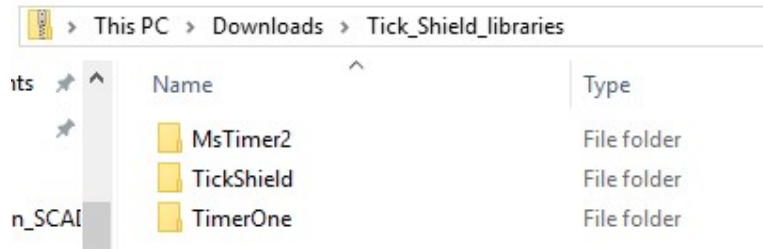


Week04 – Shields and Display

1. TickTock Shield

- http://wiki.seeedstudio.com/wiki/Tick_Tock_Shield_Kit
- Download http://wiki.seeedstudio.com/images/3/31/Tick_Shield_libraries.zip
- Unzip and Copy 3 Folder to “C:\Program Files (x86)\Arduino\libraries”



- ปิดหน้าต่าง Arduino IDE ทั้งหมดแล้วเปิดใหม่

2. ทดสอบ File → Example → TickShield → CharectorFlow

3. ใช้โปรแกรม File → Example → Basic → Blink

- ทดสอบว่า LED RED, RED, Green, Blue คือขาใดระหว่างขา 2, 3, 4, 5
- สเก็ตรูป บอร์ด Tick Tock Shield แล้วระบุขาที่ LED
- ทดสอบขา 6 เป็น Output มันต่อกับอะไร
- ทดสอบโปรแกรม **Program 1 – Tone**
- Exercise 1/5 Sound

4. ใช้โปรแกรม File → Example → Basic → DigitalReadSerial

- ทดสอบว่าปุ่ม MM Up Dw คือขาใดระหว่างขา 9, 10, 11
- สเก็ตรูป บอร์ด Tick Tock Shield แล้วระบุขาที่ Key
- Exercise 2/5 LED From Switch
- Exercise 3/5 LED Move

5. ทดสอบโปรแกรม **Program 2 – Segment Display**

- ลองเปลี่ยนเป็นรหัสนักศึกษาของตัวเอง (ใช้ 4 ตัวท้าย)

6. ทดสอบโปรแกรม Program 3 – Segment Display from Integer

- Exercise 4/5 Segment Run

7. ทดสอบโปรแกรม Program 4 – Segment Display Procedure

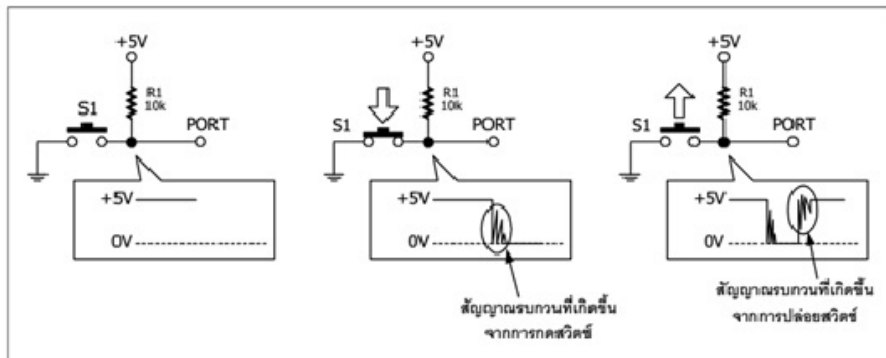
- เป็นการปรับบางส่วนของโปรแกรมให้เป็นโปรแกรมย่อย ทำให้สะดวกเวลาเรียกใช้งาน

8. ทดสอบโปรแกรม Program 5 – Count with Switch

- การเขียนโปรแกรมอ่านค่าจำนวนการกดสวิตช์
- ปัญหาที่พบ คือ การกระเพื่อมของสัญญาณ(Bounce)

