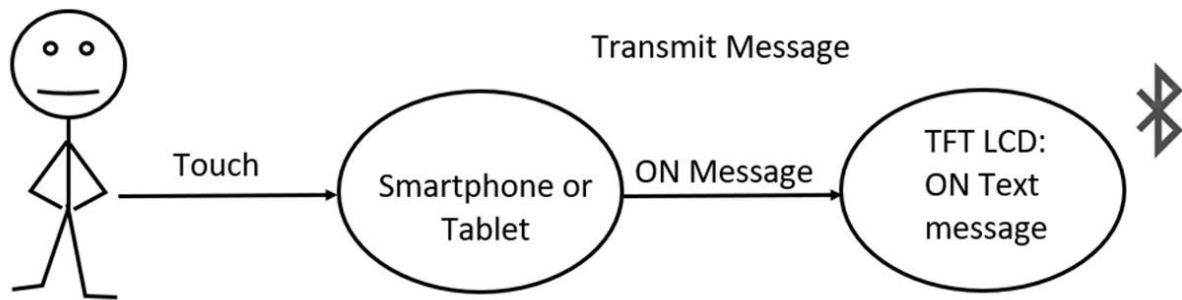
=\< " ' : ; ! ?

ABC , 12 34 .

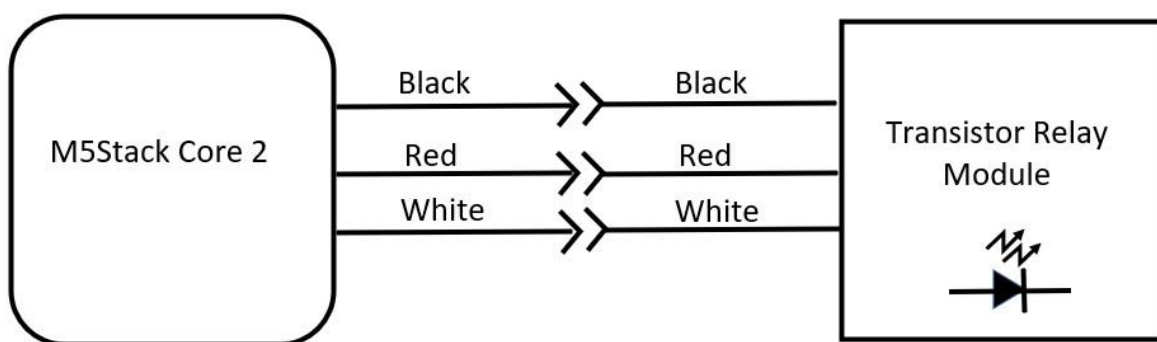


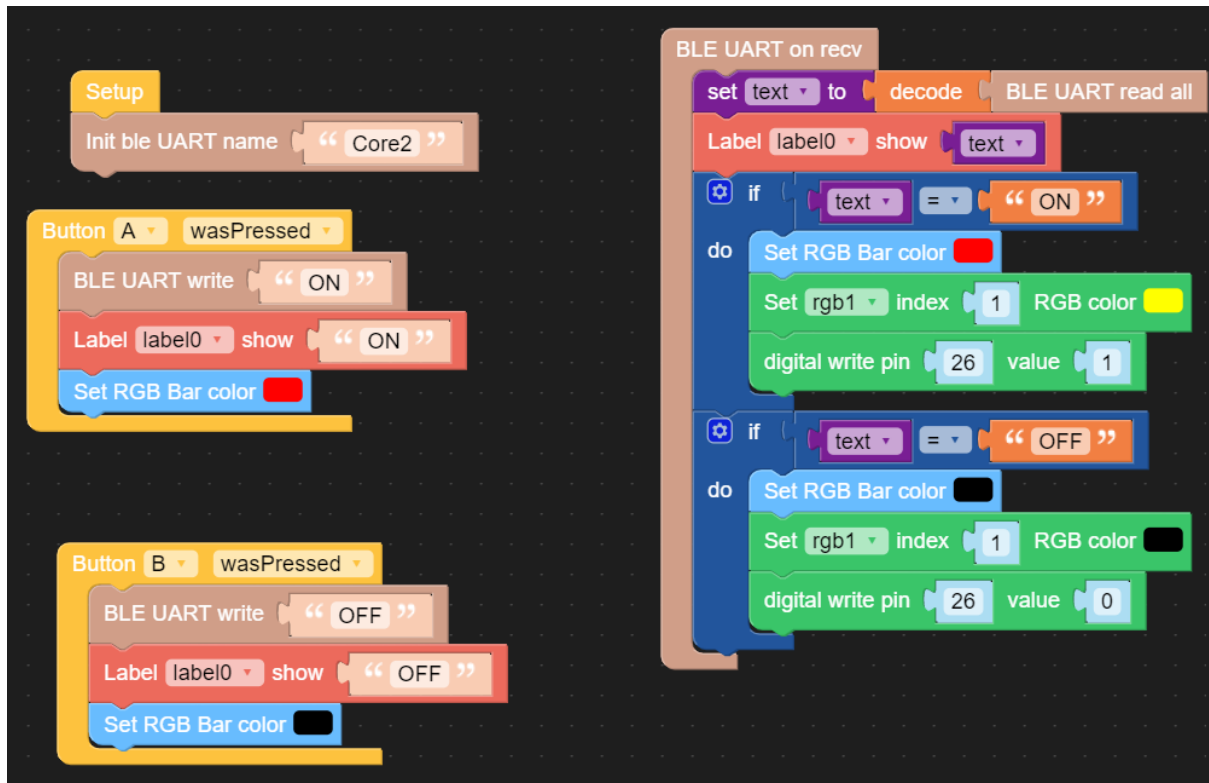
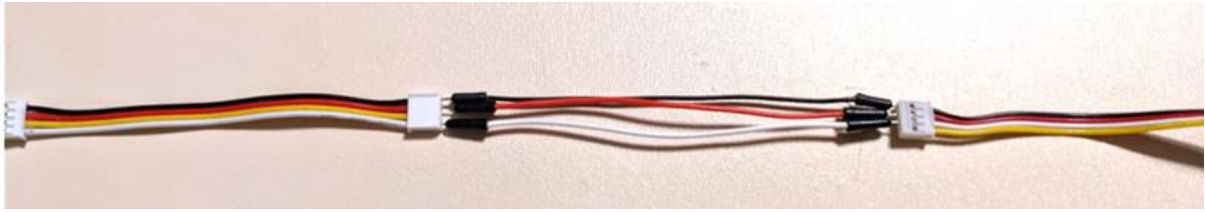


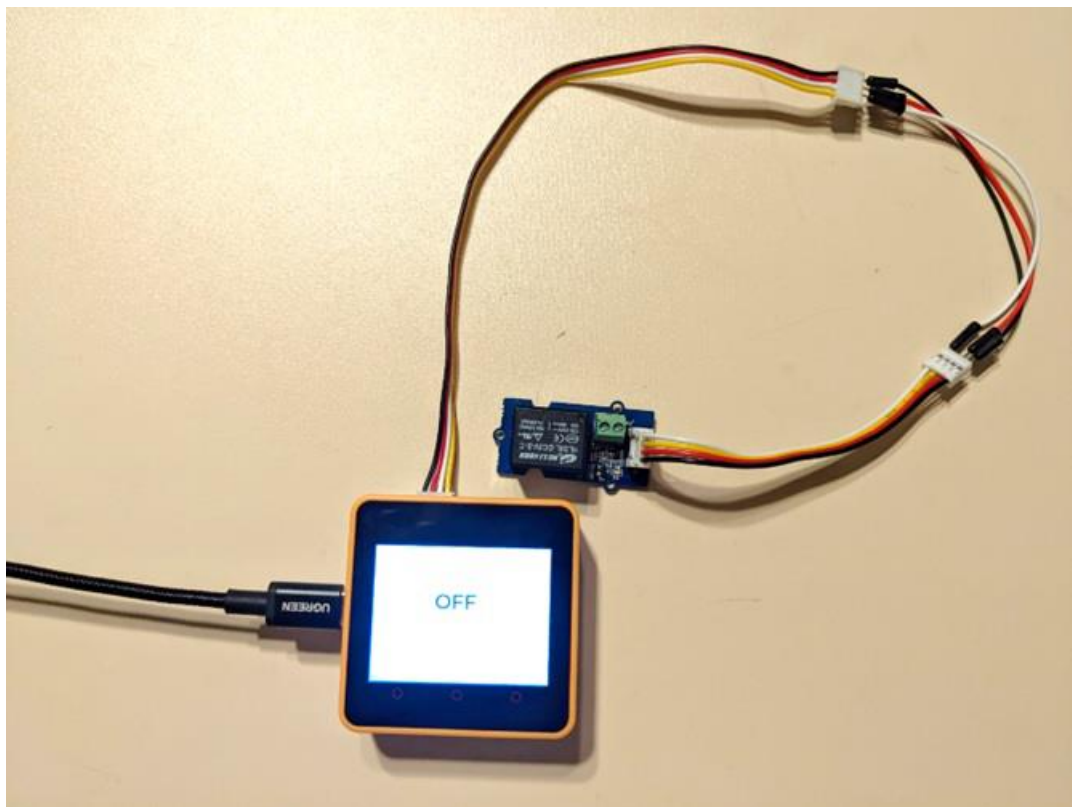
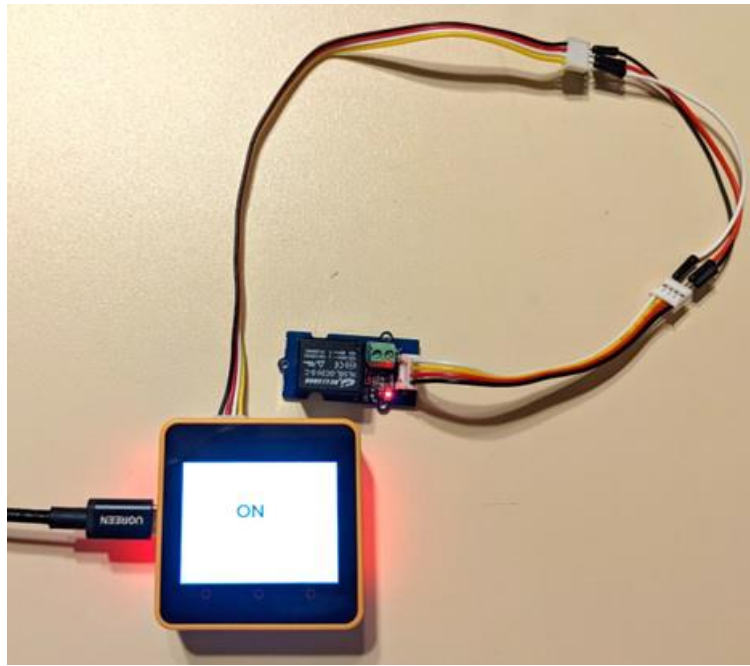


