Laboratory 2: LED and M5Stack Buttons

Objectives

1. Understand the concept of Light Emitting Diode.
2. Understand how to properly connect LED to the M5Stack development kit.
3. Be able to write and understand a sketch to control the LED from the M5Stack development kit.

Reference:

https://docs.m5stack.com/en/api/core/gpio

https://docs.m5stack.com/en/api/core/button

Digital Logic

Digital Logic is an alternative term for ***Digital Electronic (or Digital Circuits)***. Digital is a way for representing the electronic signals as High (True) and Low (False) rather than the real analog values (voltages). Thus, small changes in the analog levels (voltages), such as noise from circuits or communication channels, does not affect the digital values. This allows electronic devices to switch to known states than producing specific values (voltages).

Usually, a signal level (voltage) near a reference ground level is called Low or False or 0. A signal level (voltage) near the supply voltage is called High or True or 1. With this notion in mind, we can simply refer to our digital value as high or low.

Light Emitting Diode (LED)

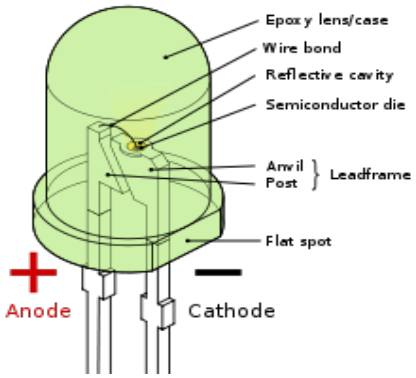

Illustration 2.1: A picture of LED (a picture from wikipedia.org)

Illustration 2.2: Electronic Symbol of LED

LED is short for ***Light Emitting Diode***. It is a special kind of diodes (unipolar electronic devices) that will emit photon (light) where there is an electronic current flow in a correct direction (from anode to cathode). Illustration 2.2 and 2.1 show the symbol and the picture of an LED.

Due to its characteristics, LED is an ideal choice for using as an output of a digital pin. Using LED as a digital output, we can easily observe the logic from the light. There are several ways to connect the LED to an output PIN. Here are two simple methods presented (see Illustration 2.3). Depending on the electronic properties of the device, the quality may vary. For most cases, LED1 should provide better quality (brighter).

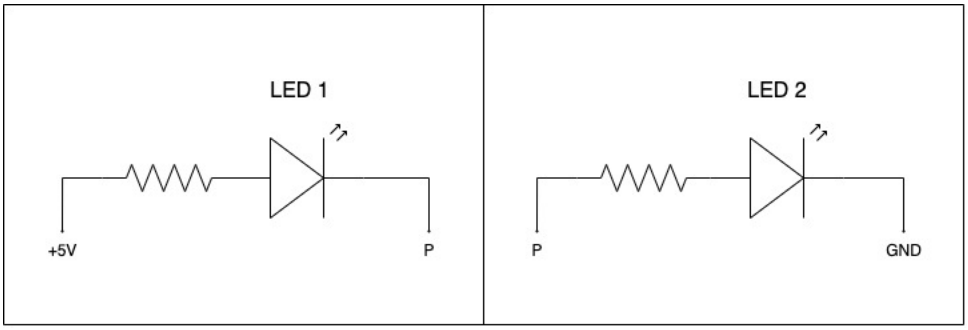

Illustration 2.3: Two methods for interfacing LED

In particular, the LED1 (connected to data pin, **P**) will emit the light when the output of P is **LOW**. The LED 2 will emit the light when the output of P is **HIGH**. To determine the correct logic, a programmer must understand the schematic of the circuits.

Arduino and Digital Input/Output (I/O) Pins

Arduino supports 13 digital input/output pins. They are called **`D0`** to **`D13`** respectively. See Illustration 2.4, 2.5 for the pin mapping. However, each pin can be either input or output at a time (specifying by the **`pinMode`** function). There are three values that can be specified as a parameter for the pinMode function: **`INPUT`**, **`OUTPUT`**, **`INPUT_PULLUP`** (case sensitive). In this lab, you will only use the **`OUTPUT`** mode with the LED. You will get more opportunities to explore the other two input modes as well in later labs.

