



SOFTWARE PARK
THAILAND



อ้างอิง <https://medium.com/@prawared.bw/owner-arduino-library-f0eea1dd3663>

Micro Controller

Day3

By: Arkom Thaicharoen



Software Park Thailand
</Code Camp>

Agenda

- Library
- การบ้าน Library
- RTC
- การบ้าน RTC
- Tact Switch and Buzzer
- การบ้าน Tact Switch and Buzzer

Library



Library





Library

- การสร้าง Libraries ภาษา C++ ใช้เองบน Arduino ใช้ฟังก์ชันต่าง ๆ เช่น digitalWrite() , digitalRead() ฯลฯ
- ฟังก์ชันดังกล่าวเพียงแค่ใส่ตัวแปรลงไป ก็ทำงานได้แล้วเช่น digitalWrite(PIN , HIGH) จะเห็นว่าเราใส่ลงไปแค่ 2 ตัวแปร คือ PIN และ HIGH
- การสร้างฟังก์ชันขึ้นมาเพื่อใช้งาน ก็จะช่วยลดการเขียนโค้ดซ้ำซ้อนในอนาคตหากต้องการทำงานแบบเดียวกัน



Library

- ปัจจุบันการหา Libraries มาใช้งานได้เลย เพราะมีคนเขียน Libraries ขึ้นมาแล้ว
- แต่หากต้องการสร้างหรือศึกษานั้นจะทำให้เราเข้าใจการทำงานของฟังก์ชันมากยิ่งขึ้น และสามารถแก้ปัญหาในการเขียนโปรแกรมอนาคตได้นั่นเอง
- เบื้องต้นต้องทำความเข้าใจการทำงานของไฟล์นามสกุล .h และ .cpp ก่อน โดยเขียน Libraries Calculator เป็นตัวอย่าง



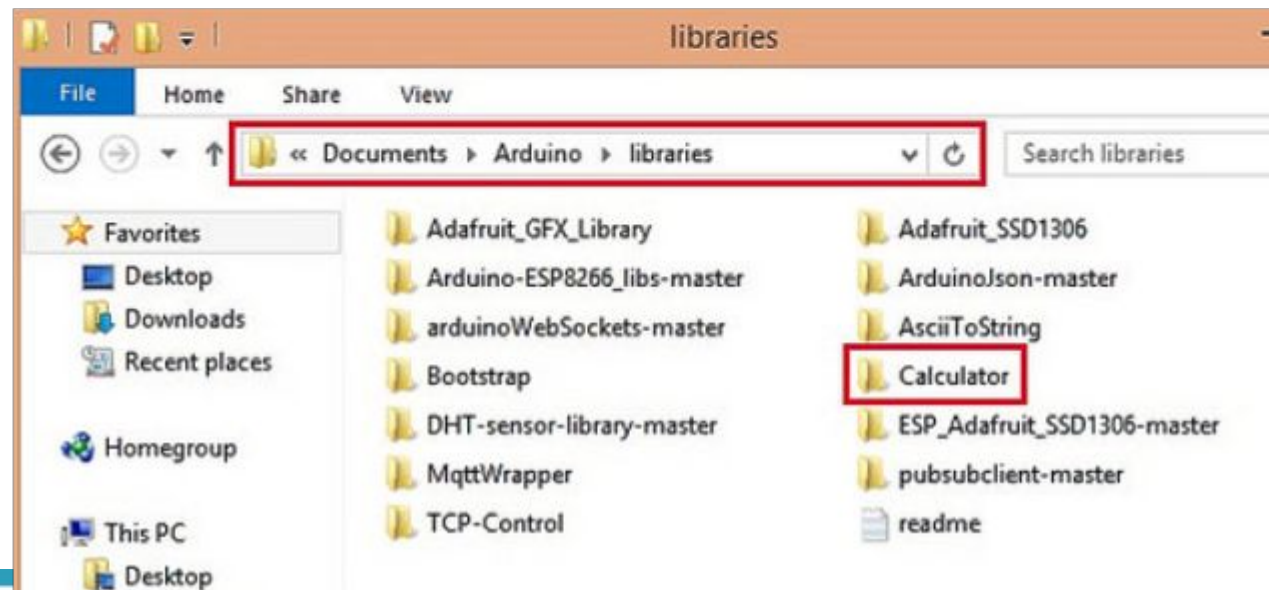
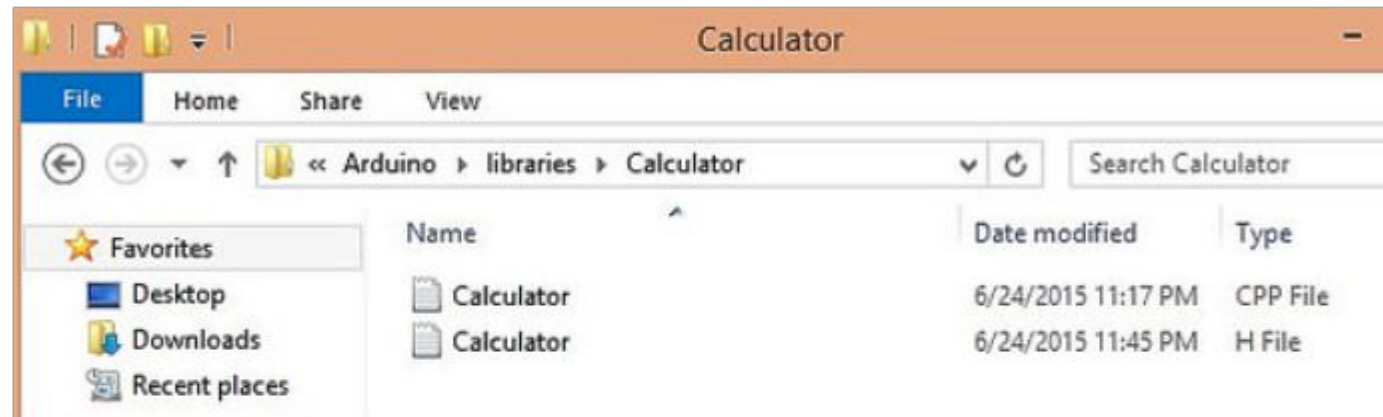
Lab 1: Digital LED

1. ทำการสร้าง library “Calculator” ตามขั้นตอน ด้วยการใช้ VScode
2. ทำการติดตั้ง library ที่เขียนแล้วลองส่งค่าคำนวณทาง serial monitor



- Documents > Arduino > libraries

- โดยทำการสร้าง Folder ให้ตั้งชื่อเดียวกันกับ Class นำไฟล์ .h และ .cpp ใส่ไว้ข้างใน





Library

Calculator.h

```
Calculator.h  Calculator.cpp
1  class Calculator
2  {
3
4      public:
5
6          int Sum = 0;
7
8          Calculator();
9
10         void Fn1(int value1, int value2);
11         void Fn2(int value1, int value2);
12         void Fn3(int value1, int value2);
13         void Fn4(int value1, int value2);
14
15     };
```



Library

อธิบาย -> สร้างไฟล์ .h

- บรรทัดที่ 1 ประกาศชื่อ Class (ตัวอย่าง จะใช้ชื่อ Calculator)
- บรรทัดที่ 4 ประกาศแบบ Public
Public คือ เรียกใช้ตัวแปรหรือฟังก์ชัน ได้ทุกไฟล์ เพียงแค่ `#include .h` ข้างบนไฟล์
Private คือ เรียกใช้ตัวแปรหรือฟังก์ชัน ได้แค่ภายในไฟล์ .h เท่านั้น
- บรรทัดที่ 6 ประกาศตัวแปร ชื่อ Sum ชนิด int เพื่อนำมาเก็บค่าผลรวม
- บรรทัดที่ 8 ประกาศฟังก์ชัน ชื่อเดียวกับ Class เพื่อให้ ฟังก์ชันในไฟล์ .cpp รู้จักกับ Class หลัก
- บรรทัดที่ 10-13 ให้ตั้งชื่อฟังก์ชันได้เลย พร้อมกับ กำหนดตัวแปรที่จะรับเข้ามา เช่น `digitalWrite(1, 1)` ตัวแปรที่ใส่เข้าไปใน ฟังก์ชัน digitalWrite