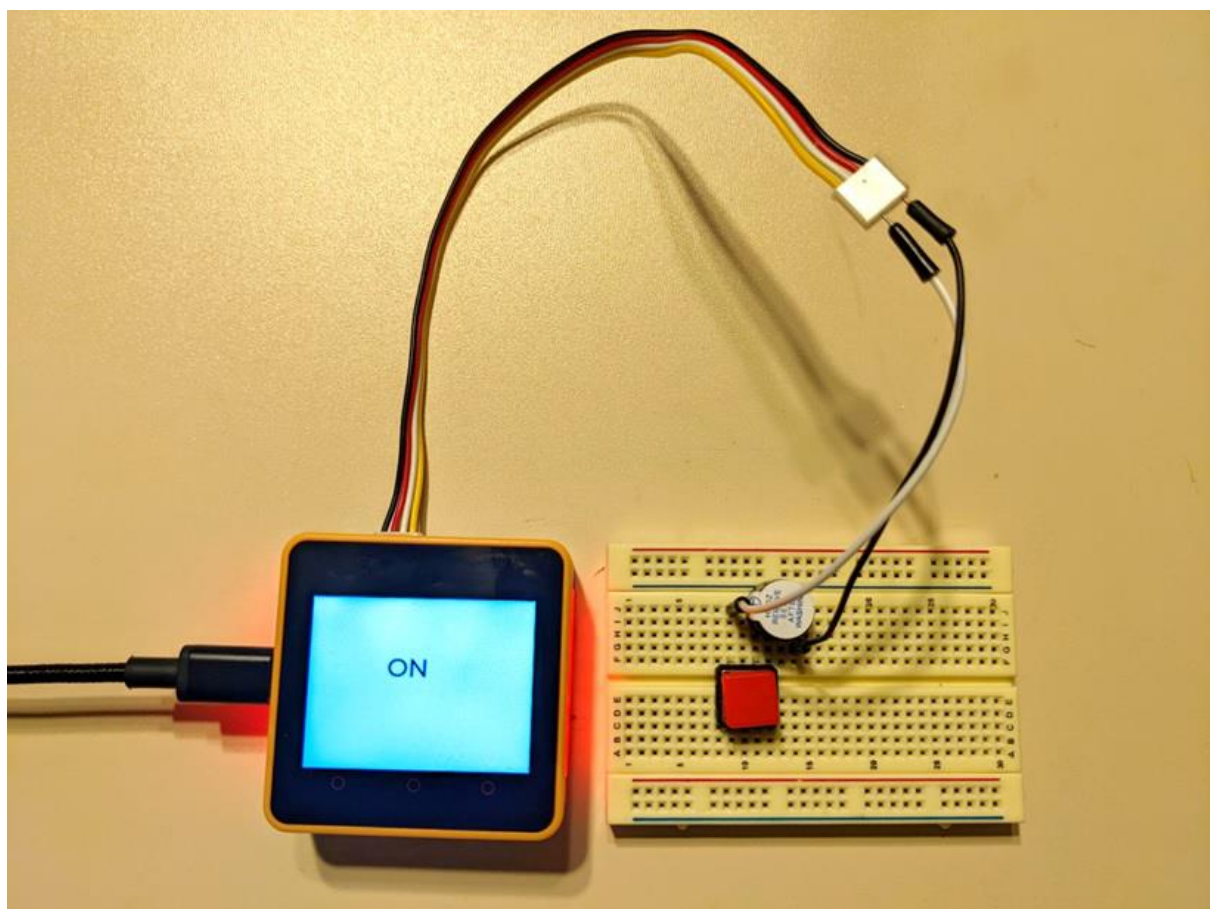
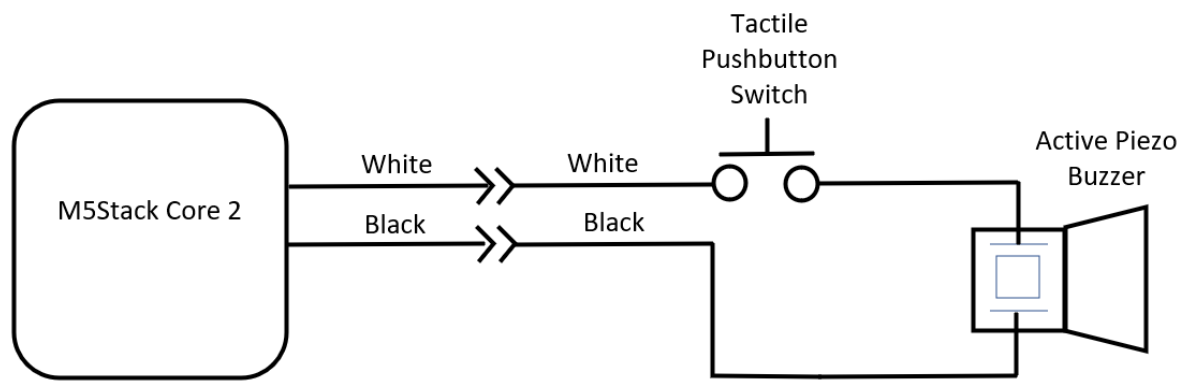


