**感測元件－期末報告 學生：陳冠瑜**

題目：溫度感測＋數位顯示＋播放音樂

程式碼：

#include <DFMiniMp3.h>

//使用軟體Serial

#include <SoftwareSerial.h>

SoftwareSerial mySerial(11,10); // RX, TX

const int sw = 2500; // 定義播放雨音的長度 2 秒

TM1637 tm1637(CLK,DIO); //宣告顯示晶片涵式庫

DHT dht(DHTPIN, DHTTYPE);

// 宣告 notify class

//

class Mp3Notify;

// 定義類型，其中第一個參數為決定用硬體Serial或軟體Serail

//

typedef DFMiniMp3<SoftwareSerial, Mp3Notify> DfMp3; //如果用UNO板就用軟體Serail

//typedef DFMiniMp3<HardwareSerial, Mp3Notify> DfMp3; //如果用Mega板就用硬體Serail

// 建立mp3物件

//

DfMp3 dfmp3(mySerial);

// 實作notification的類別，可以在不同的事件中，寫入想要進行的動作

// 若沒有特別要進行的處理，這裡不用修改

class Mp3Notify

{

public:

static void PrintlnSourceAction(DfMp3\_PlaySources source, const char\* action)

{

if (source & DfMp3\_PlaySources\_Sd)

{

Serial.print("SD Card, ");

}

if (source & DfMp3\_PlaySources\_Usb)

{

Serial.print("USB Disk, ");

}

if (source & DfMp3\_PlaySources\_Flash)

{

Serial.print("Flash, ");

}

Serial.println(action);

}

static void OnError([[maybe\_unused]] DfMp3& mp3, uint16\_t errorCode)

{

// 錯誤訊息

Serial.println();

Serial.print("Com Error ");

Serial.println(errorCode);

}

static void OnPlayFinished([[maybe\_unused]] DfMp3& mp3, [[maybe\_unused]] DfMp3\_PlaySources source, uint16\_t track)

{

Serial.print("Play finished for #");

Serial.println(track);

}

static void OnPlaySourceOnline([[maybe\_unused]] DfMp3& mp3, DfMp3\_PlaySources source)

{

PrintlnSourceAction(source, "online");

}

static void OnPlaySourceInserted([[maybe\_unused]] DfMp3& mp3, DfMp3\_PlaySources source)

{

PrintlnSourceAction(source, "inserted");

}

static void OnPlaySourceRemoved([[maybe\_unused]] DfMp3& mp3, DfMp3\_PlaySources source)

{

PrintlnSourceAction(source, "removed");

}

};

int dig1 = 0;

int dig2 = 0;

int dig3 = 0;

int dig4 = 0;

void setup() {

Serial.begin(9600);

dht.begin();

tm1637.init();

tm1637.set(BRIGHT\_TYPICAL);

Serial.begin(9600); //啟用監控視窗

Serial.println("initializing...");

dfmp3.begin(); //開始使用DFPlayer模組

//重置DFPlayer模組，會聽到"波"一聲

dfmp3.reset();

//音量控制，0~30

//uint16\_t volume = dfmp3.getVolume();

//Serial.print("volume ");

//Serial.println(volume);

dfmp3.setVolume(20);

//volume = dfmp3.getVolume();

//Serial.print("volume ");

//Serial.println(volume);

//取得所有MP3檔的總數

//uint16\_t count = dfmp3.getTotalTrackCount(DfMp3\_PlaySource\_Sd);

//Serial.print("files ");

//Serial.println(count);

Serial.println("Play Mp3 Voice...");

// 播放MP3，參數就是第N首。如要播第一首MP3，參數就是1，第二首就是2

dfmp3.playMp3FolderTrack(1);

delay(sw);

dfmp3.playMp3FolderTrack(2);

delay(sw);

dfmp3.playMp3FolderTrack(3);

delay(sw);

}

void loop()

{

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(t) || isnan(h)) {

Serial.println("Failed to read from DHT");

}

else {

//int dig1 = 0, dig2 = 0, dig3 = 0, dig4 = 0;

dig4 = (t / 10);

dig3 = t - ( dig4 \* 10);

dig2 = (h / 10);

dig1 = h - ( dig2 \* 10 );

tm1637.display(0,dig4); //千位數 千位 百位 = 溫度

tm1637.display(1,dig3); //百位數

tm1637.display(2,dig2); //十位數 十位 個位 = 濕度

tm1637.display(3,dig1); //個位數

dfmp3.playMp3FolderTrack(1);

delay(sw);

dfmp3.playMp3FolderTrack(2);

delay(sw);

dfmp3.playMp3FolderTrack(3);

delay(sw);

}

}