Library

Calculator.cpp

```
Calculator.h  Calculator.cpp
1  #include "Calculator.h"
2
3  Calculator::Calculator() {
4
5  }
6
7  void Calculator::Fn1(int value1, int value2) {
8      Sum = value1 + value2;
9  }
10
11 void Calculator::Fn2(int value1, int value2) {
12     Sum = value1 - value2;
13 }
14
15 void Calculator::Fn3(int value1, int value2) {
16     Sum = value1 * value2;
17 }
18
19 void Calculator::Fn4(int value1, int value2) {
20     Sum = value1 / value2;
21 }
```



Library

อธิบาย -> สร้างไฟล์ .cpp

- บรรทัดที่ 1 ทำการ `#include` ไฟล์ `.h` เข้ามา
- เหตุผลที่ต้อง `#include .h` เข้ามาเนื่องจากทำให้ทั้ง 2 ไฟล์ รู้จักกันนั่นเอง เพื่อให้ ตัวหลัก (Class `.h`) รู้จักกับฟังก์ชัน และสามารถเรียกใช้งานได้
- บรรทัดที่ 3 ไฟล์ `.cpp` จะเห็นได้ว่าต้องประกาศชื่อ Class ที่เราตั้งใน `.h` เพื่อให้ Class หลักรู้จักกับฟังก์ชัน
- บรรทัดที่ 7 เป็นต้นไป สามารถสร้าง ฟังก์ชันโดย ประกาศ `void` ตามด้วย ชื่อ Class และเครื่องหมาย `::` ตามด้วยชื่อฟังก์ชัน หลังจากนั้น ให้กำหนด ตัวแปรที่จะรับเข้ามาทำงาน ภายในฟังก์ชัน เหมือนที่อธิบายข้างต้นที่ผ่านมา
- โดยภายในรูป ตัวอย่างฟังก์ชัน สามารถเรียกใช้ตัวแปร `Sum` เพราะ ในไฟล์ `.h` ประกาศ `Public`



Library

ทดลองสร้างไฟล์ที่
Calculator →
examples →
Calculator.ino
แล้วทดลองผล

The screenshot shows the Arduino IDE interface with the 'Calculator' sketch loaded. The sketch code is as follows:

```
1 #include "Calculator.h"
2
3 void setup() {
4
5   Serial.begin(115200);
6
7 }
8
9 void loop() {
10
11   Calculator Me;
12
13   Me.Fn1(50, 50);
14
15   Serial.println(Me.Sum);
16
17   delay(500);
18
19 }
```

Below the code editor, the 'Done Saving' message is visible. The serial monitor at the bottom shows the output 'flush complete'.

Generic ESP8266 Module, 80 MHz, 115200, 512K (64K SPIFFS) on COM5

Homework Library



การบ้าน

1. สร้าง Library การใช้งาน LDR **ไม่ต้องต่อ LED**
 - a. ตัวอย่าง code ตัวเซนเซอร์วัดอุณหภูมิคือ LM35
<https://robotics-za.blogspot.com/2015/02/arduino.html>
 - b. พัฒนา Library LDR ขึ้นเองแล้วส่ง source code
 - c. แสดงผล Library LDR แล้วส่ง clip บันทึกการส่งไฟและ serial monitor แสดงค่าความเข้มแสง

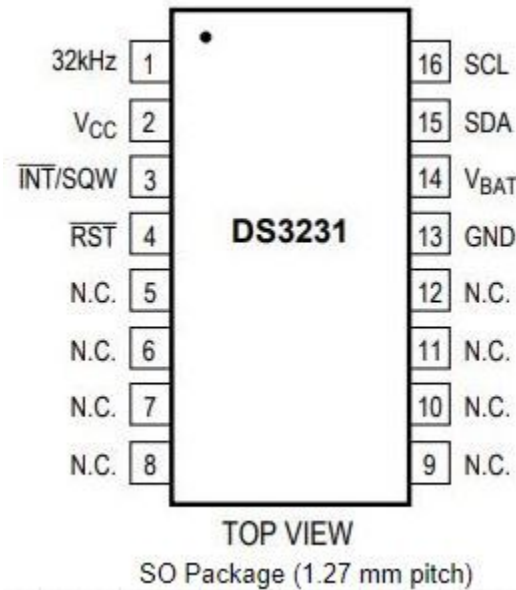
RTC

Real Time Clock



RTC:Real Time Clock

- DS3231 เป็น chip แบบ RTC (Real Time Clock) ที่มีความแม่นยำสูงสามารถตั้งค่า วัน เวลา ได้อย่างง่าย และมีวงจรวัดอุณหภูมิ ที่สามารถแสดงอุณหภูมิที่ภายนอกได้



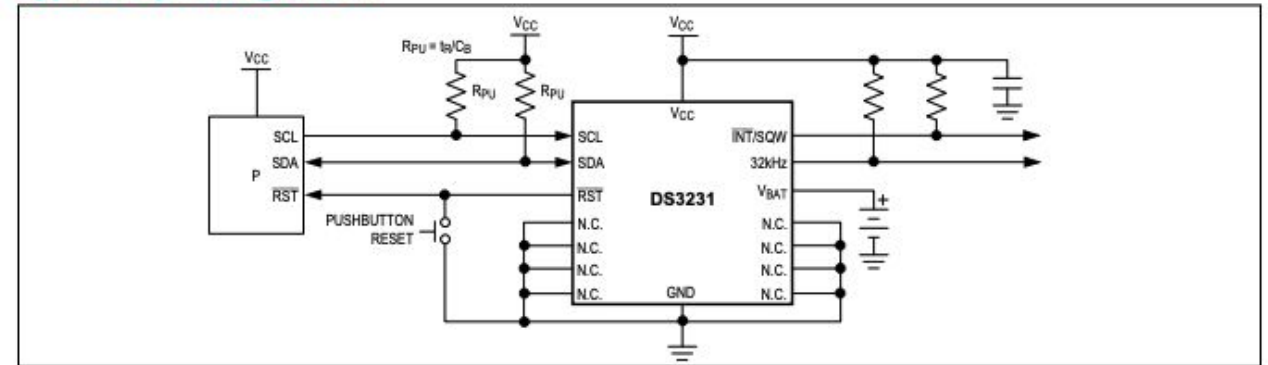


RTC:Real Time Clock

Benefits and Features

- Highly Accurate RTC Completely Manages All Timekeeping Functions
 - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
 - Accuracy $\pm 2\text{ppm}$ from 0°C to $+40^{\circ}\text{C}$
 - Accuracy $\pm 3.5\text{ppm}$ from -40°C to $+85^{\circ}\text{C}$
 - Digital Temp Sensor Output: $\pm 3^{\circ}\text{C}$ Accuracy
 - Register for Aging Trim
 - RST Output/Pushbutton Reset Debounce Input
 - Two Time-of-Day Alarms
 - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
- Fast (400kHz) I2C Interface
- Battery-Backup Input for Continuous Timekeeping
- Low Power Operation Extends Battery-Backup Run Time
- 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to $+70^{\circ}\text{C}$) and Industrial (-40°C to $+85^{\circ}\text{C}$)
- Underwriters Laboratories® (UL) Recognized