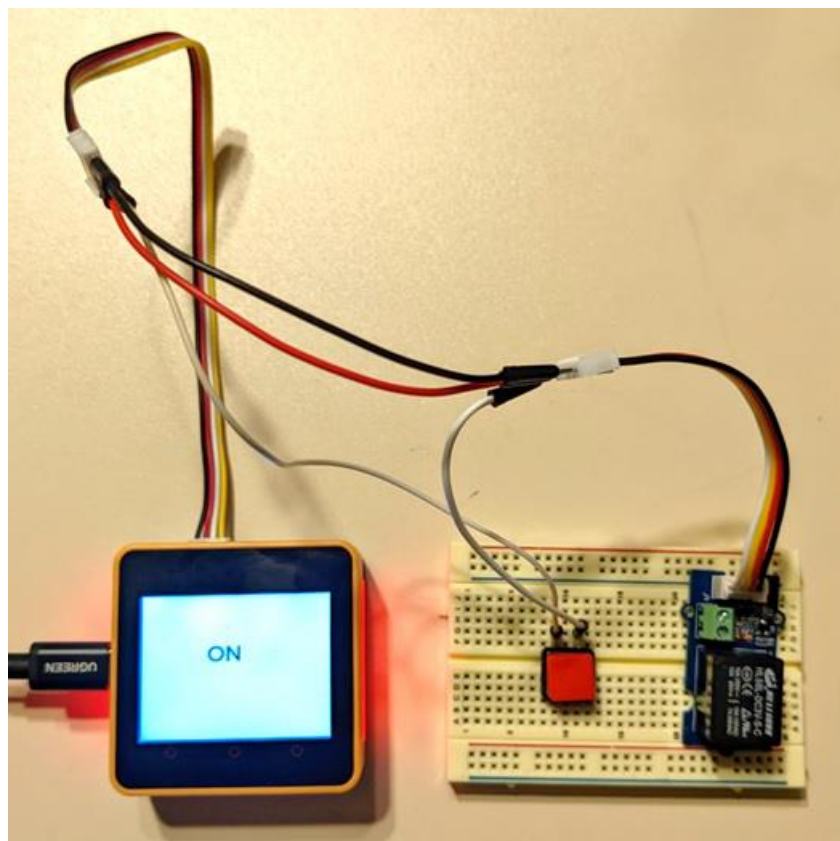
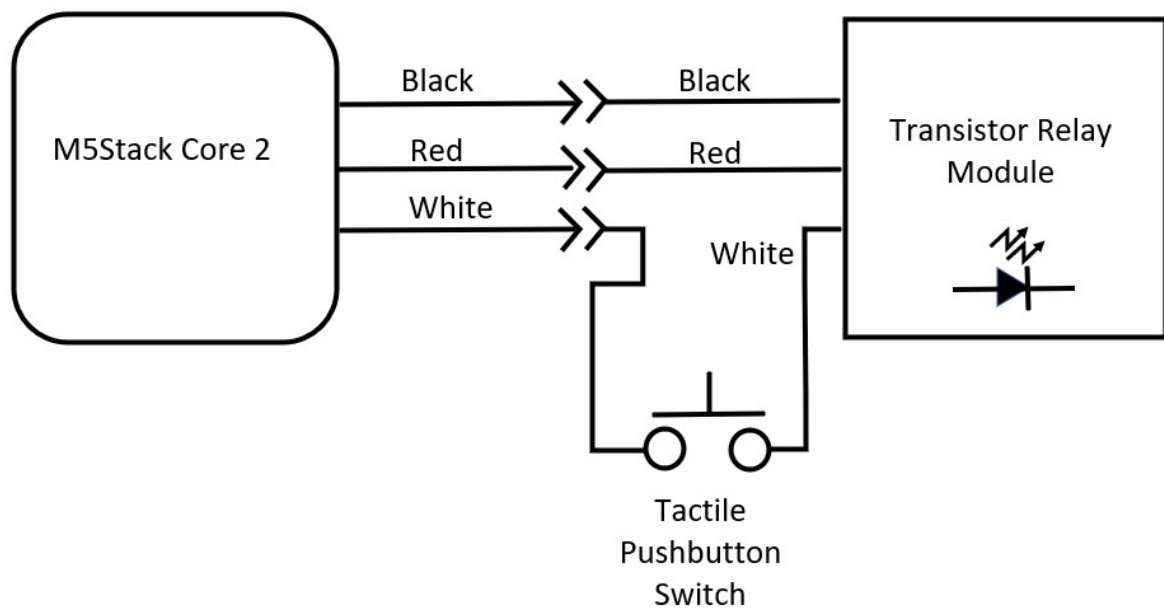






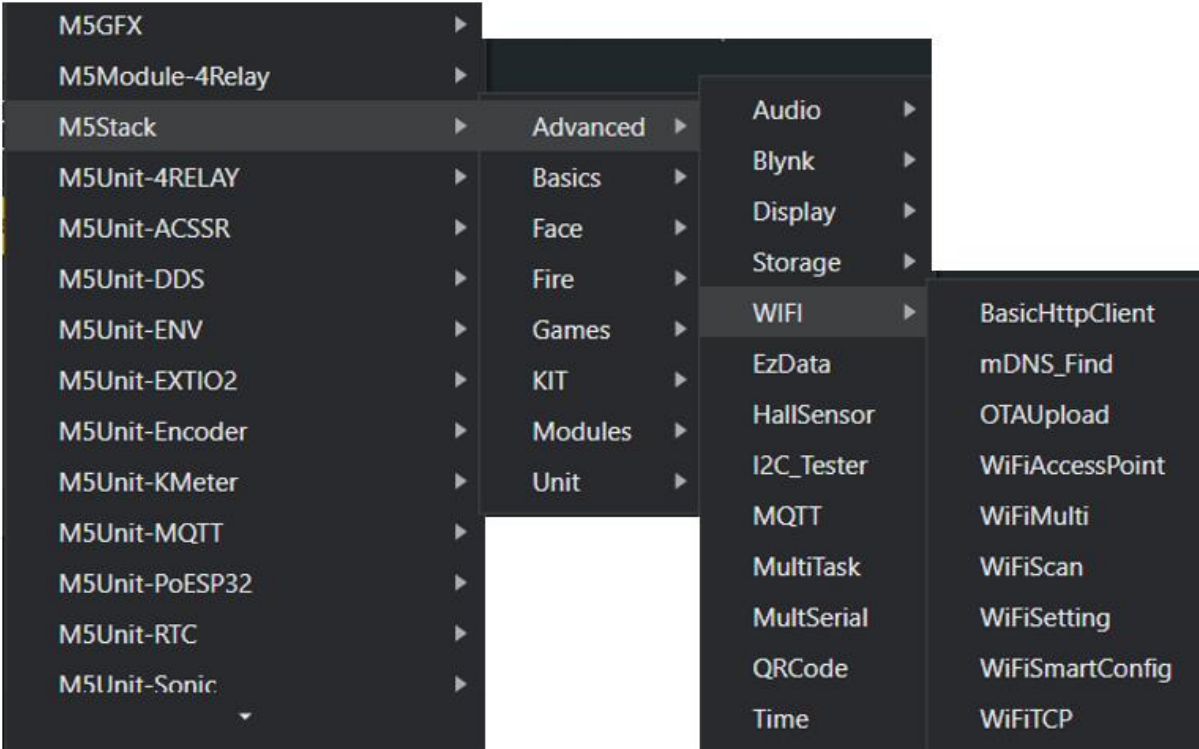
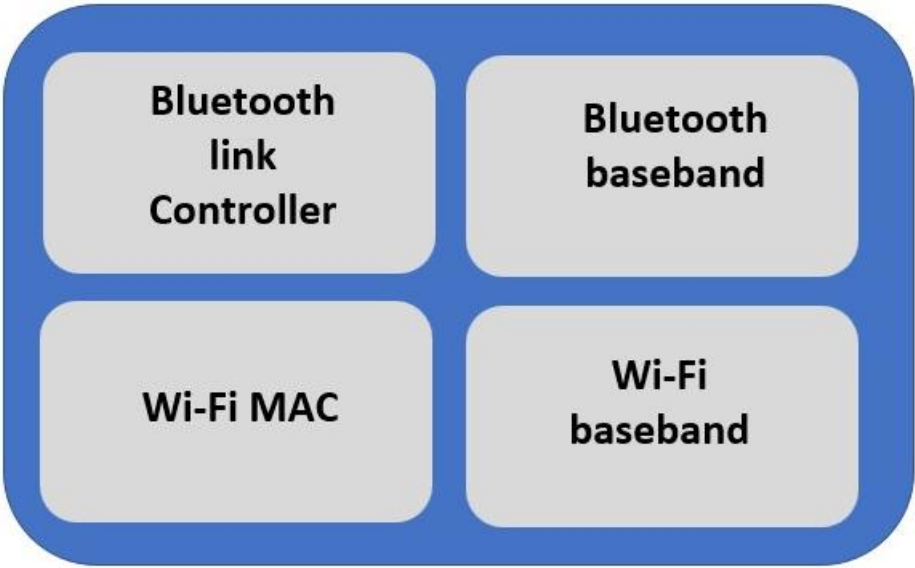