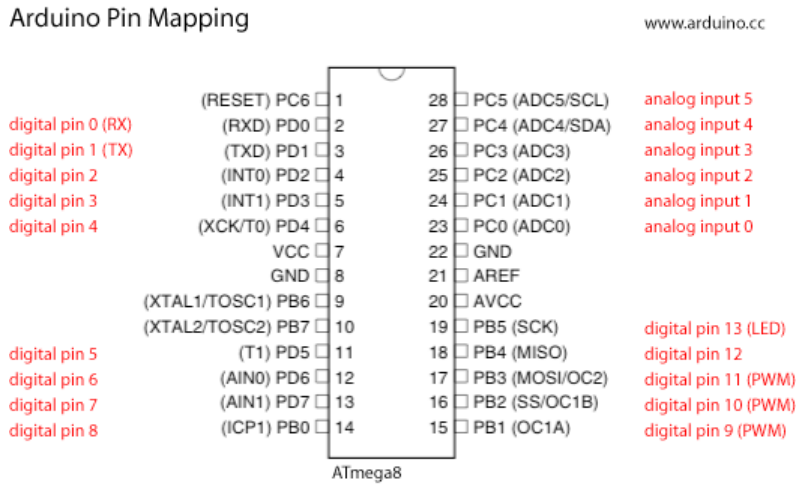

Illustration 2.4: Arduino Pin Mapping (picture taken from

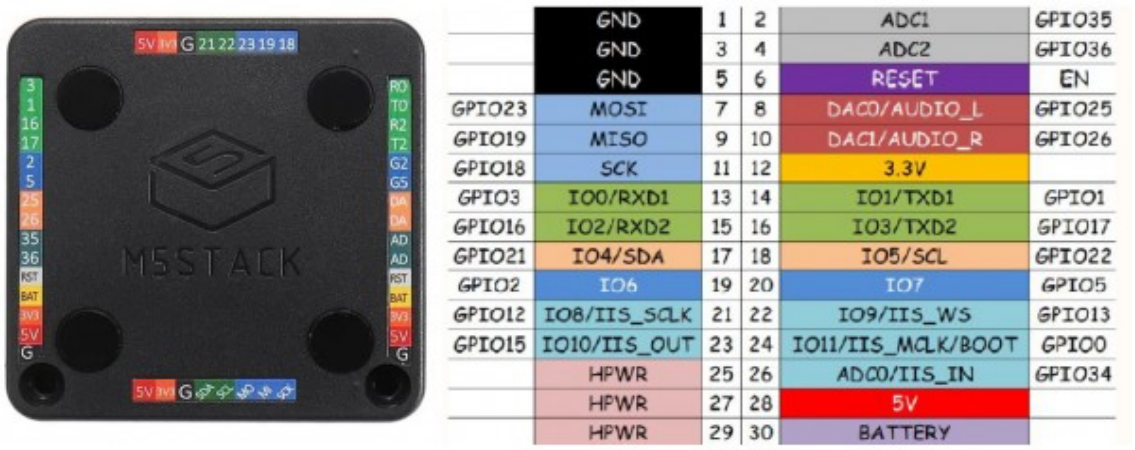

Illustration 2.5: M5Stack Pins

Arduino provides two constant values for the two digital logic (**`True/False`**). They are **`HIGH`** and **`LOW`** (case sensitive). Like others constant values provided by the system, they are all case sensitive.

The following show two functions that is required for managing the digital output of the Arduino: **`pinMode`** and **`digitalWrite`**, taken from the Arduino site. Please note that these functions do not return any values.

pinMode()

Description

Configures the specified pin to behave either as an input or an output. See the description of digital pins for details on the functionality of the pins.

As of Arduino 1.0.1, it is possible to enable the internal pullup resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pullups.

Syntax

pinMode(pin, mode)

Parameters

pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT, or INPUT_PULLUP. (see the digital pins page for a more complete description of the functionality.)

Returns

None

---

digitalWrite()

Description

Write a HIGH or a LOW value to a digital pin.

If the pin has been configured as an OUTPUT with pinMode(), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.

If the pin is configured as an INPUT, writing a HIGH value with digitalWrite() will enable an internal 20K pullup resistor (see the tutorial on digital pins). Writing LOW will disable the pullup. The pullup resistor is enough to light an LED dimly, so if LEDs appear to work, but very dimly, this is a likely cause. The remedy is to set the pin to an output with the pinMode() function.

NOTE: Digital pin 13 is harder to use as a digital input than the other digital pins because it has an LED and resistor attached to it that's soldered to the board on most boards. If you enable its internal 20k pull-up resistor, it will hang at around 1.7 V instead of the expected 5V because the onboard LED and series resistor pull the voltage level down, meaning it always returns LOW. If you must use pin 13 as a digital input, use an external pull down resistor.

Syntax

digitalWrite(pin, value)

Parameters

pin: the pin number

value: HIGH or LOW

Returns

none

Connecting LED to the M5Stack Development Kit

Illustration 2.6 show the side/top-view locations of the digital I/O pins in the M5Stack kit.