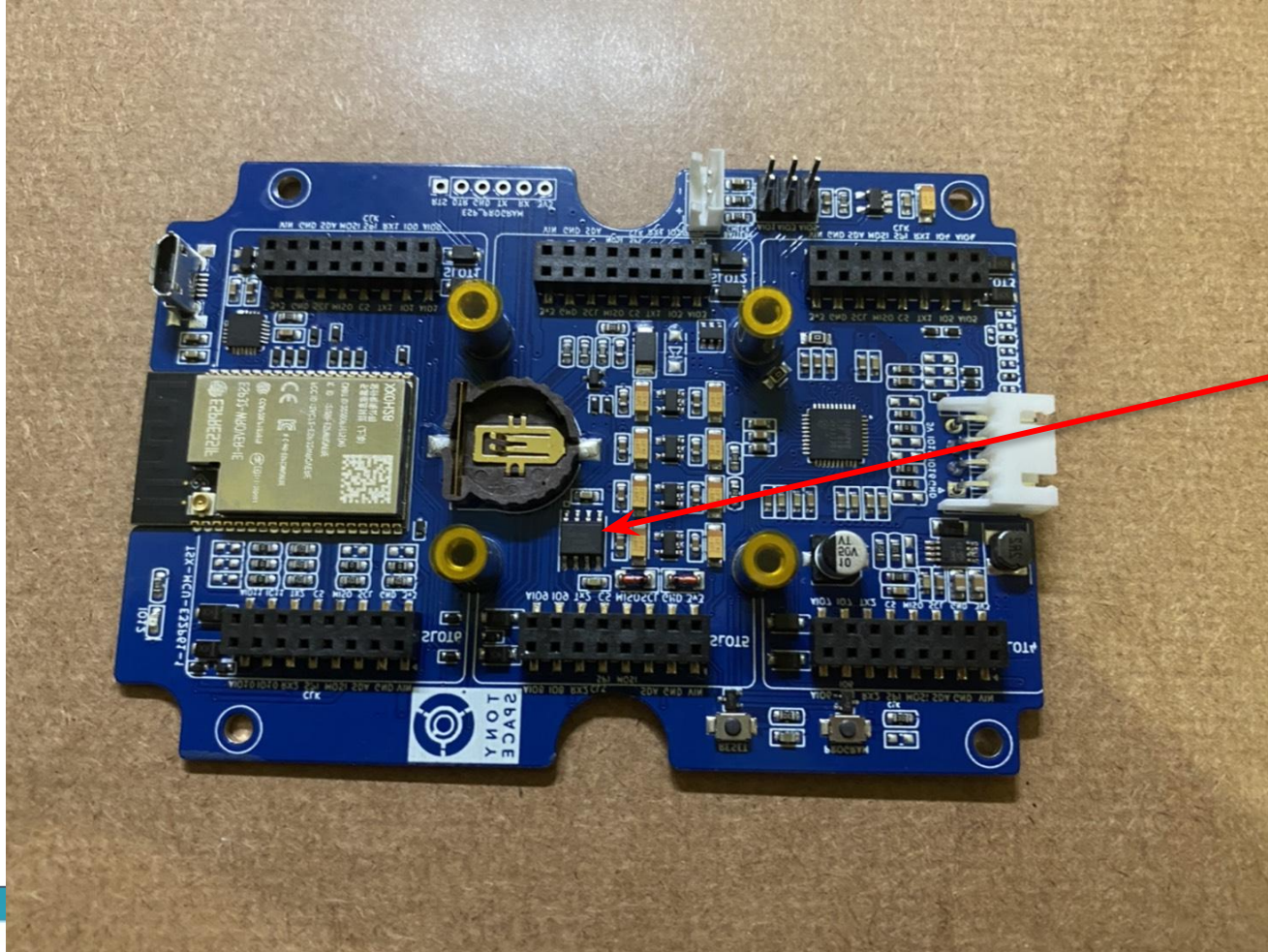
Typical Operating Circuit



Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.



RTC:Real Time Clock

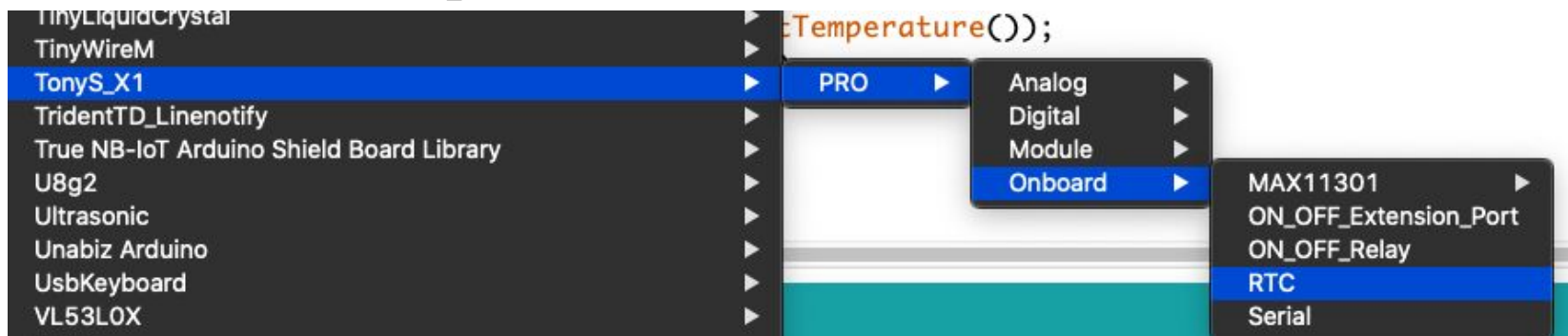


RTC

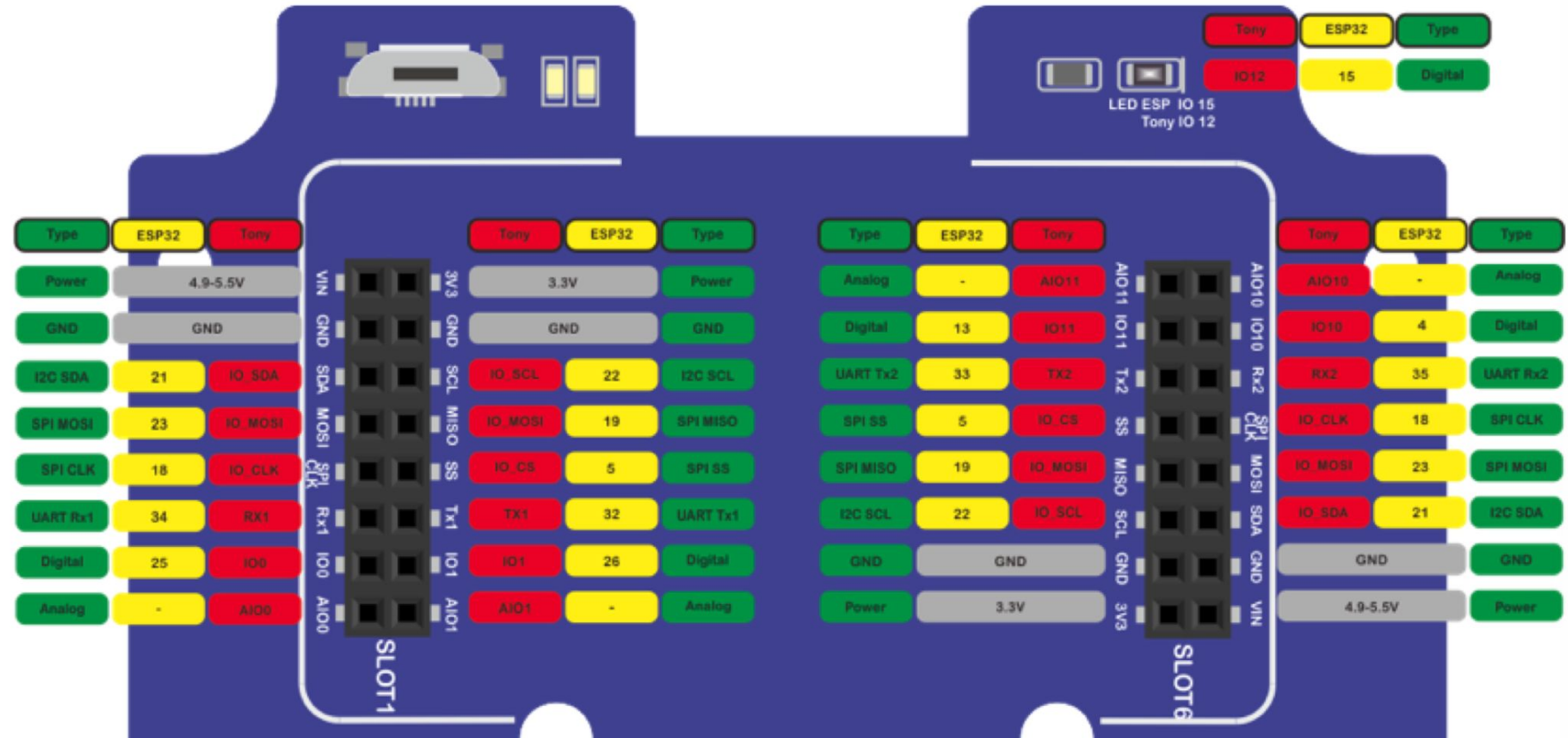


Lab 2: RTC:Real Time Clock

1. ใส่ถ่าน 1220 แล้วใช้ code ตัวอย่างลงบอร์ด TonySpace โดยที่ไม่ต้องต่อวงจรแล้วดูผลทาง serial monitor



1.