Chapter 8: Working with the M5Stack and Wi-Fi

Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse
MAC: b8:f0:09:c6:16:c4



```

#include <M5Stack.h>

/* After M5Core is started or reset
the program in the setUp () function will be run, and this part will only be run once.
After M5Core is started or reset, it will start to execute the program in the setup() function, and this part will only be
executed once. */
void setup(){
    M5.begin(); //Init M5Core. Initialize M5Core
    M5.Power.begin(); //Init Power module. Initialize the power module
        /* Power chip connected to gpio21, gpio22, I2C device
        Set battery charging voltage and current
        If used battery, please call this function in your project */
    M5.Lcd.print("Hello World"); // Print text on the screen (string) Print text on the screen (string)
}

/* After the program in setup() runs, it runs the program in loop()
The loop() function is an infinite loop in which the program runs repeatedly
After the program in the setup() function is executed, the program in the loop() function will be executed
The loop() function is an endless loop, in which the program will continue to run repeatedly */
void loop() {

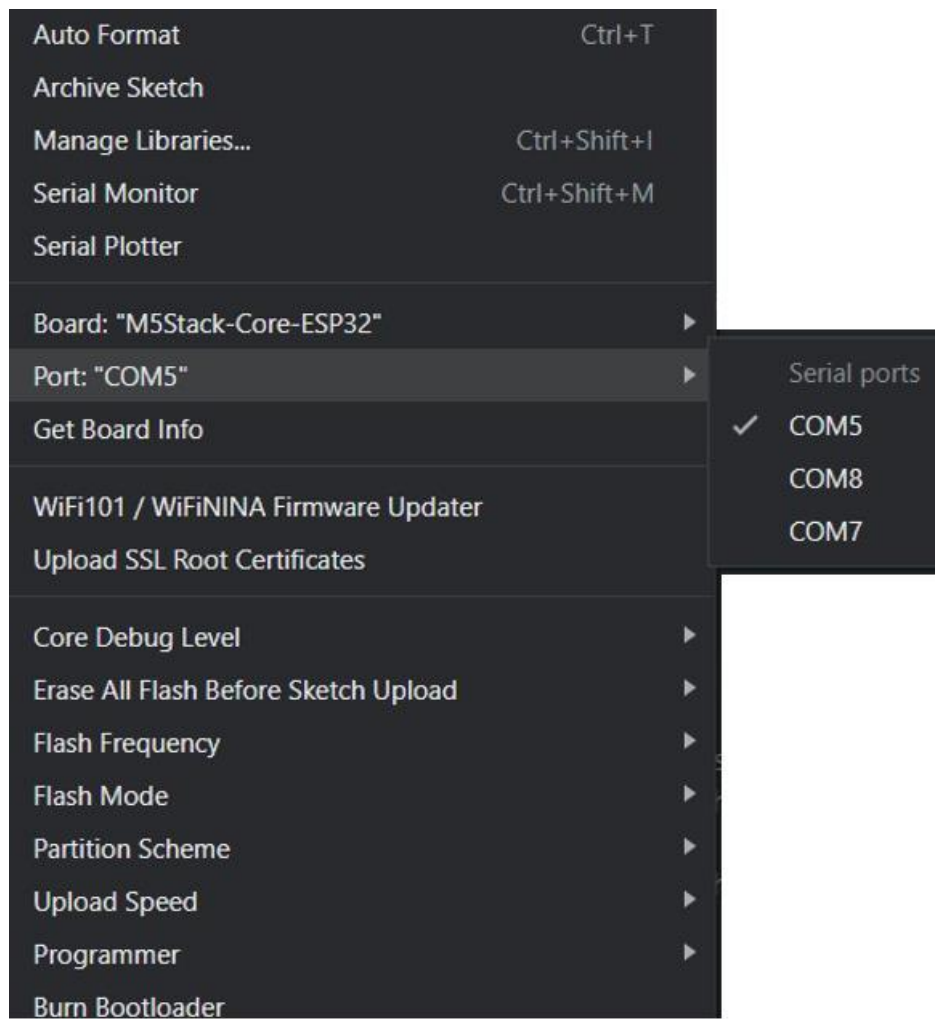
}

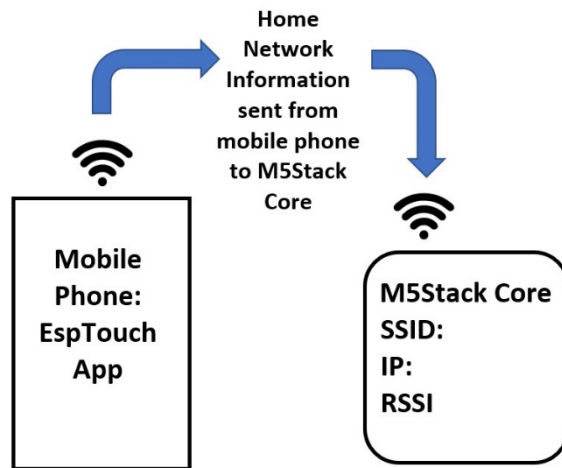
```

```

7 void setup(){
8     M5.begin(); //Init M5Core. Initialize M5Core
9     M5.Power.begin(); //Init Power module. Initialize the power module
10    /* Power chip connected to gpio21, gpio22, I2C device
11    Set battery charging voltage and current
12    If used battery, please call this function in your project */
13    M5.Lcd.setTextSize(2);
14    M5.Lcd.print("Hello World"); // Print text on the screen (string) Print text on the screen (string)
15 }

```





Home

EspTouch

EspTouch V2

← EspTouch ⓘ

SSID:

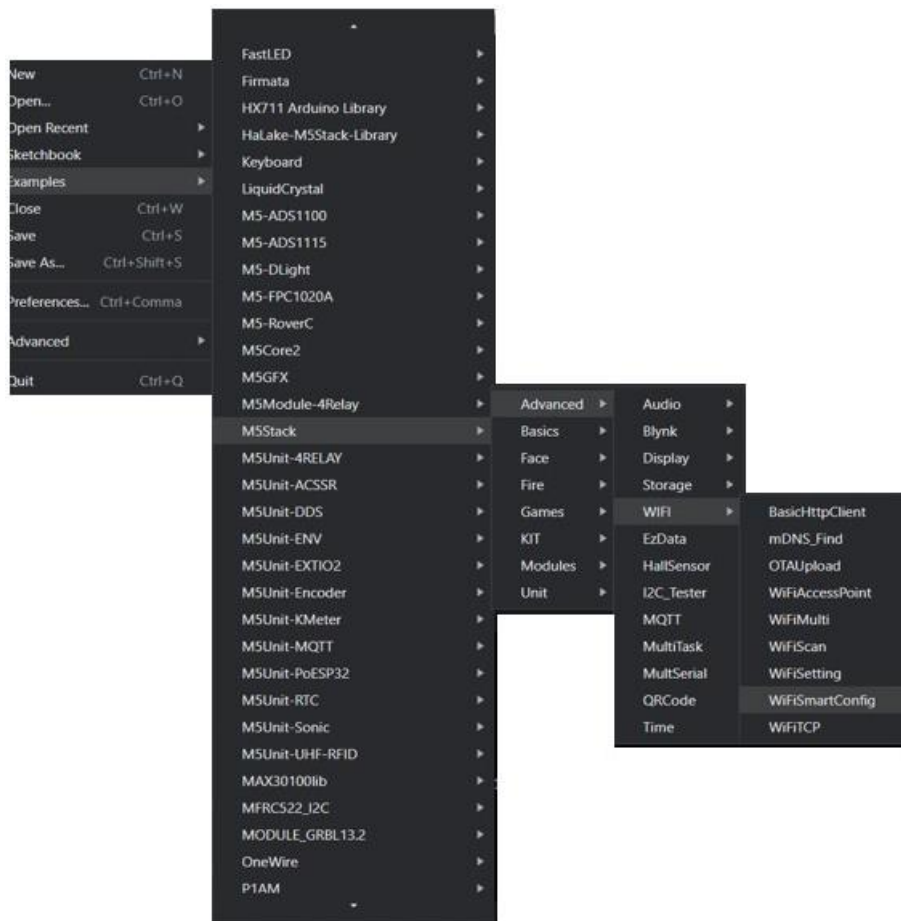
BSSID:

Wi-Fi Password:

Device count:
1

☒ Broadcast ☐ Multicast

START



```

18 #include <M5Stack.h>
19
20 #include "WiFi.h"
21
22 void setup() {
23     M5.begin();           // Init M5Core. 初始化 M5Core
24     M5.Power.begin();     // Init power 初始化电源模块
25     WiFi.mode(WIFI_AP_STA); // Set the wifi mode to the mode compatible with
26                             // the AP and Station, and start intelligent
27                             // network configuration
28     WiFi.beginSmartConfig(); // 设置wifi模式为AP 与 station
29                             // 兼容模式,并开始智能配网
30
31     // Wait for the M5Core to receive network information from the phone
32     // 等待M5Core接收到来自手机的配网信息
33     M5.Lcd.println(
34         "\nWaiting for Phone SmartConfig."); // Screen print format string.
35                                             // 屏幕打印格式化字符串
36     while (!WiFi.smartConfigDone()) { // If the smart network is not completed.
37                                     // 若智能配网没有完成
38         delay(500);
39         M5.Lcd.setTextSize(2);

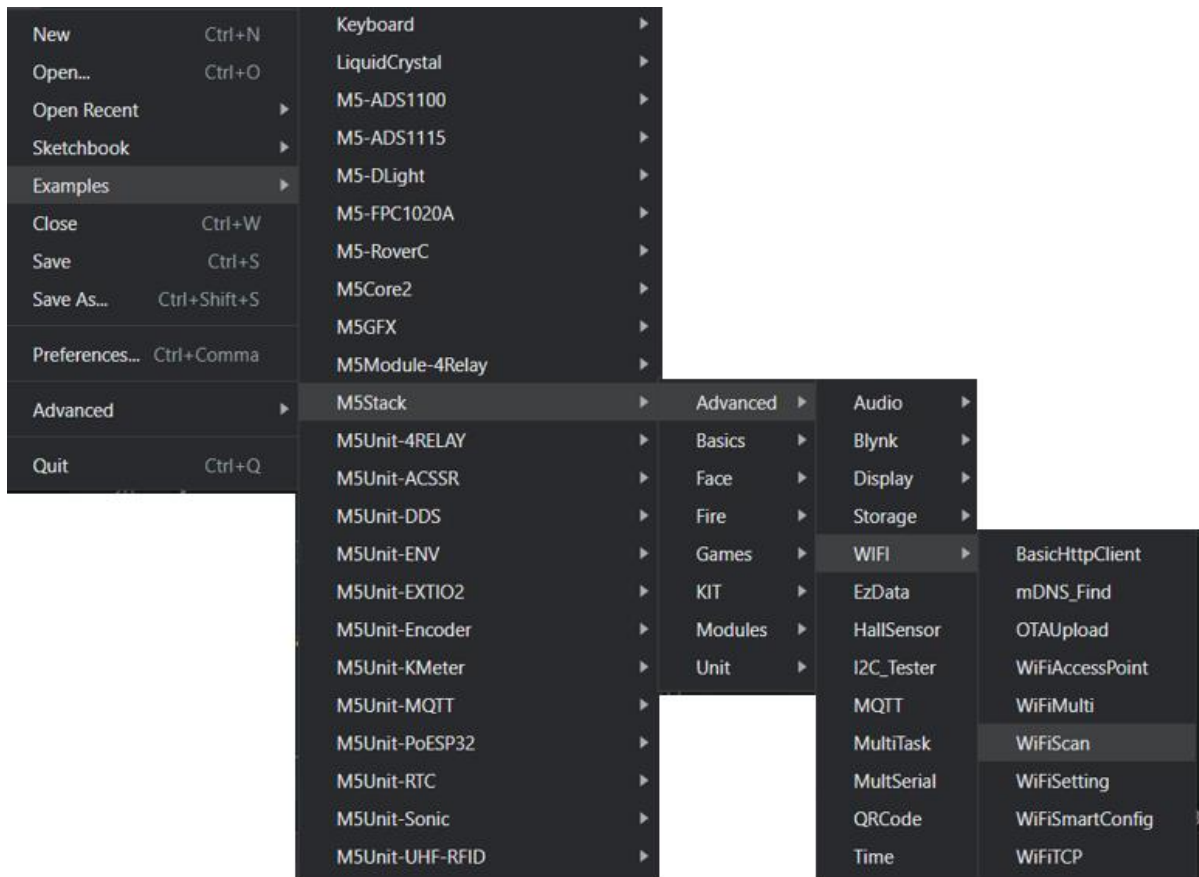
```

```

38     delay(500);
39     M5.Lcd.setTextSize(2);
40     M5.Lcd.print(".");
41 }
42 M5.Lcd.setTextSize(2);
43 M5.Lcd.println("\nSmartConfig received.");
44 M5.Lcd.setTextSize(2);
45 M5.Lcd.println("Waiting for WiFi");
46 while (
47     WiFi.status() !=
48     WL_CONNECTED) { // M5Core will connect automatically upon receipt of
49                     // the configuration information, and return true if
50                     // the connection is successful.
51                     // 收到配网信息后M5Core将自动连接, 若连接成功将返回true
52     delay(500);
53     M5.Lcd.setTextSize(2);
54     M5.Lcd.print(".");
55 }

```





```

15  #include "WiFi.h"
16
17  void setup() {
18      M5.begin();           // Init M5Stack. 初始化M5Stack
19      M5.Power.begin();     // Init power 初始化电源模块
20      WiFi.mode(WIFI_STA); // Set WiFi to station mode and disconnect from an AP
21                          // if it was previously connected.
22                          // 将WiFi设置为站模式, 如果之前连接过AP, 则断开连接
23      WiFi.disconnect();   // Turn off all wifi connections. 关闭所有wifi连接
24      delay(100);          // 100 ms delay. 延迟100ms
25      M5.Lcd.setTextSize(2);
26      M5.Lcd.print("WIFI SCAN"); // Screen print string. 屏幕打印字符串
27  }
28
29  void loop() {
30      M5.Lcd.setCursor(0, 0); // Set the cursor at (0,0). 将光标设置在(0,0)处
31      M5.Lcd.println("Please press Btn.A to (re)scan");
32      M5.update();           // Check the status of the key. 检测按键的状态
33      if (M5.BtnA.isPressed()) { // If button A is pressed. 如果按键A按下
34          M5.Lcd.clear();      // Clear the screen. 清空屏幕
35          M5.Lcd.setTextSize(2);
36          M5.Lcd.println("scan start");

```



| UIFLOW FIRMWARE BURNING TOOL

NO	Name	Download
1	M5Burner Win10 x64 v3.0	↓
2	M5Burner MacOS x64 v3.0	↓
3	M5Burner Linux x64 v3.0	↓

UIFLOW(CORE)

The graphic coding IDE for everyone

UIFlow
official

v1.7.5-en ▼

Developed based on Micropython-1.12, For BASIC/GRAY/M5GO device.