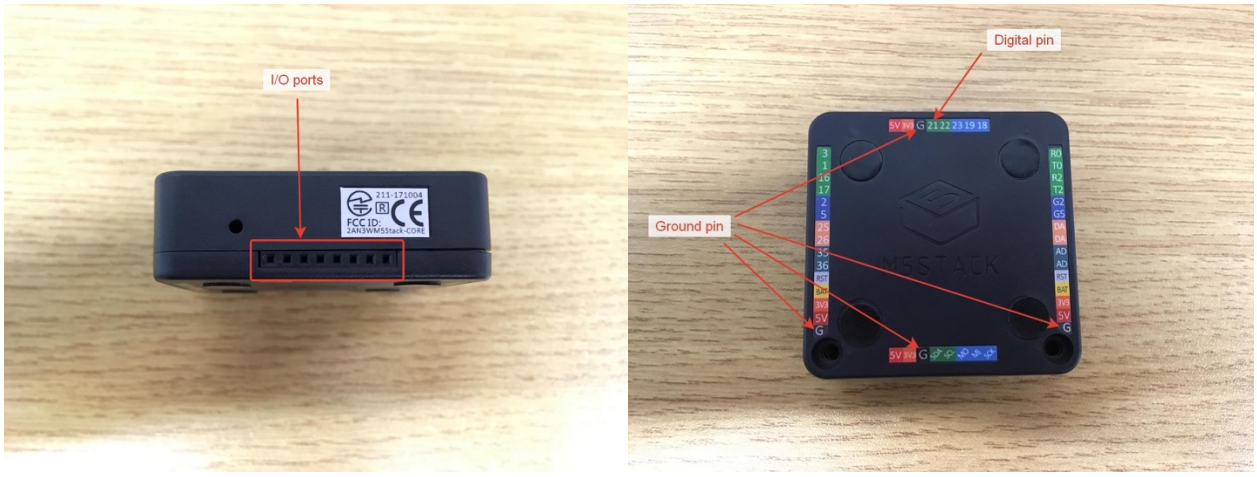

Illustration 2.6: Side/top-view locations of the digital I/O pins in the M5Stack development kit

In this lab, we will provide you with a small LED kit. The kits may come in variable forms. Illustration 2.7, show some examples of the kit with 1, 3 and 8 LEDs that are prewired with resistors. You will be using the **digital PIN 21** to connect M5Stack to the LED in the kit. Depending on your specific LED kit, you will be using either the **GD (ground) PIN** or **3V3 (VCC) PIN** on the same side on M5Stack as the PIN 21 and choose the right configuration in Illustration 2.3 to properly make your LED kit works with M5Stack.

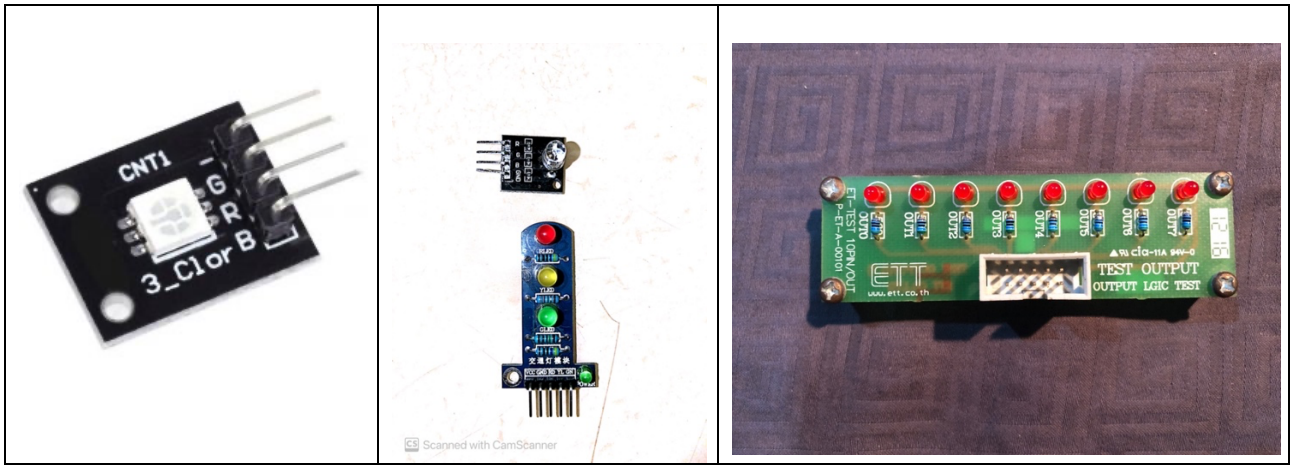

Illustration 2.7: Examples of LED Kits with prewired with resistors

M5Stack Buttons

As previously mentioned, we only use LED to connect to an output digital port, because we will use a built-in input in this lab: the three buttons on the M5Stack development kit to control the LED. When you look at the kit with the LCD screen on top of the three buttons, the three buttons: A, B, C, from left to right, can be accessed in a sketch using the variable **`M5.BtnA`**,