Homework

RTC:Real Time Clock



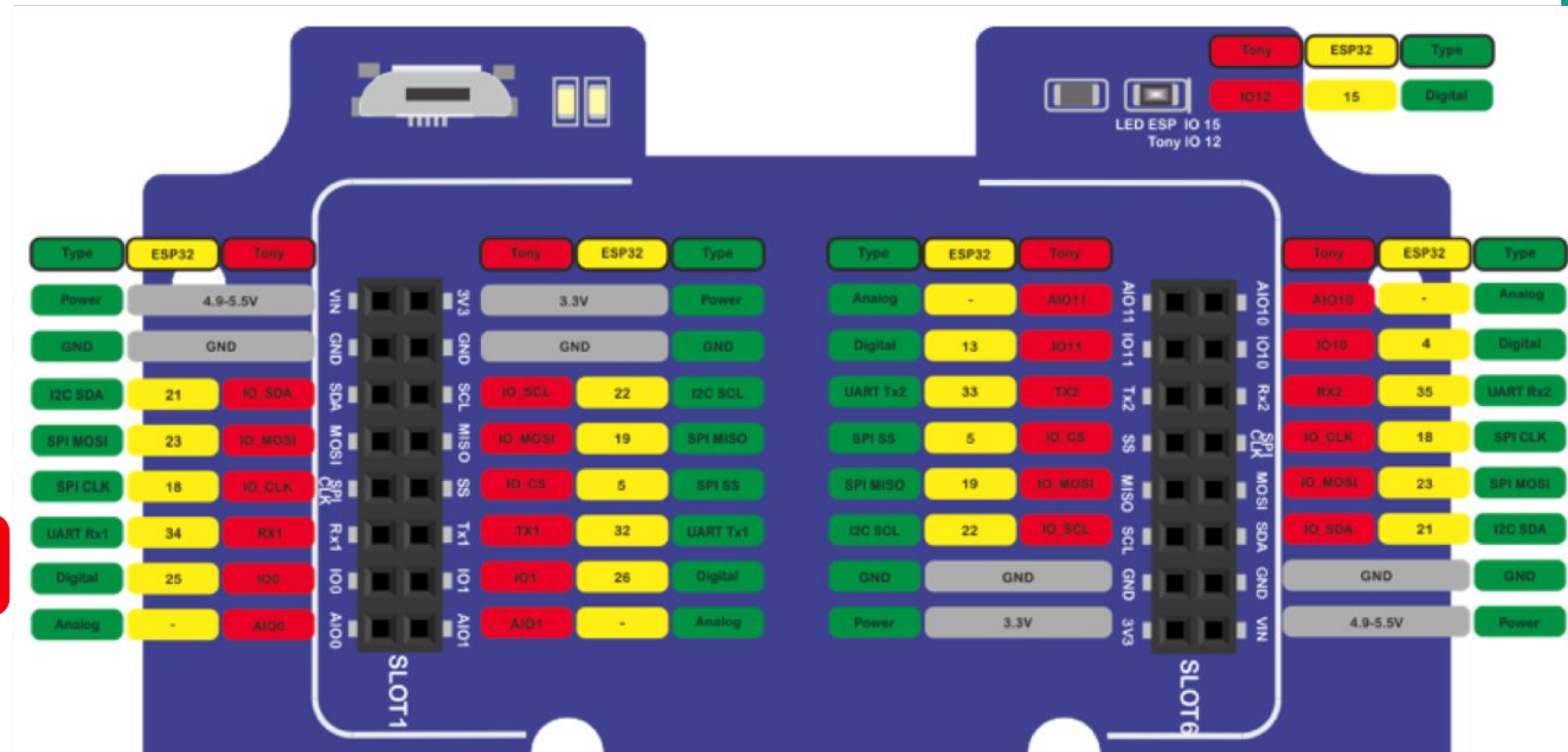
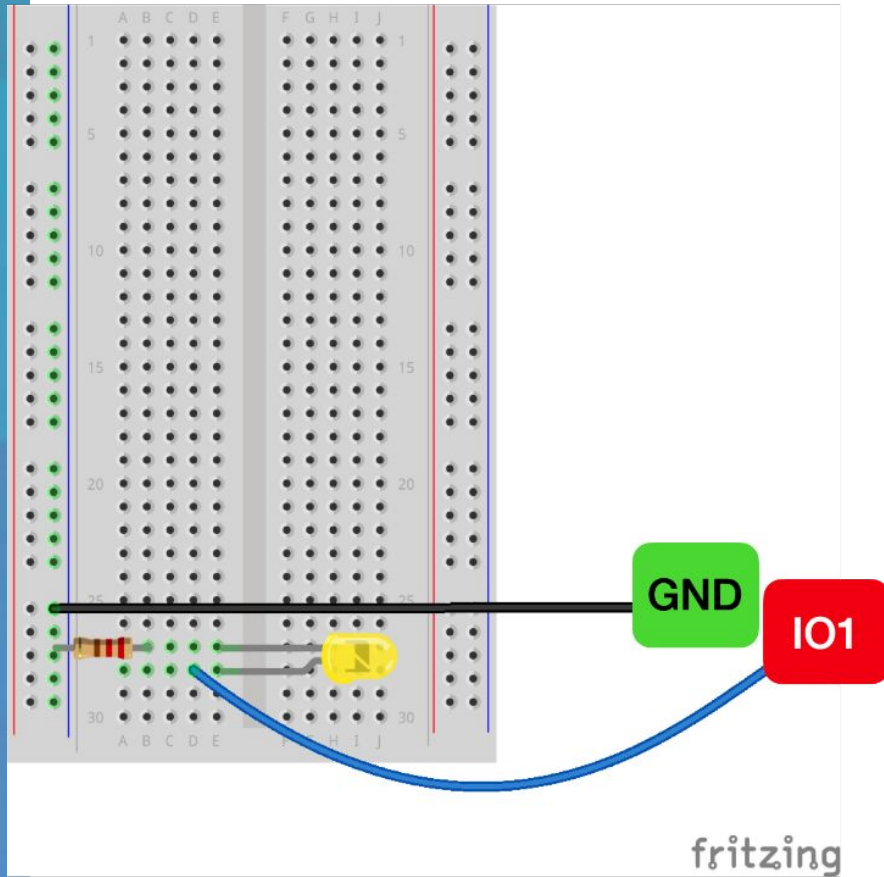
การบ้าน

1. ดัดแปลงโปรแกรม RTC และต่อวงจรจากตัวอย่างภาพ LED_1 โดยทำงานดังนี้ หาก Temperature = 32 ที่บ้าน ให้สร้างเงื่อนไขว่าหากมากกว่า 32 ให้ LED ติด หากไม่ใช่ให้ LED ดับ ส่ง clip และ source code
2. ดัดแปลงโปรแกรม RTC และต่อวงจรจากตัวอย่างภาพ Multi_LED โดยทำงานดังนี้
 1. serial print “hour:minute - day/month/year”
 2. serial print “Temperature = อุณหภูมิจริงเป่าเบาๆให้ร้อนได้ห้ามเอาของร้อนมานาบเพราะจะพังเอา”
3. เงื่อนไข หากครบ 1 นาที LED ติด 1 ดวงและ serial print “LED1 on”
หากครบ 2 นาที ติด 2 ดวงและ serial print “LED1 on”
หากครบ 3 นาที ไฟดับทุกดวงและ serial print “All LED off”

