

MIDI Controller

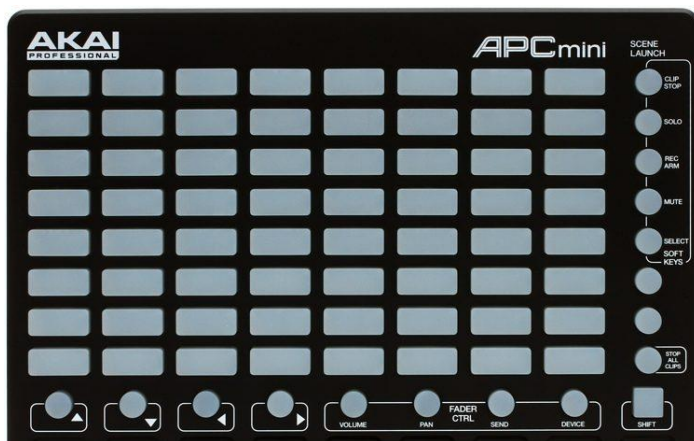
Microprocessor and embedded system

57010227 CHATTARIN JUNYAVACH

57010987 PEEM SRINIKORN

57011524 ATHAWAT SANGKHAM

Background/Reference

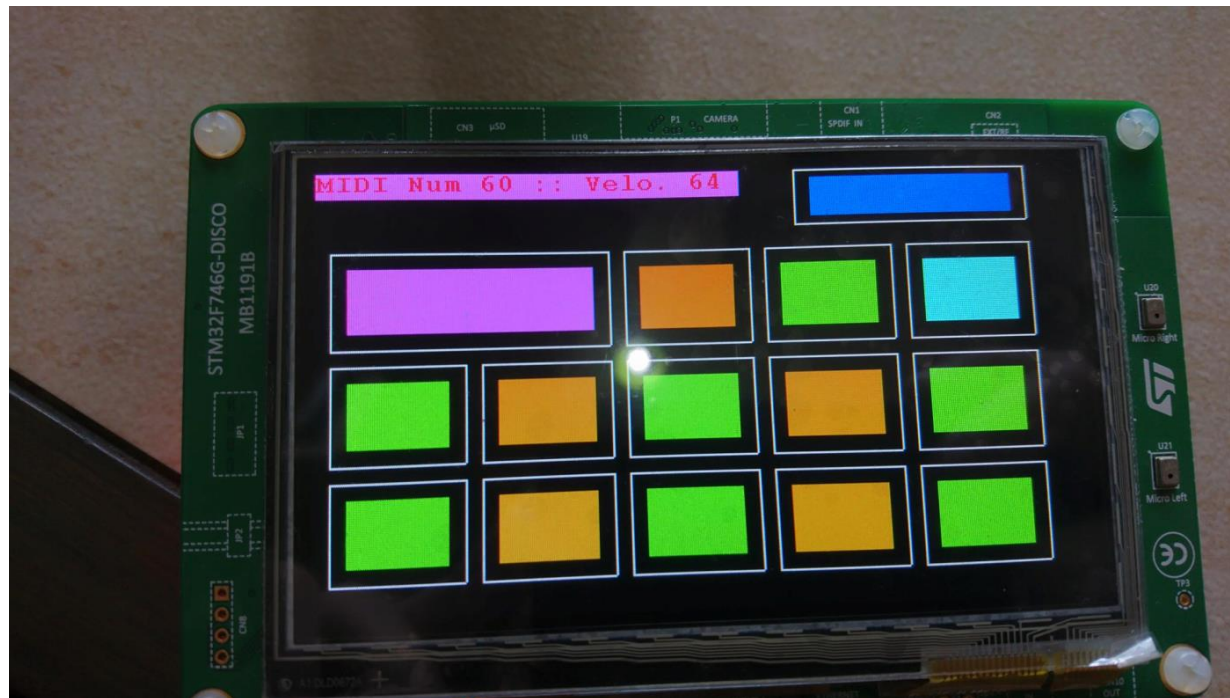


MIDI Controller หรืออุปกรณ์สำหรับการควบคุม MIDI Message โดยลักษณะของชิ้นงานนี้ ได้รับแรงบันดาลใจมาจากเครื่องเล่น MIDI ต่าง ๆ เช่น Lunchpad และ Sequencer เป็นต้น