<https://www.slideshare.net/yogendra48/arduino-day-1-presentation>

- Exercise 5/5 Counter



Program 1 – Tone - <https://www.arduino.cc/en/Tutorial/Melody>

```
# Demo function: Use the buttons to adjust the pitch of the buzzer sound.
#
# Author: Frankie.Chu
# Date: 23 September, 2012
#
# This library is free software; you can redistribute it and/or
# modify it under the terms of the GNU Lesser General Public
# License as published by the Free Software Foundation; either
# version 2.1 of the License, or (at your option) any later version.
#
# This library is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# Lesser General Public License for more details.
#
# You should have received a copy of the GNU Lesser General Public
# License along with this library; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
#
# Modified record:
#
#-----/

/*If do not want to print any message on the serial monitor,
you can comment out the following line of code like *///#define DEBUG 1*/

#define DEBUG 1
/*----- pin definition of keys-----*/
#define KEY_MENU 11
#define KEY_UP 10
#define KEY_DOWN 9
/*pin definition of the buzzer*/
#define ALARM_BUZZER 6

int tempo = 300;

void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(ALARM_BUZZER, HIGH);
    delayMicroseconds(tone);
    digitalWrite(ALARM_BUZZER, LOW);
    delayMicroseconds(tone);
  }
}

void setup()
{
  #ifdef DEBUG
    Serial.begin(9600);
  #endif
  init_pins();
}

void loop()
{
  static int tone = 1000;
  int key_pressed = scanKey();
  if (KEY_UP == key_pressed)
  {
    /*Every time the KEY_UP key pressed, it will raise the pitch of the */
    /*buzzer sound. */
    if (tone > 1000)
      tone -= 100;
    playTone(tone, tempo);
    #ifdef DEBUG
      Serial.print("tone");
      Serial.println(tone);
    #endif
  }
  else if (KEY_DOWN == key_pressed)
  {
    /*Every time the KEY_DOWN key pressed, it will lower the pitch of the */
    /*buzzer sound. */
    if (tone < 2000)
      tone += 100;
    playTone(tone, tempo);
    #ifdef DEBUG
      Serial.print("tone");
      Serial.println(tone);
    #endif
  }
  /*If a key has been pressed, wait for extra 200ms*/
  if (key_pressed > 0) delay(200);
}

/*Initialization for the keys and the buzzer*/
void init_pins()
{
  /*K1 and K2 are set to be input pins*/
  pinMode(KEY_UP, INPUT);
  digitalWrite(KEY_UP, HIGH);
  pinMode(KEY_DOWN, INPUT);
  digitalWrite(KEY_DOWN, HIGH);
  /*The buzzer is set to be output pin*/
  pinMode(ALARM_BUZZER, OUTPUT);
  digitalWrite(ALARM_BUZZER, LOW);
}

/*Scan keys and get to know which one is pressed*/
/*Return: if no key is pressed, return -1*/
int key_pressed = scanKey()
{
  int pin_number = -1;
  if (digitalRead(KEY_UP) == LOW)
  {
    delay(50);
    if (digitalRead(KEY_UP) == LOW)
    {
      pin_number = KEY_UP;
    }
  }
  #ifdef DEBUG
    Serial.println("KEY_UP is pressed");
  #endif
  }
  else if (digitalRead(KEY_DOWN) == LOW)
  {
    delay(50);
    if (digitalRead(KEY_DOWN) == LOW)
    {
      pin_number = KEY_DOWN;
    }
  }
  #ifdef DEBUG
    Serial.println("KEY_DOWN is pressed");
  #endif
  #endif
  }
  return pin_number;
}
```

Program 2 – Segment Display

```
#include "Wire.h"
#include "TM1636.h"
#include "TickShield.h"

TickShield tickshield;

void setup()
{ tickshield.init();
}

void loop()
{ int8_t DispBuff[4];
  DispBuff[0] = 1;
  DispBuff[1] = 2;
  DispBuff[2] = 3;
  DispBuff[3] = 4;
  tickshield.display(DispBuff);
  while(1);
}
```

Program 3 –Segment Display from Integer

```
#include "Wire.h"
#include "TM1636.h"
#include "TickShield.h"

TickShield tickshield;

void setup()
{ tickshield.init();
}

void loop()
{ int8_t DispBuff[4];
  int temp, Counter = 4567;
  while (1)
  { temp = Counter;
    DispBuff[3] = temp % 10; temp /= 10;
    DispBuff[2] = temp % 10; temp /= 10;
    DispBuff[1] = temp % 10; temp /= 10;
    DispBuff[0] = temp % 10; temp /= 10;
    tickshield.display(DispBuff);
    Counter++;
    delay(500);
  }
}
```