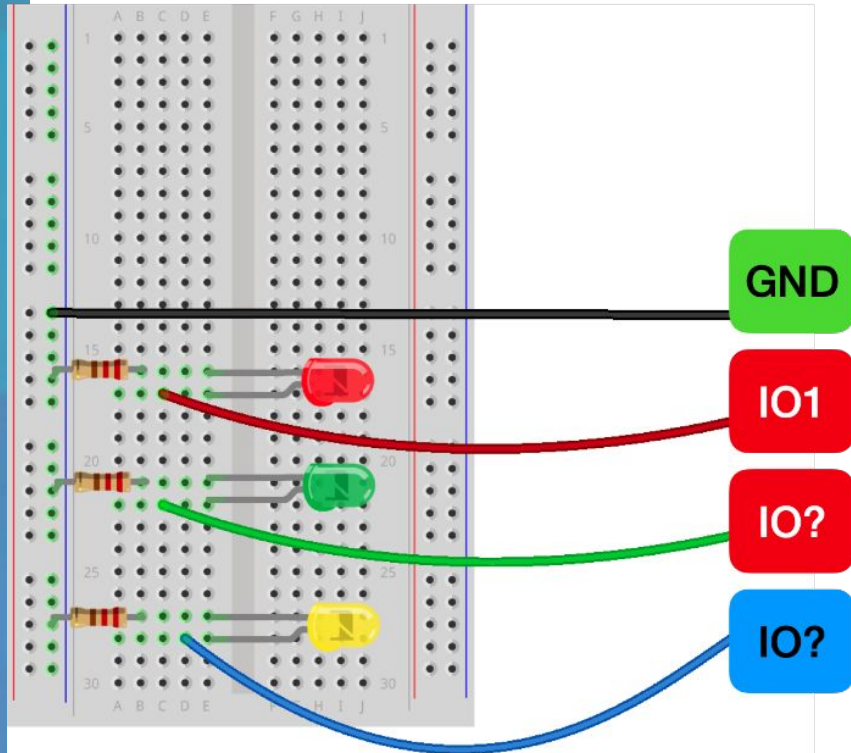
1.

```
1 #include "TonyS_X1.h"
2
3 int incomingByte = 0;
4 int ledPin = I01;
5
6 void setup() {
7   Serial.begin(115200);
8   Tony.pinMode(ledPin, OUTPUT);
9   Serial.println ("Type 'a' for blinky");
10 }
11 void loop() {
12
13
14
15
16
17   เขียนขึ้นเอง
18
19
20
21
22
23 }
```

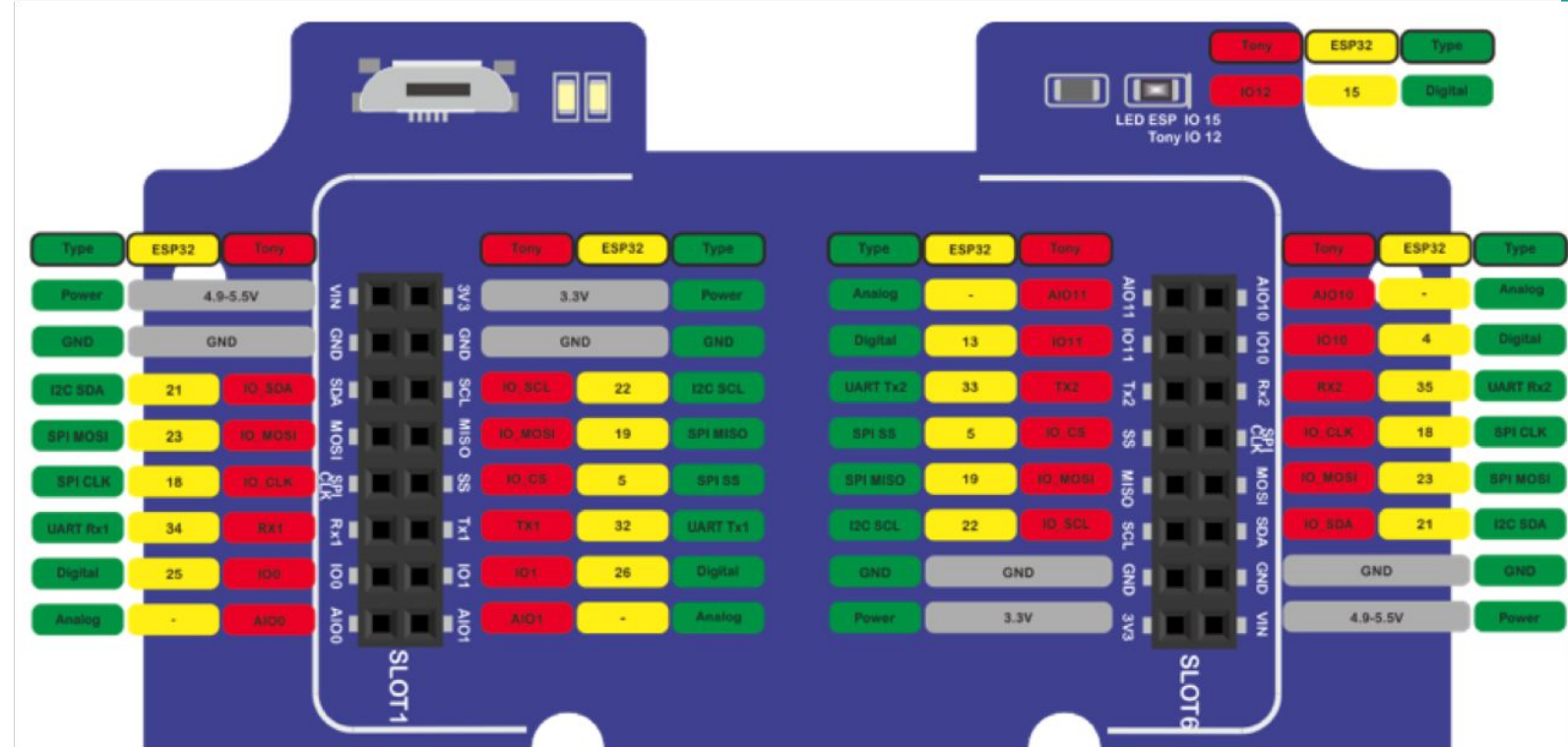
1.



2.