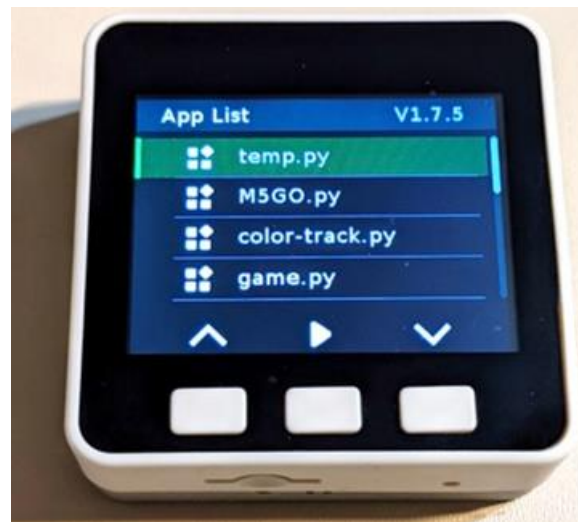
Remove

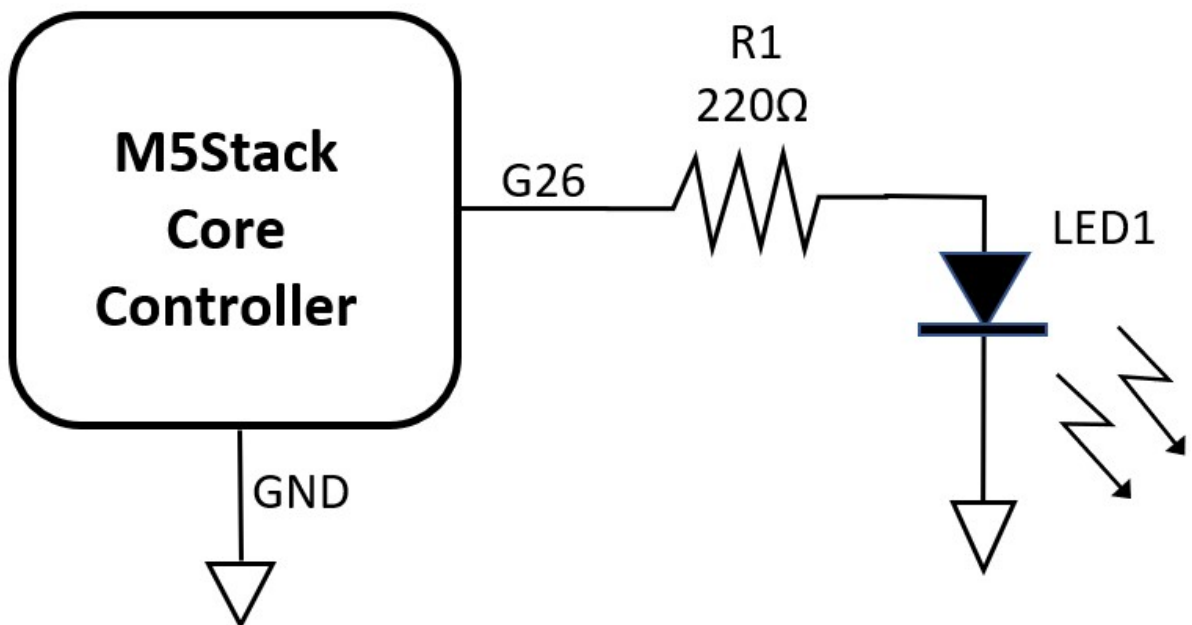
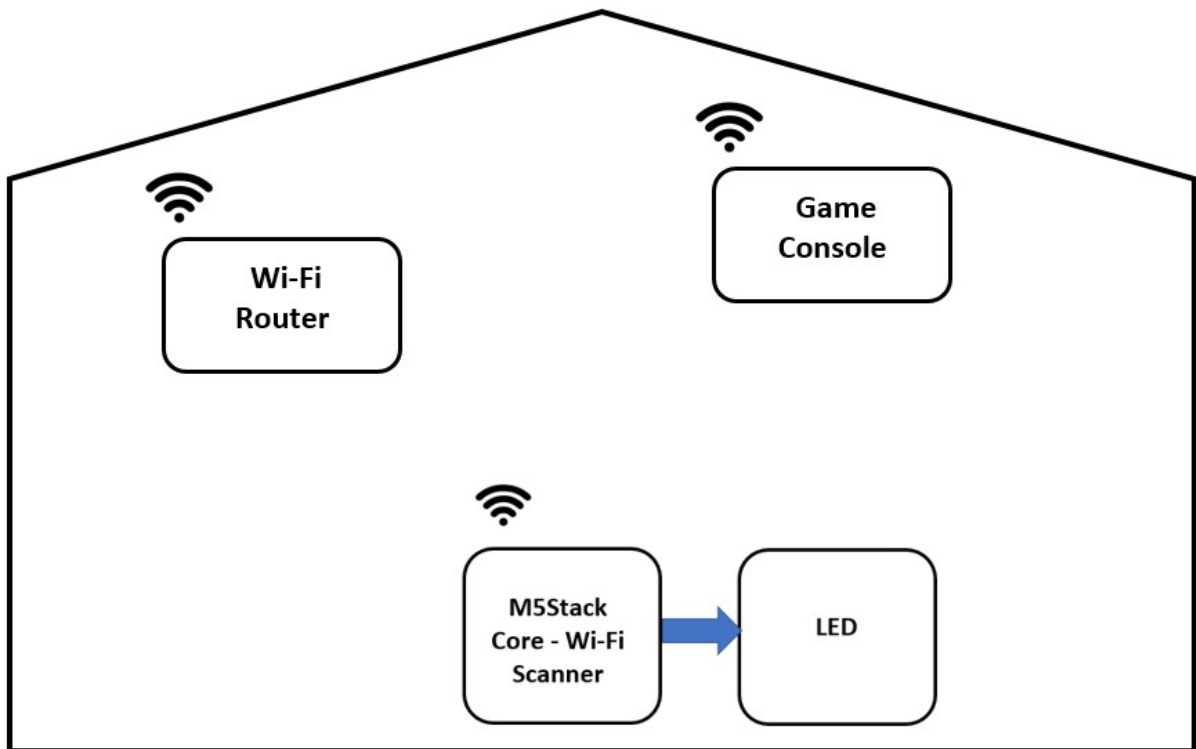
Configuration

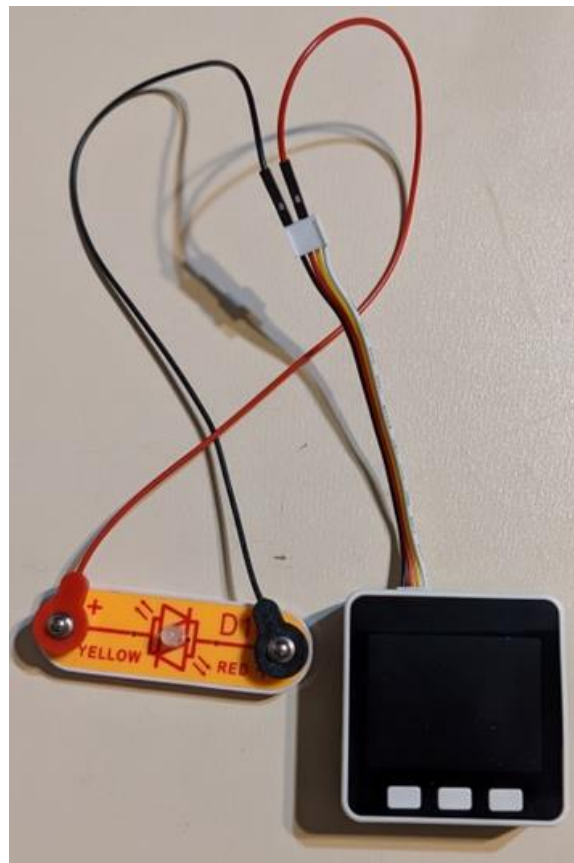
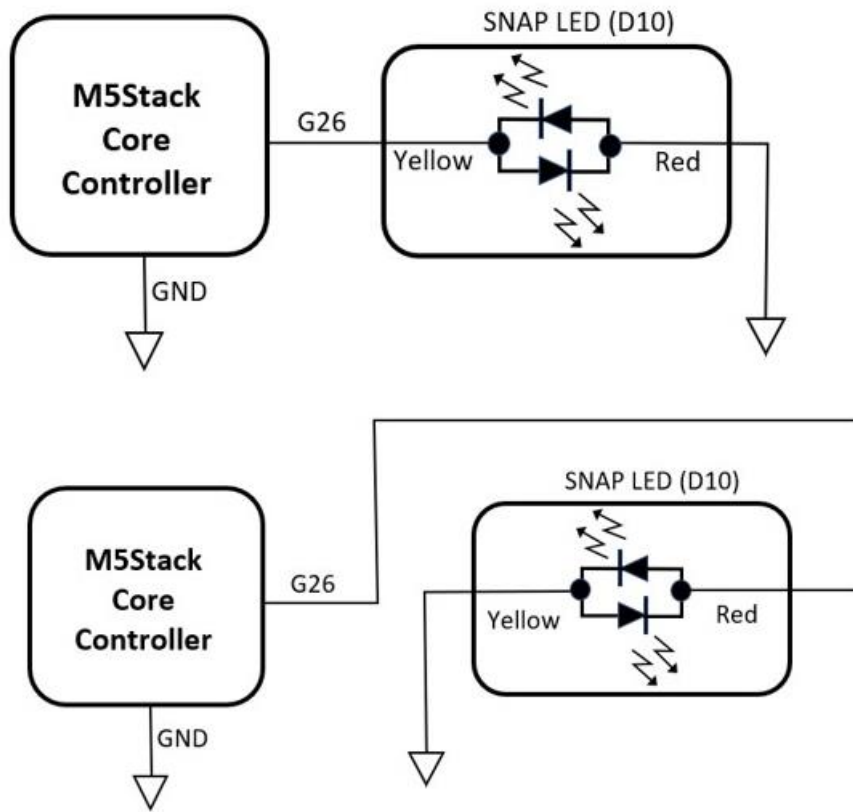
Burn

Author: M5Stack

46.4k







```

17 void setup() {
18     M5.begin();           // Init M5Stack. 初始化M5Stack
19     M5.Power.begin();     // Init power 初始化电源模块
20     WiFi.mode(WIFI_STA); // Set WiFi to station mode and disconnect from an AP
21                           // if it was previously connected.
22                           // 将WiFi设置为站模式, 如果之前连接过AP, 则断开连接
23     WiFi.disconnect();   // Turn off all wifi connections. 关闭所有wifi连接
24     delay(100);          // 100 ms delay. 延迟100ms
25     M5.Lcd.setTextSize(2);
26     M5.Lcd.print("WIFI SCAN"); // Screen print string. 屏幕打印字符串
27     pinMode(26, OUTPUT);
28 }

```

```

51     M5.Lcd.setTextSize(2);
52     M5.Lcd.printf("%d:", i + 1);
53     M5.Lcd.setTextSize(2);
54     M5.Lcd.print(WiFi.SSID(i));
55     M5.Lcd.setTextSize(2);
56     M5.Lcd.printf("(%d)", WiFi.RSSI(i));
57     M5.Lcd.setTextSize(2);
58     M5.Lcd.println(
59         (WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " " : "*");
60     digitalWrite(26, 1);