โดยจะนำความรู้ของวิชา Microprocessor and embedded system มาประยุกต์เพื่อสร้างชิ้นงานที่สำหรับใช้ในการควบคุม MIDI Message โดยใช้หน้าจอ LCD ของบอร์ด STM32 DISCO-F746NG เป็นตัวควบคุมและทำงาน

Overview

ตัวอย่างชิ้นงาน



Mapping

Display

Adjust MIDI Num



Adjust Velocity

MIDI keynote

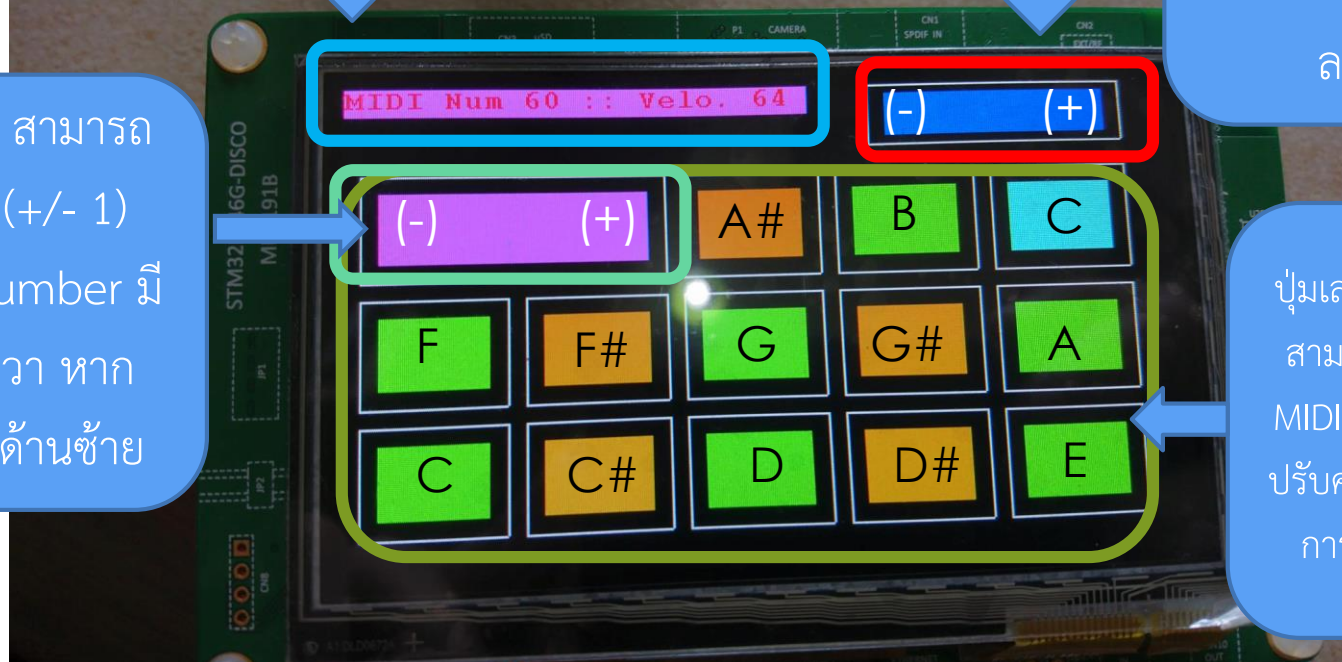
Mapping

แถบสีชมพูหรือ display คือ แถบ
ที่แสดงค่า MIDI Message และ
Velocity

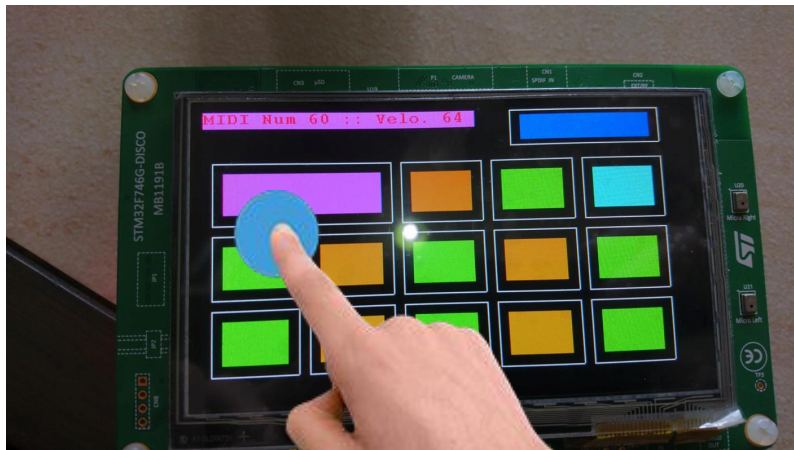
ปุ่มปรับค่า MIDI Message สามารถ
ปรับได้ทีละ 1 step หรือ (+/- 1)
โดยหากต้องการให้ MIDI number มี
ค่ามากขึ้นให้สัมผัสที่ด้านขวา หาก
ต้องการให้ลดลงให้สัมผัสที่ด้านซ้าย

ปุ่มปรับค่า Velocity สามารถปรับได้
ทีละ 1 step หรือ (+/- 1) โดยหาก
ต้องการให้ Velocity มีค่ามากขึ้นให้
สัมผัสที่ด้านขวา หากต้องการให้
ลดลงให้สัมผัสที่ด้านซ้าย

ปุ่มเล่นโน้ต โดยจะมีทั้งหมด 13 โน้ต ซึ่ง
สามารถตั้งค่าโน้ตเริ่มต้นได้โดยปรับค่า
MIDI Number (ไม่มีคีย์ตายตัว สามารถ
ปรับค่าได้ตามความต้องการ) ในภาพเป็น
การยกตัวอย่างวิธีการไล่โน้ตในสเกล