fritzing

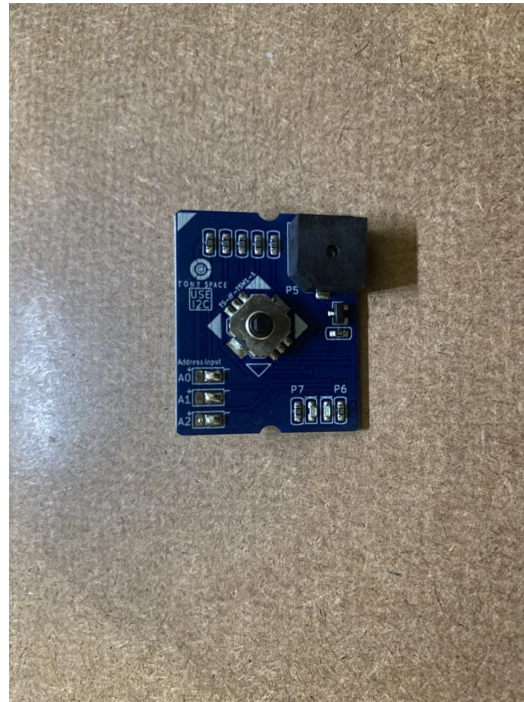


Tact Switch and Buzzer



Tact Switch and Buzzer

Tony S Tact Switch and Buzzer module

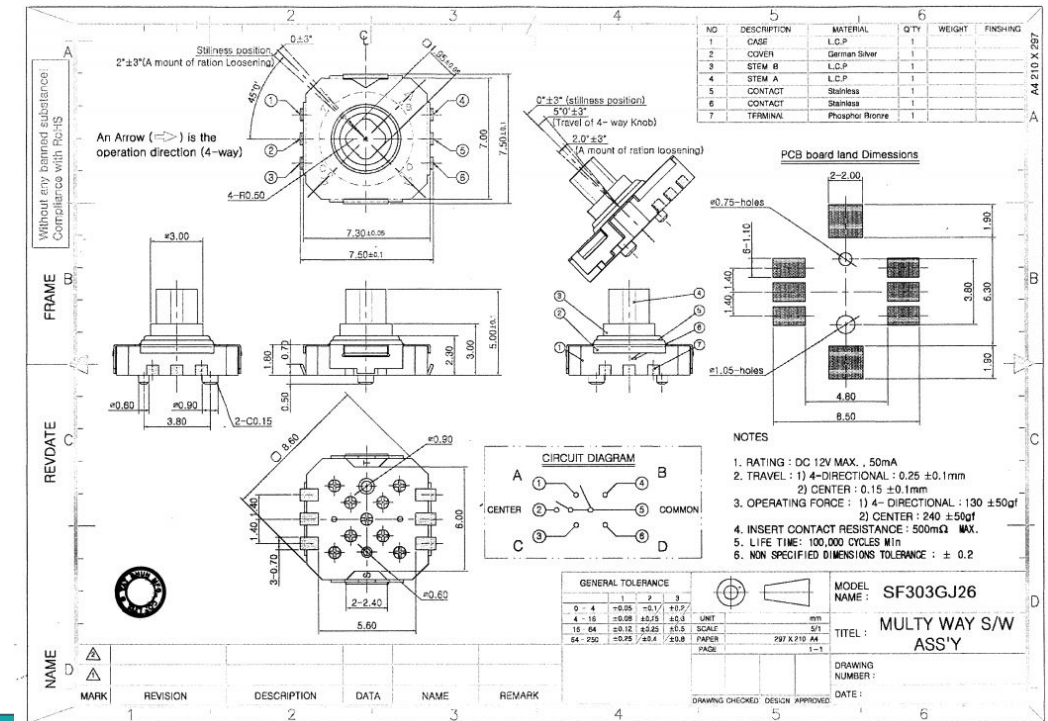




Tact Switch and Buzzer

Tony S Tact Switch and Buzzer module

- A 5-way tactile switch allows for a joystick-like interface in a very small package.

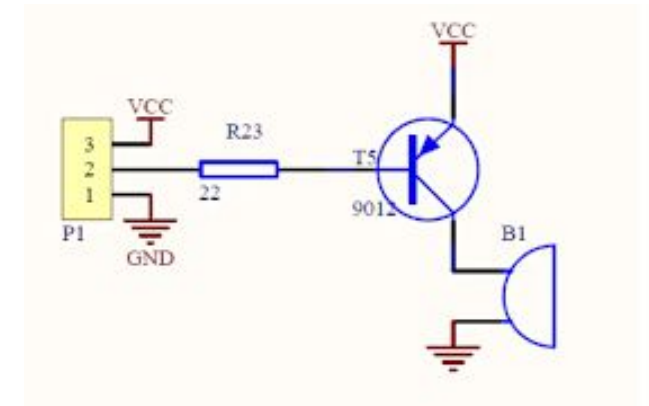
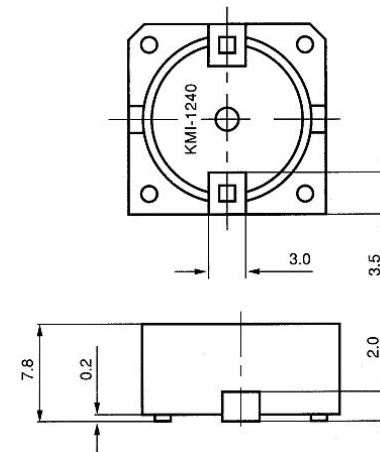




Tact Switch and Buzzer

Tony S Tact Switch and Buzzer module

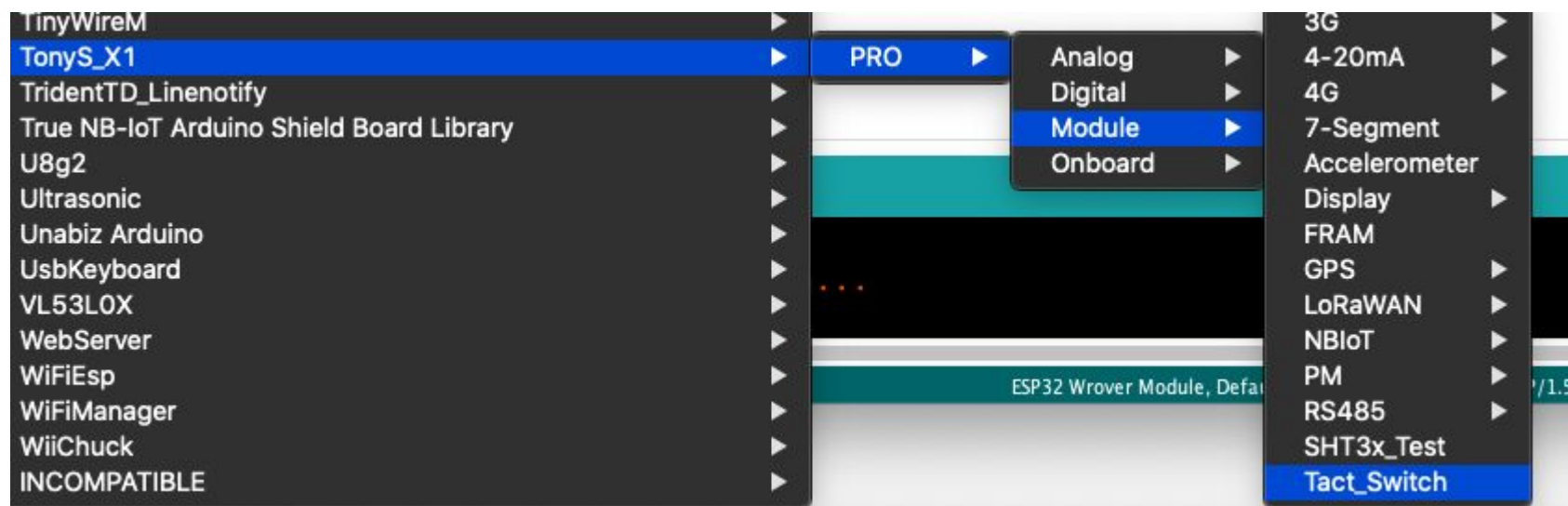
- ลำโพงบี๊เซอร์เป็นอุปกรณ์ที่ให้กำเนิดเสียงทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าให้อยู่ในรูปสัญญาณเสียง ลำโพงบี๊เซอร์มีอยู่ 2 ประเภท ได้แก่
 1. แบบแอคทีฟ (Active Buzzer) ลำโพงชนิดนี้มีวงจรกำเนิดความถี่อยู่ภายใน สามารถสร้าง สัญญาณเสียงเตือนได้ทันทีเพียงแค่จ่ายแรงดันไฟฟ้าเข้าไป
 2. แบบพาสซีฟ (Passive Buzzer) ลำโพงชนิดนี้ทำงานเหมือนลำโพงขนาดเล็ก คือ ถ้าป้อน แรงดันไฟฟ้ากระแสตรงเข้าไปไม่มีเสียงถ้าต้องการให้มีสัญญาณเสียงต้องทำการป้อนสัญญาณความถี่ เข้าไป ลำโพงชนิดนี้สามารถกำเนิดเสียงที่มีความแตกต่างกันตามความถี่ที่ป้อนเข้ามา
- Tony S Tact Switch and Buzzer module คือชนิด Active Buzzer
- **Features and Benefits**
 - Piezo buzzer with a continuous tone
 - Surface mounting
 - Minimum supply voltage of 3 V DC
 - Maximum supply voltage of 20 V DC
 - 83 dB sound level
 - Internal drive
 - Length of 16.9 mm
 - Depth of 16.9 mm
 - Height of 7.8 mm
 - Operating temperature range of -30°C to +70°C
 - Supply current of 8 mA
 - Minimum frequency of 3500 Hz
 - Maximum frequency of 4.5 kHz





Lab 3: Tact Switch and Buzzer

1. เปิด code Tact_Switch



1.