```

```

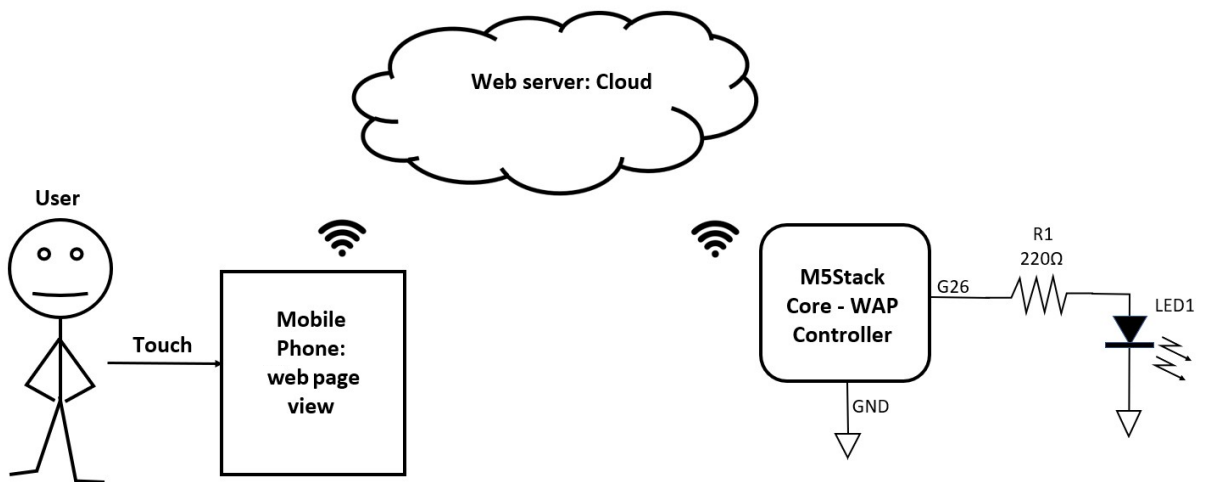
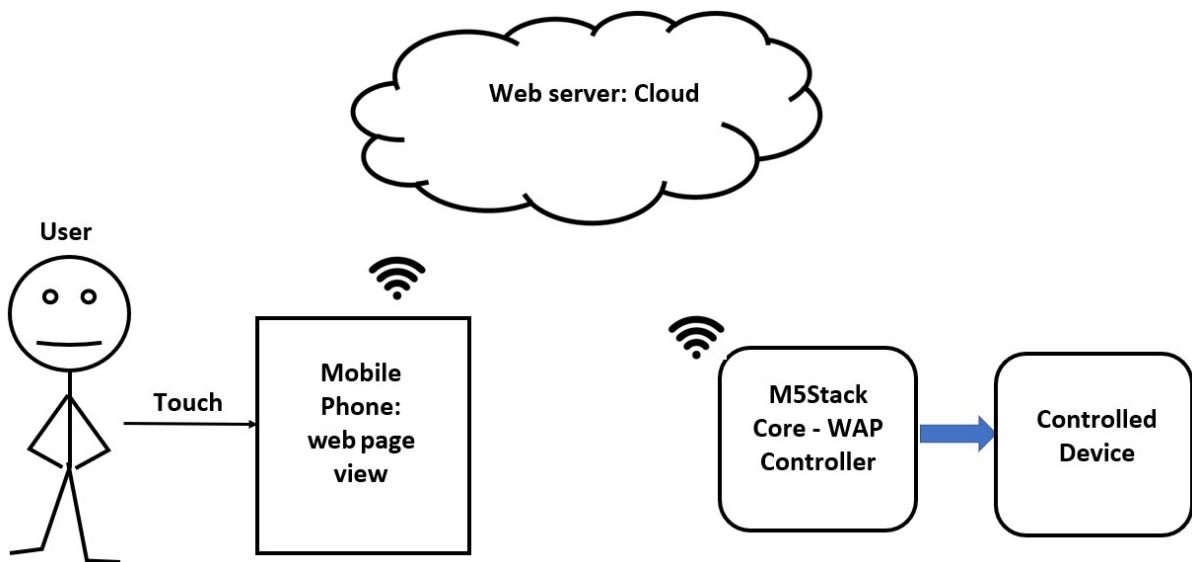
30 void loop() {
31     M5.Lcd.setCursor(0, 0); // Set the cursor at (0,0). 将光标设置在(0,0)处
32     M5.Lcd.println("Please press Btn.A to (re)scan");
33     M5.update();           // Check the status of the key. 检测按键的状态
34     if (M5.BtnA.isPressed()) { // If button A is pressed. 如果按键A按下
35         M5.Lcd.clear();      // Clear the screen. 清空屏幕
36         M5.Lcd.setTextSize(2);
37         M5.Lcd.println("scan start");
38         digitalWrite(26, 0);

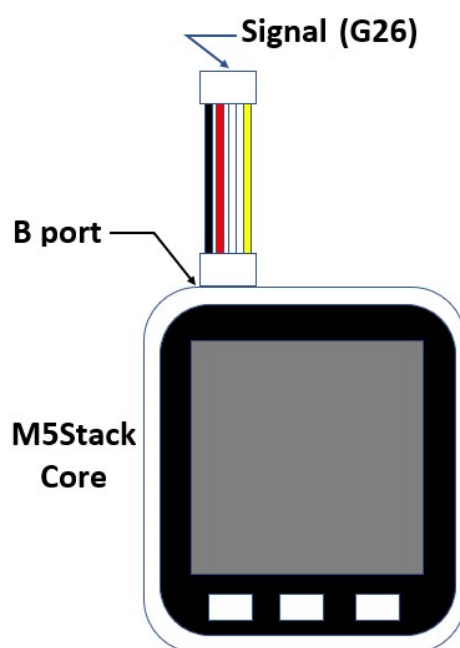
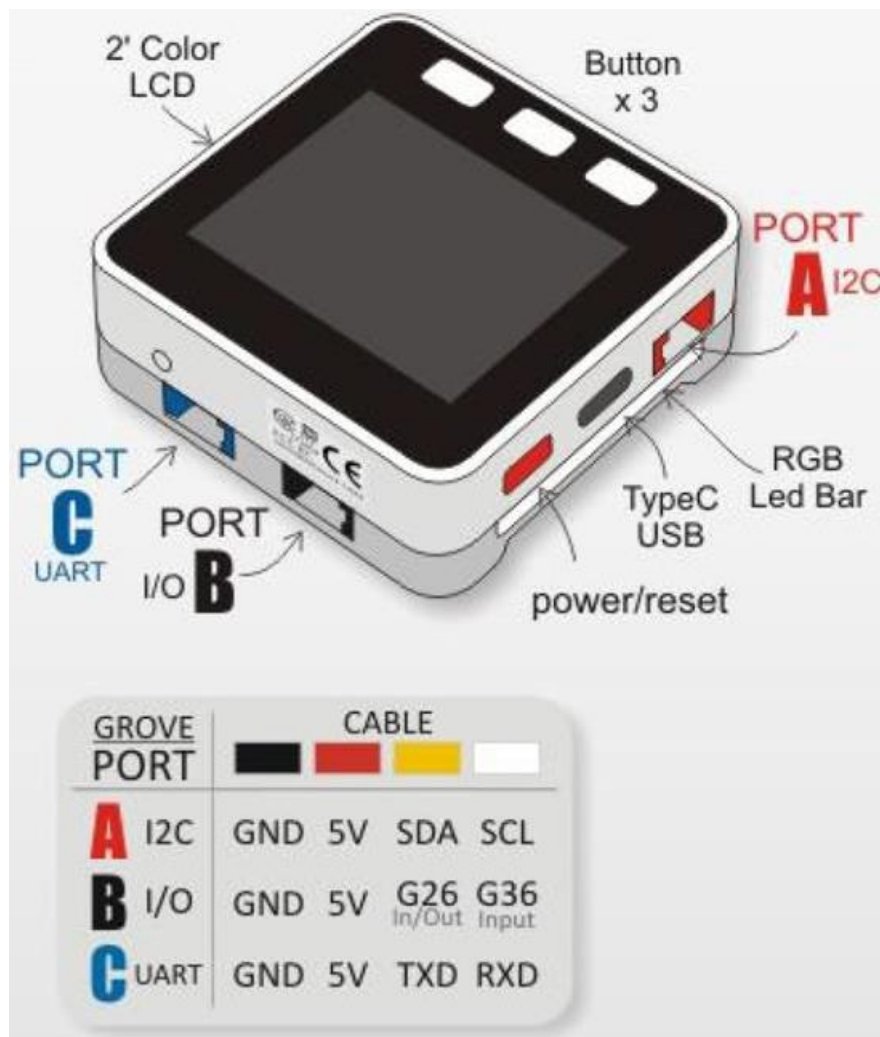
```

```

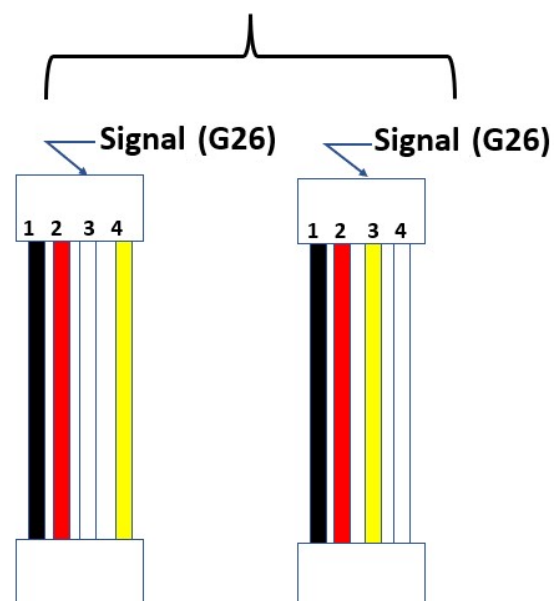
41         if (n == 0) { // If no network is found. 如果没有找到网络
42             M5.Lcd.setTextSize(2);
43             M5.Lcd.println("no networks found");
44             digitalWrite(26, 0);

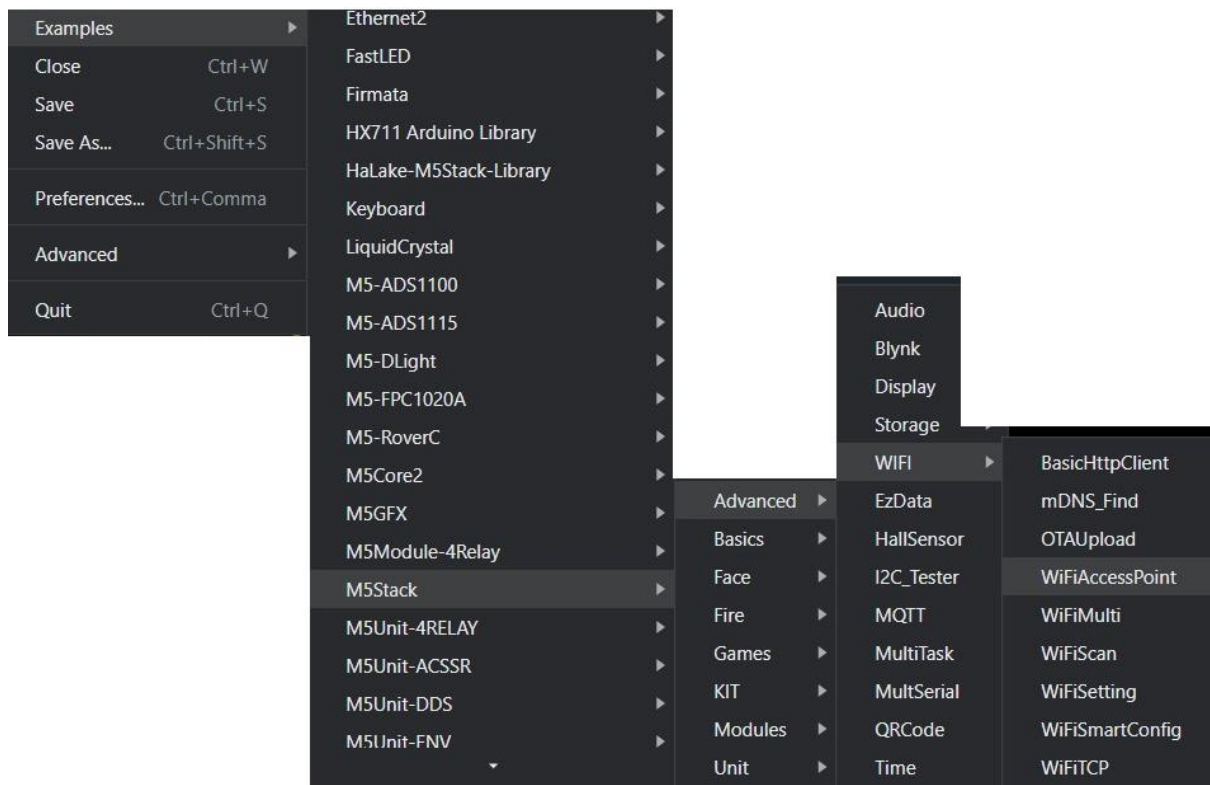
```