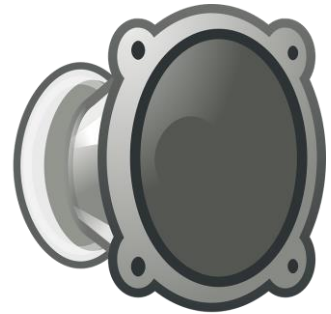
Working



Touch LCD

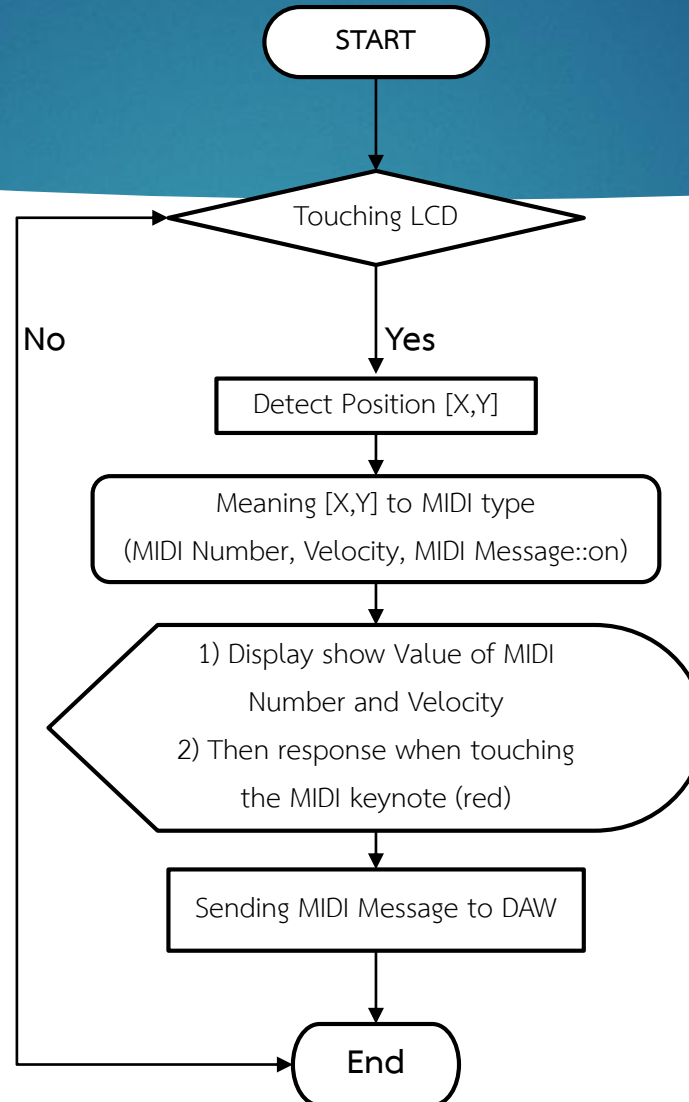


DAW



Output

Flowchart



Coding

- ขั้นตอนแรกคือเรียกใช้งาน library และสร้างปุ่มกดต่าง ๆ เพื่อรองรับการสัมผัสหน้าจอ

```
#include "mbed.h"
#include "USBMIDI.h"
#include "TS_DISCO_F746NG.h"
#include "LCD_DISCO_F746NG.h"

#define FastSpeedInterface 0
#define HighSpeedInterface 1

LCD_DISCO_F746NG lcd;
TS_DISCO_F746NG ts;
USBMIDI midi(FastSpeedInterface);
Serial pc(USBTX, USBRX);
```

```
//////////////////////////////// Pad //////////////////////////////////
lcd.SetTextColor(LCD_COLOR_WHITE); // C note
lcd.DrawRect(10,190, 80,60);
  lcd.SetTextColor(LCD_COLOR_GREEN);
  lcd.FillRect(20,200,60,40);

  lcd.SetTextColor(LCD_COLOR_WHITE); // C# note
  lcd.DrawRect(100,190, 80,60);
    lcd.SetTextColor(LCD_COLOR_ORANGE);
    lcd.FillRect(110,200,60,40);

lcd.SetTextColor(LCD_COLOR_WHITE); // D note
lcd.DrawRect(190,190, 80,60);
  lcd.SetTextColor(LCD_COLOR_GREEN);
  lcd.FillRect(200,200,60,40);

lcd.SetTextColor(LCD_COLOR_WHITE); // D# note
lcd.DrawRect(280,190, 80,60);
  lcd.SetTextColor(LCD_COLOR_ORANGE);
  lcd.FillRect(290,200,60,40);

lcd.SetTextColor(LCD_COLOR_WHITE); // E note
lcd.DrawRect(370,190, 80,60);
  lcd.SetTextColor(LCD_COLOR_GREEN);
  lcd.FillRect(380,200,60,40);
```