- เมื่อเขียน code เสร็จทำการ upload
- ดึงสาย USB ออก
- เสียบ Tact Switch module ที่ slot1
- ให้ TA ตรวจสอบว่าเสียบตรงช่องทุกช่อง
- TA ตรวจสอบเสร็จแล้วทำการเสียบ USB เพื่อจ่ายไฟ
- เปิด serial monitor เพื่ออ่านค่า
- ทดลองขยับคันโยกไปทางขวาหรือซ้ายไฟ LED ที่บอร์ด จึงติด



1.

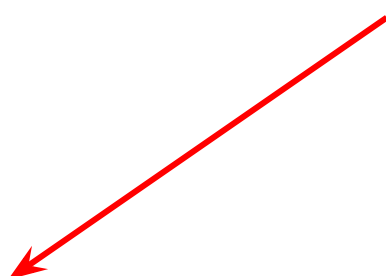
```
5 void setup()
6 {
7   Serial.begin(115200);
8   Tony.begin(); //---- begin Library
9   TonyTactSW.begin(SLOT1); //---- Select Slot of module
10
11
12   // ----- TEST BUZZER -----//
13   TonyTactSW.onBuzzer();
14   delay(500);
15   TonyTactSW.offBuzzer();
16   Tony.pinMode(LED_BUILTIN, OUTPUT); //---- Set Pin IO12 (LED_BUILTIN) to OUTPUT
17
18   //----- TEST LED -----//
19   for (uint8_t i = 0; i < 5; i++)
20   {
21     TonyTactSW.onLED1(); // ON LED 1
22     delay(500);
23     TonyTactSW.offLED1(); // OFF LED 1
24     TonyTactSW.onLED2(); // ON LED 2
25     delay(500);
26     TonyTactSW.offLED2(); // OFF LED 2
27   }
28 }
```

เพิ่ม

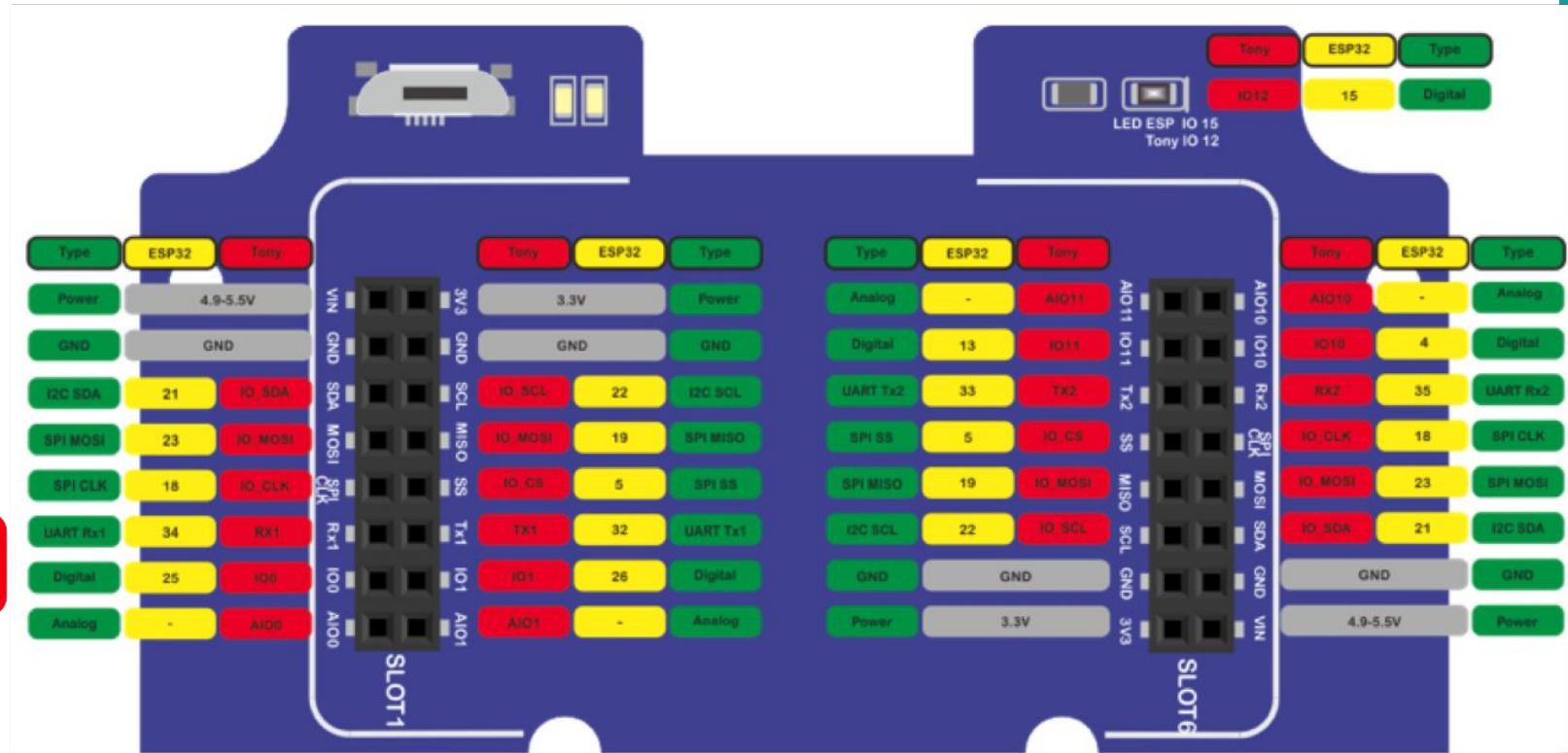
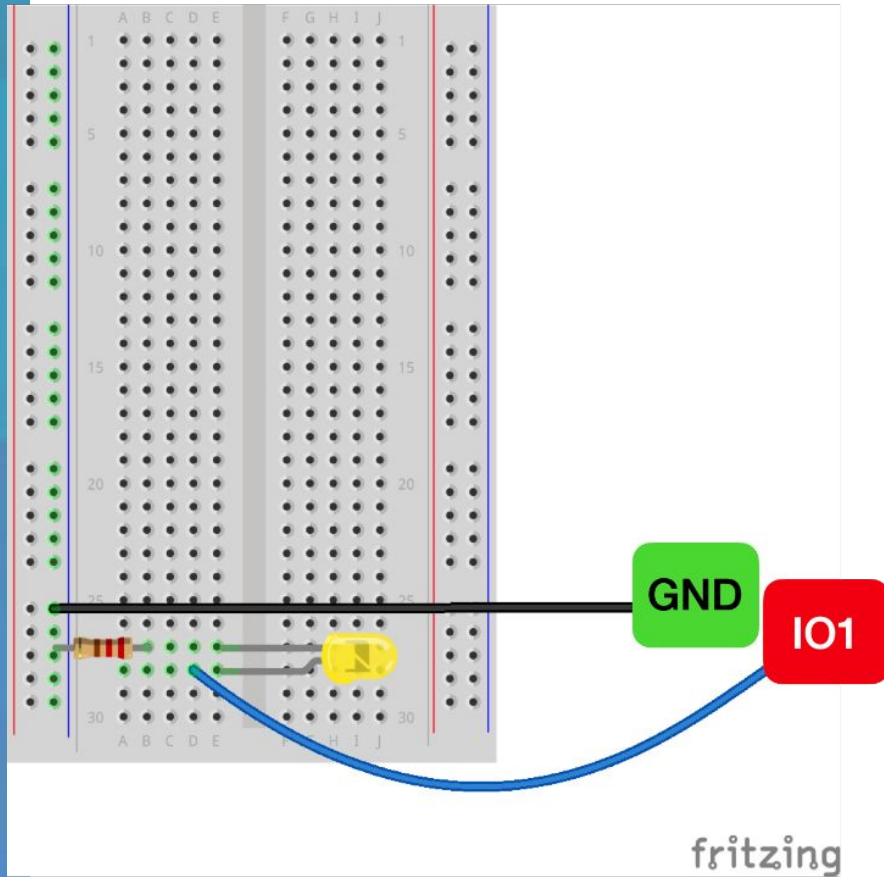
1.

```
38     Serial.print(" = ");
39     Serial.println(dataADC);
40 }
41
42 /* Read all port (This method is faster than read port by port
43     31 = non
44     30 = down 0
45     29 = pull 1
46     23 = left 3
47     27 = right 2
48     015 = up 4
49 */
50 uint8_t data_all = TonyTactSW.readAll();
51 Serial.print("DATA ALL PORTS ");
52 Serial.print(" = ");
53 Serial.println(data_all);
54
55 if (data_all == 27)
56 { Tony.digitalWrite(LED_BUILTIN, HIGH); //-- Write HIGH to pin I012 (LED_BUILTIN)
57   delay(1000);
58 }
59 Tony.digitalWrite(LED_BUILTIN, LOW); //-- Write LOW to pin I012 (LED_BUILTIN)
60 delay(1000);
61
62 Serial.println();
63 delay(100);
64 }
```