`M5.BtnB`, `M5.BtnC`, respectively. In addition, we can detect the status of each button in the sketch using the following functions:

- `M5.BtnA/B/C.read()`: returns the state of the button, `True(1)` == pressed and `False(0)` == released.
- `M5.BtnA/B/C.isPressed(), M5.BtnA/B/C.isReleased()`: check the button state when it was last read, and return False or True accordingly. **They do not cause the button to be read.**
- `M5.BtnA/B/C.wasPressed(), M5.BtnA/B/C.wasReleased()`: check the button state to see if it changed between the last two reads and return False and True accordingly. **They do not cause the button to be read.**

Reference: https://github.com/m5stack/M5Stack/blob/master/src/utility/Button.cpp

****Important**** Please note that with the use of the buttons, you need to add the function call `M5.update()` to read the state of the button in your `loop()` function as well.

Lab Exercises

Note: Save each task into different .ino, i.e., lab_2_1.ino, lab_2_2.ino.

Task 1: Control LED1 configuration from M5Stack

1.1 Connect M5Stack to your computer.

Connect your M5Stack to the computer using the USB cable, open the Arduino IDE, if Arduino IDE does not automatically detect M5Stack board (see status bar), select M5Stack as your board by going to the menu ***Tools > Boards > M5Core*** and choose the appropriate port that the computer connects to M5Stack by going to the menu ***Tools > PORT*** and select the right port for your set up.

1.2 Test the LEDs by connecting the LED's negative pin (cathode, marked **'-'**) to the M5Stack's **'G'** (Ground) pin. Then, connect the positive pin (anode) of the LED to either the **'3V3'** or **'5V'** pin on the M5Stack. Try each of the LED's color pins (**'R'** for Red, **'G'** for Green, and **'B'** for Blue) individually. If the LED is working, it will light up. You must have at least two functional LED colors to proceed with this lab. If an LED fails to light up, request a replacement.

1.3 Connect the LED kit in Illustration 2.6 to the M5Stack development kit according to the **LED1** configuration in Illustration 2.3. **(Note: when connect LED to VCC, use 3V3 pin)**

1.4 Create a new Arduino sketch as follows.

```
#include <M5Stack.h>

int ledPin = 21;
int ledState = LOW;

void setup(){
  M5.begin();
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, ledState);
}

void loop() {
  M5.update(); // read the press state of the key
  if (M5.BtnA.wasPressed()){
    ledState = !ledState;
    digitalWrite(ledPin, ledState);
  }
}
```

1.5 Upload the sketch to Arduino with the ‘upload’ command button  .

Task 2: Control LED with all buttons

Modify the code so that the LED state is toggled when any button; Button A, B or C is pressed.

Task 3: Connect with LED2 configuration

Change the configuration of the connection between the LED kit and the M5Stack development kit to **LED2** in Illustration 2.3 and retest the codes. You are not required to reupload the code, just press **RESET** button (the RED button next to USB port). Observe the brightness of the LED compared to **LED1** configuration. Also, observe how LED behaves differently from **LED1**.

Task 4: Control two LED colors

Connect two of the R, G, B pins to the M5Stack (use pin **21**, **22** or **19**, using other pins might conflict with internal circuit). Use Button A to control one LED color, and Button B to control another LED color, independently. Also, observe the LED when both colors are **ON**.

Task 5: Answer the following questions

5.1 According to the initial state of `dataPin` is set to **LOW**, how do you know that which LED configuration is used (**LED1** or **LED2**)?

5.2 Observe the brightness of LED when ON in both configurations. Which configuration is brighter? Try to give reason why?

5.3 What is the role of `M5.update()` in the sketch? What will happen if you remove it?

Task 6: Give feedback about this laboratory?

Group

Lab 2: LED and M5Stack Buttons

Section: _____ Date: _____

Members

Name: _____ Student ID: _____

Name: _____ Student ID: _____

Name: _____ Student ID: _____

Task		Graded by
1	LED toggling (**LED1** configuration) Note: LED should be **ON** initially	
2	Changing the M5Stack buttons to toggle the LED (must complete all the buttons)	
3	Changing the configuration of the connection between the LED and the M5Stack to **LED2** and press RESET Note: LED should be **OFF** initially	
4	Connect two LED colors controlled with two buttons independently.	

5. Answer the following questions.

5.1 According to the initial state of **dataPin** is set to **LOW**, how do you know that which LED configuration is used (**LED1** or **LED2**)?

5.2 Observe the brightness of LED when **ON** in both configurations. Can you notice a difference? Which configuration is brighter? Try to provide a reason why.

5.3 What is the role of **M5.update()** in the sketch? What will happen if you remove it?

6. Reflection about this lab.

- What did you learn from this lab?

- What was the most valuable takeaway from this lab?

- Is there anything that you did not understand?