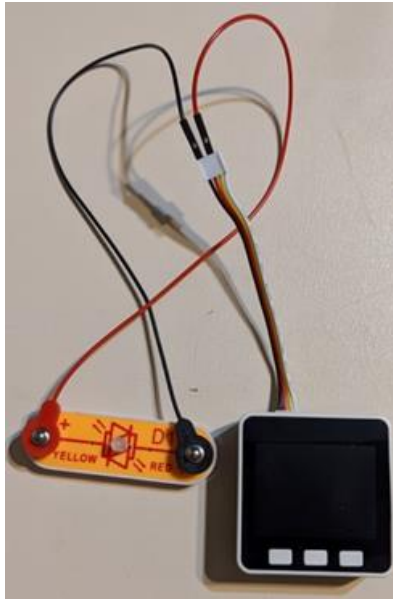
2 Jumper Wire Harness Configurations





```
54     M5.Lcd.setTextSize(2);  
55     M5.lcd.print("New Client:");  
56     String currentLine =
```

```
22 // Set these to your desired credentials. 设置你的热点名称和密码  
23 const char *ssid = "";  
24 const char *password = "";
```

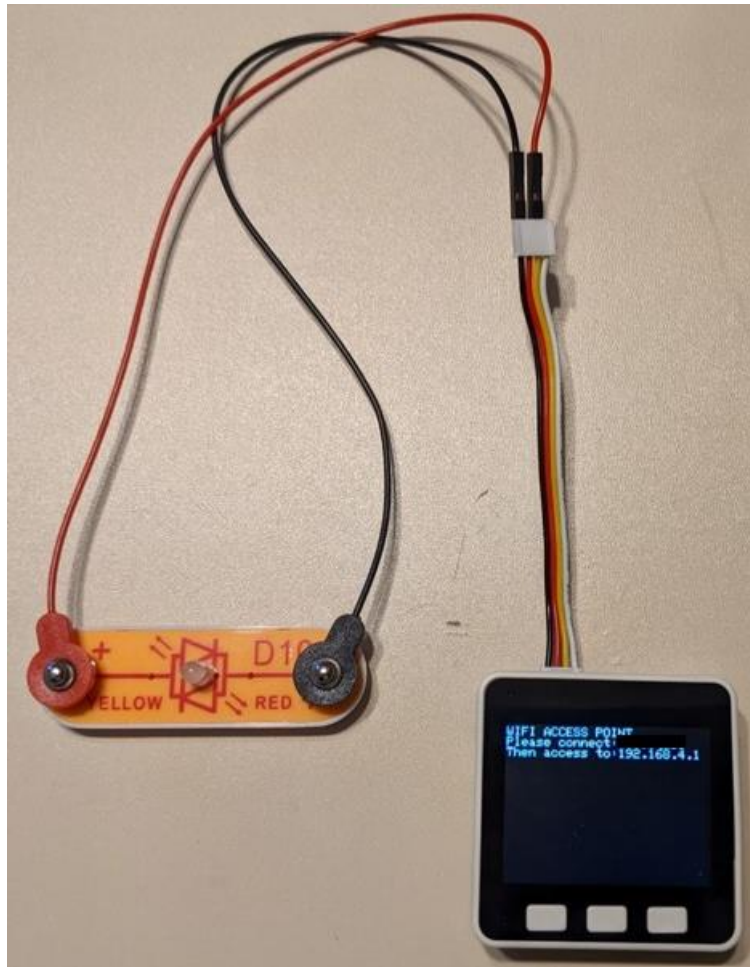


Output

```
Writing at 0x000d2753... (100 %)
Wrote 816992 bytes (521787 compressed) at 0x00010000 in 7.4 seconds (effective 881.4 kbit/s)...
Hash of data verified.
```

```
Leaving...
```

```
Hard resetting via RTS pin...
```

```
28 void setup() {
29     M5.begin();           // Init M5Stack. 初始化M5Stack
30     M5.Power.begin();     // Init power 初始化电源模块
31     M5.lcd.setTextSize(2); // Set text size to 2. 设置字号大小为2
32     M5.lcd.println(
33         "WIFI ACCESS POINT"); // Screen print string. 屏幕打印字符串.
34     M5.lcd.printf("Please connect:%s \nThen access to:", ssid);
35     WiFi.softAP(
36         ssid,
37         password); // You can remove the password parameter if you want the AP
38                   // to be open. 如果你想建立开放式热点,可以删除密码
39     IPAddress myIP = WiFi.softAPIP(); // Get the softAP interface IP address.
40                   // 获取AP接口IP地址
41     M5.lcd.println(myIP);
42     server.begin(); // Start the established Internet of Things network server.
43                   // 启动建立的物联网网络服务器
44     pinMode(26, OUTPUT);
45 }
```

```
116     if (currentLine.endsWith("GET /High")) {  
117         M5.Lcd.print("ON\n");  
118         digitalWrite(26, 1);  
119     } else if (currentLine.endsWith("GET /Low")) {  
120         M5.Lcd.print("OFF\n");  
121         digitalWrite(26, 0);  
122     }
```