edit code



1.



Homework

Tact Switch and Buzzer



การบ้าน

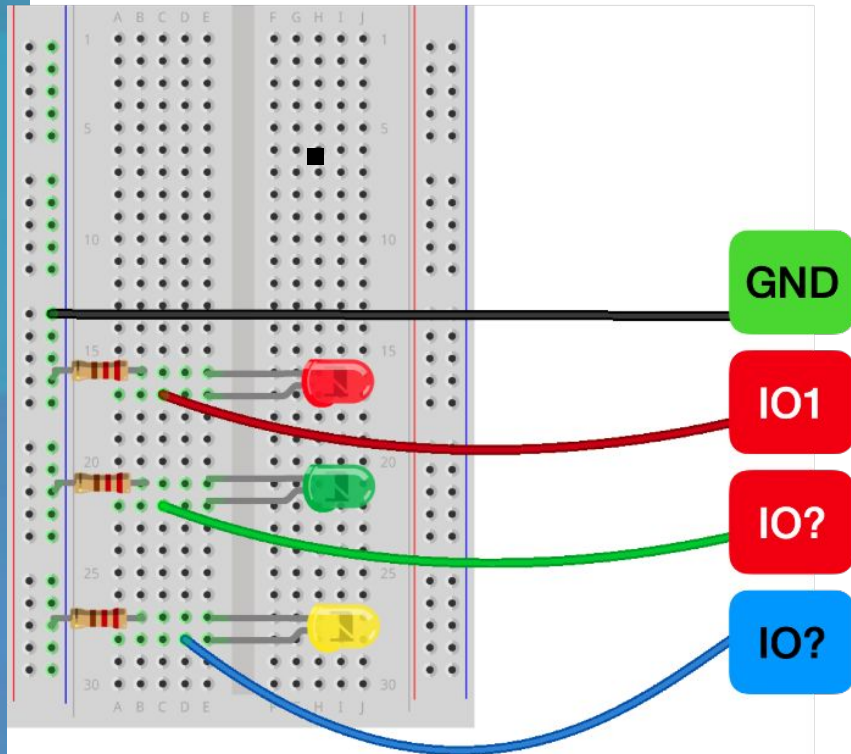
1. เขียน code ทำการเลียน Tact Switch and Buzzer module และต่อวงจรไฟ 3 ดวงสำหรับ Multi_LED

- หากโยกไปทางขวาเสียงจะร้องและไฟติด 2 ดวง
- หากโยกไปทางซ้ายเสียงจะเงียบและไฟติด 1 ดวง
- หากไม่โยกไปทางไหนเลยไฟจะไม่ติด
 - ส่ง video clip เท่านั้น

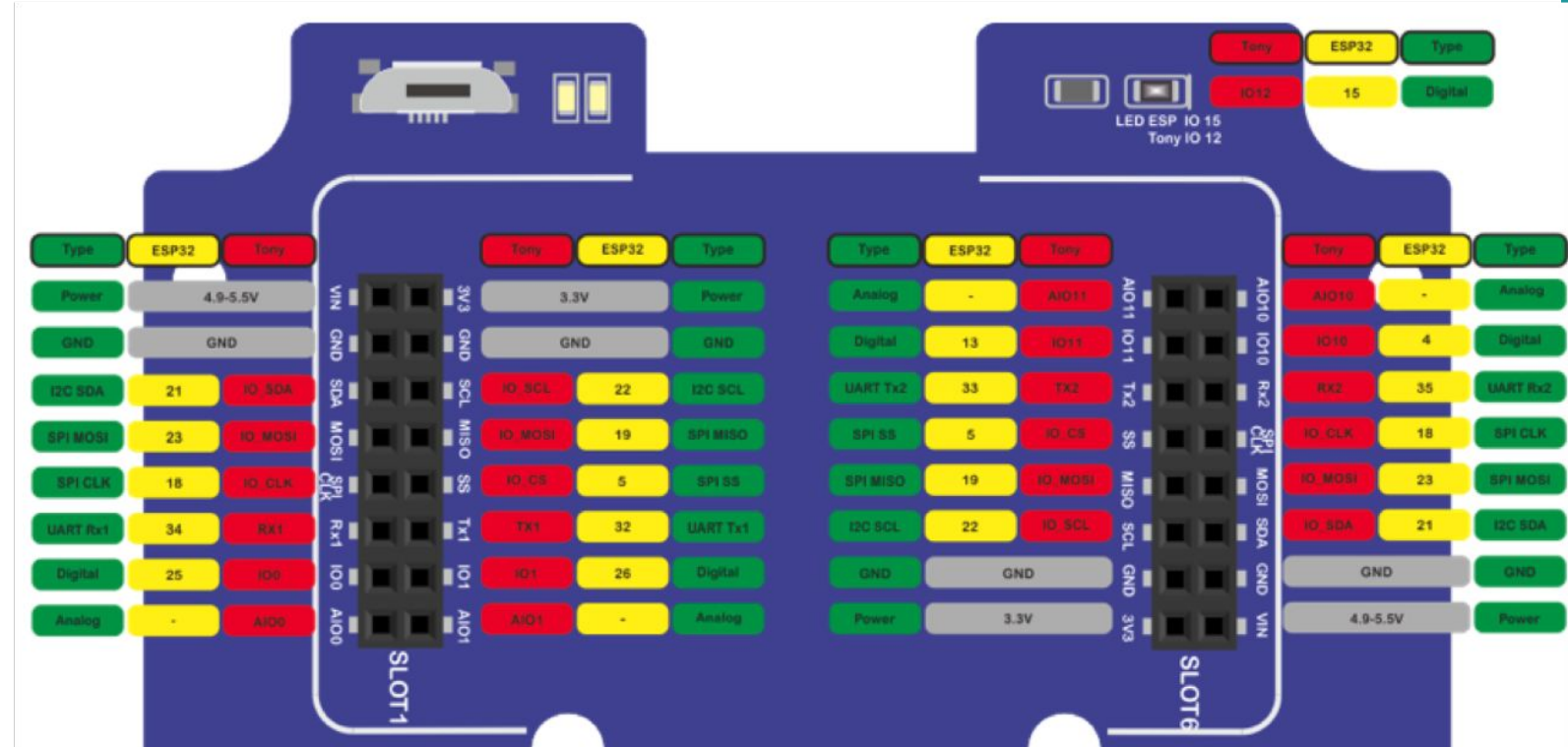
1.



1



fritzing





END Day 3

-