Program 4 – Segment Display Procedure

```
#include "Wire.h"
#include "TM1636.h"
#include "TickShield.h"

TickShield tickshield;

void setup()
{ tickshield.init();
}

void loop()
{ int Counter = 4567;
  while (1)
  { Display7Sement(Counter);
    Counter--;
    delay(500);
  }
}

void Display7Sement(int temp)
{ int8_t DispBuff[4];
  DispBuff[3] = temp % 10; temp /= 10;
  DispBuff[2] = temp % 10; temp /= 10;
  DispBuff[1] = temp % 10; temp /= 10;
  DispBuff[0] = temp % 10; temp /= 10;
  tickshield.display(DispBuff);
}
```

Program 5 – Count with Switch

```
#include "Wire.h"
#include "TM1636.h"
#include "TickShield.h"

TickShield tickshield;
int Counter = 4567;
int KeyUp = 9;

void setup()
{ tickshield.init();
  pinMode(KeyUp, INPUT_PULLUP);
  Display7Sement(Counter);
}

void loop()
{ while (digitalRead(KeyUp) == LOW)
  { delay(20);
    Counter--;
    Display7Sement(Counter);
    while (digitalRead(KeyUp) == LOW);
    delay(20);
  }
}

void Display7Sement(int temp)
{ int8_t DispBuff[4];
  DispBuff[3] = temp % 10; temp /= 10;
  DispBuff[2] = temp % 10; temp /= 10;
  DispBuff[1] = temp % 10; temp /= 10;
  DispBuff[0] = temp % 10; temp /= 10;
  tickshield.display(DispBuff);
}
```

Week04 - Exercise

เลขที่ _____ ID _____ Name _____

1. สเก็ตบอร์ด Tick Tock Shield พร้อมระบุ

1. Exercise 1/5 Sound ➡ อยากได้ Melody Jingle Bells Song

2. Exercise 2/5 LED from Switch ➡ อยากได้โปรแกรมที่

- กด M ให้ LED สีแดง 2 หลอดติด เมื่อปล่อยมือให้ดับ
- กด Up ให้ LED สีเขียวติด เมื่อปล่อยมือให้ดับ
- กด Dw ให้ LED สีน้ำเงินติด เมื่อปล่อยมือให้ดับ

3. Exercise 3/5 LED Move ➡ อยากได้โปรแกรมไฟวิ่ง โดยควบคุมจากการกดปุ่ม

- กด M ไฟกระพริบทั้ง 4 ดวง (หากซ้ำจะแก้ไขเป็นแบบมีเสียงด้วย)
- กด Up ไฟวิ่งจากซ้ายไปขวา
- กด Dw ไฟวิ่งจากขวาไปซ้าย

4. Exercise 4/5 Segment Run ➡ อยากได้โปรแกรมนับเลขจาก 0000 ถึงรหัสนักศึกษาตัวเองแล้วหยุดค้างไว้ (หากซ้ำจะแก้ไขเป็นแล้วหยุดกระพริบรอ)

5. Exercise 5/5 Counter ➡ อยากได้โปรแกรมนับรถ

- กด M ตั้งค่าเป็น 0000
- กด Up นับ +1 < ถ้าเป็น 9999 แล้วจะนับขึ้นอีกไม่ได้ >
- กด Dw นับ -1 < ถ้าเป็น 0 แล้วจะนับลงอีกไม่ได้ >

6. Exercise Challenge ➡ จาก Exercise 2/5 LED from Switch อยากแบบกดแล้วติด ปล่อยมือก็ยังติดอยู่ จนกว่าจะกดอีกครั้งจึงจะดับ จากลักษณะการทำงานแบบเดิมจะเรียก สวิตช์แบบนี้ว่า “กดติด-ปล่อยดับ” และแบบใหม่จะเรียกว่า “กดติด-กดดับ”