Coding

- สร้างเงื่อนไขและตัวแปรตั้งต้นเพื่อรองรับการสัมผัสหน้าจอแบบ 2 นิ้วพร้อมกันได้ และสามารถแสดงค่าที่ปรับแล้วได้

```
int i = 60;
int vol= 64;

while(1) {
    int x,y,x2,y2;

    ts.GetState(&TS_State);
    if ((TS_State.touchDetected))
    {
        // Clear lines corresponding to old touches coordinates
        if (TS_State.touchDetected < prev_nb_touches) {
            for (idx = (TS_State.touchDetected + 1); idx <= 5; idx++) {
                //lcd.ClearStringLine(idx);
            }
        }
        prev_nb_touches = TS_State.touchDetected;

        cleared = 0;

        x = TS_State.touchX[0];
        y = TS_State.touchY[0];
        x2 = TS_State.touchX[1];
        y2 = TS_State.touchY[1];
```

```
        if (x < 80 && x> 10 && y<110 && y>60)
        {
            i--;
            wait(0.1);
        }
        else if (x < 170 && x> 80 && y<110 && y>60)
        {
            i++;
            wait(0.1);
        }

        if(x< 430 && x>365 && y<40 && y>0)
        {
            vol++;
            wait(0.1);
        }

        else if(x<360 && x>300 && y<40 && y>0)
        {
            vol--;
            wait(0.1);
        }

        sprintf((char*)text, "MIDI Num %d :: Velo. %d ", i,vol); //display num
        lcd.DisplayStringAt(0, LINE(0), (uint8_t *)&text, LEFT_MODE);
```

Coding

- ▶ สร้างเงื่อนไขเมื่อสัมผัสหน้าจอแล้วจะทำการส่ง MIDI Message

```
/////note
if( x<80 && x>10 && y<250 && y>190) // C note
{

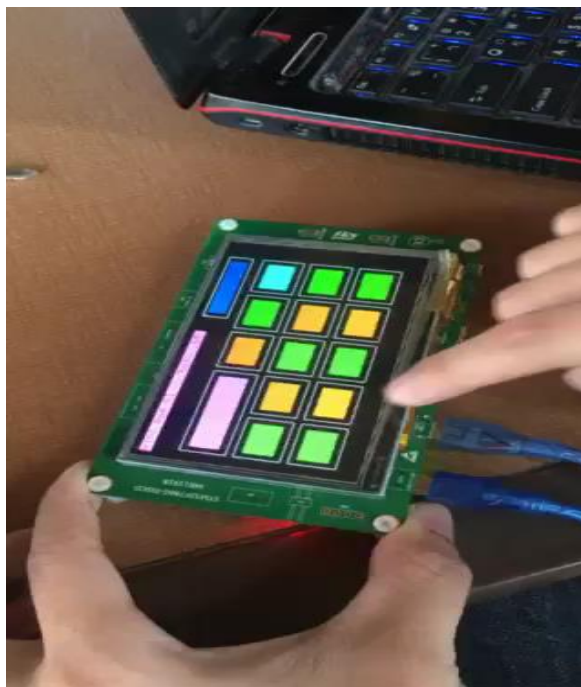
    midi.write(MIDIMessage::NoteOn(i,vol,0)); ///midi send
    lcd.SetTextColor(LCD_COLOR_RED); // display
    lcd.FillRect(20,200,60,40); // display
    if (x2 < 80 && x2> 10 && y2<110 && y2>60) i--; //down note
    else if (x2 < 170 && x2> 80 && y2<110 && y2>60) i++; //up midi
    if(x2< 430 && x2>365 && y2<40 && y2>0) vol++; // up vol
    else if(x2<360 && x2>300 && y2<40 && y2>0)vol--; // down vol
    sprintf((char*)text, "MIDI Num %d :: Velo. %d ", i,vol); //display num
    lcd.DisplayStringAt(0, LINE(0), (uint8_t *)&text, LEFT_MODE); //display num
```

Coding

- ▶ สร้างเงื่อนไขเมื่อไม่สัมผัสหน้าจอแล้วจะทำการส่ง MIDI Message

```
}  
else  
{  
    midi.write(MIDIMessage::AllNotesOff(0));  
    TS_State.touchX[1] = 0;  
    TS_State.touchY[1] = 0;  
    ////////// ex-map /////  
    lcd.SetTextColor(LCD_COLOR_WHITE); // C note  
    lcd.DrawRect(10,190, 80,60);  
    lcd.SetTextColor(LCD_COLOR_GREEN);  
    lcd.FillRect(20,200,60,40);  
}
```

Actual test



► ปัญหาที่พบ

- 1) โน้ต เล่นค้างไม่ได้จะเป็นการ วนลูปซ้ำแทน
- 2) หน้าจอสัมผัสไม่เสถียรและระบบค้างบ่อยเมื่อใช้งานในระยะเวลาที่นาน