**電通二甲微處理器實驗 實驗結報**

|  |  |  |  |
| --- | --- | --- | --- |
| **實驗名稱** | Lab06-音樂教室 | | |
| **組別** | **25** | **組員** | 04050033梁艾蓉 / 04050475 胡皓雯 |

1. **實驗目的**

如何使 Arduino 發出特定旋律的聲音?

1. Arduino 接喇叭如何接線?

2. 如何使用 tone library?

3. 如何演奏一段音樂?

4. 如何使用 4x4 鍵盤演奏音樂?

5. 如何發報摩斯電碼?

1. **實驗步驟**

\*Arduino 演奏一段特定的音樂, 旋律不得是小蜜蜂

\* 使用 4x4 鍵盤演奏音樂

\* 發報摩斯電碼

1. **程式碼**

**音階**

**void playTone(int tone, int duration) {**

**for (long i =0; i < duration \* 1000L; i += tone \* 2) {**

**digitalWrite(speakerPin, HIGH);**

**delayMicroseconds(tone);**

**digitalWrite(speakerPin, LOW);**

**delayMicroseconds(tone);**

**}**

**}**

**節拍**

**void playNote(char note, int duration) {**

**char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };**

**int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };**

**// 利用比對字元的方式來播放某一個音階**

**for (int i = 0; i < 8; i++) {**

**// 將比對到的音階高電位時間長度傳送給playTone函式播放**

**if (names[i] == note) playTone(tones[i], duration);**

**}**

**}**

**使用 4x4 鍵盤演奏音樂**

**void loop() {**

**x = xxxxxxxxx; // 如何透過 4x4 鍵盤讀出0 -15?**

**if (x!= NOKEY) { // 有按下按鍵**

**tone(8, tune[x]);**

**delay(0.5);**

**noTone(8);**

**}**

**}**

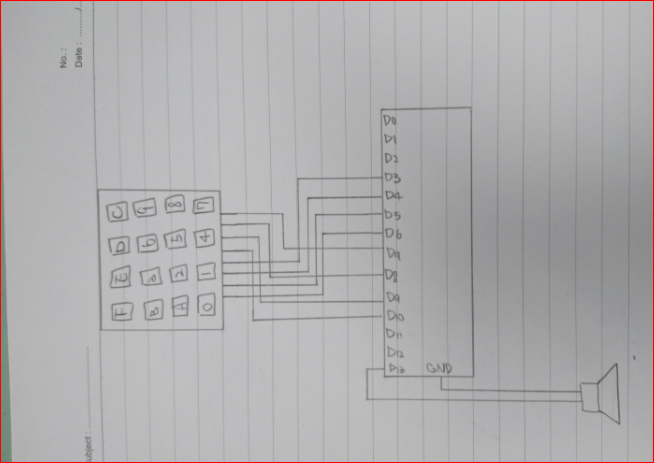
1. **實驗結果及分析**
2. 喇叭一端接Arduino腳位，另一端接地
3. 程式碼中可以用#define在程式開頭定義音高代碼，這樣比直接用頻率好
4. Tone程式庫可以輸出指定頻率的聲音和持續時間

* tone(輸出腳位, 頻率, 持續時間);

1. 摩斯密碼的delay時間設定為：每一個電碼隔0.1s，字母間隔0.3s，字間隔0.7s
2. **心得討論**

此次實驗我們學習到如何將轉換為音階的樂譜，設定在程式碼中，讓Arduino演奏出此樂譜；還有透過接上4x4鍵盤，讓各個按鍵可以發出特定的音階，就像是自製了一台簡易的鋼琴。最後，我們還學會了摩斯密碼的設定方法，讓Arduino讀取序列埠的輸入，並以摩斯電碼發報。

1. **修正電路圖**



1. **修正程式碼**

**演奏加沃特舞曲**

**#define NOTE\_ 0**

**#define NOTE\_B0 31**

**#define NOTE\_C1 33**

**#define NOTE\_CS1 35**

**#define NOTE\_D1 37**

**#define NOTE\_DS1 39**

**#define NOTE\_E1 41**

**#define NOTE\_F1 44**

**#define NOTE\_FS1 46**

**#define NOTE\_G1 49**

**#define NOTE\_GS1 52**

**#define NOTE\_A1 55**

**#define NOTE\_AS1 58**

**#define NOTE\_B1 62**

**#define NOTE\_C2 65**

**#define NOTE\_CS2 69**

**#define NOTE\_D2 73**

**#define NOTE\_DS2 78**

**#define NOTE\_E2 82**

**#define NOTE\_F2 87**

**#define NOTE\_FS2 93**

**#define NOTE\_G2 98**

**#define NOTE\_GS2 104**

**#define NOTE\_A2 110**

**#define NOTE\_AS2 117**

**#define NOTE\_B2 123**

**#define NOTE\_C3 131**

**#define NOTE\_CS3 139**

**#define NOTE\_D3 147**

**#define NOTE\_DS3 156**

**#define NOTE\_E3 165**

**#define NOTE\_F3 175**

**#define NOTE\_FS3 185**

**#define NOTE\_G3 196**

**#define NOTE\_GS3 208**

**#define NOTE\_A3 220**

**#define NOTE\_AS3 233**

**#define NOTE\_B3 247**

**#define NOTE\_C4 262**

**#define NOTE\_CS4 277**

**#define NOTE\_D4 294**

**#define NOTE\_DS4 311**

**#define NOTE\_E4 330**

**#define NOTE\_F4 349**

**#define NOTE\_FS4 370**

**#define NOTE\_G4 392**

**#define NOTE\_GS4 415**

**#define NOTE\_A4 440**

**#define NOTE\_AS4 466**

**#define NOTE\_B4 494**

**#define NOTE\_C5 523**

**#define NOTE\_CS5 554**

**#define NOTE\_D5 587**

**#define NOTE\_DS5 622**

**#define NOTE\_E5 659**

**#define NOTE\_F5 698**

**#define NOTE\_FS5 740**

**#define NOTE\_G5 784**

**#define NOTE\_GS5 831**

**#define NOTE\_A5 880**

**#define NOTE\_AS5 932**

**#define NOTE\_B5 988**

**#define NOTE\_C6 1047**

**#define NOTE\_CS6 1109**

**#define NOTE\_D6 1175**

**#define NOTE\_DS6 1245**

**#define NOTE\_E6 1319**

**#define NOTE\_F6 1397**

**#define NOTE\_FS6 1480**

**#define NOTE\_G6 1568**

**#define NOTE\_GS6 1661**

**#define NOTE\_A6 1760**

**#define NOTE\_AS6 1865**

**#define NOTE\_B6 1976**

**#define NOTE\_C7 2093**

**#define NOTE\_CS7 2217**

**#define NOTE\_D7 2349**

**#define NOTE\_DS7 2489**

**#define NOTE\_E7 2637**

**#define NOTE\_F7 2794**

**#define NOTE\_FS7 2960**

**#define NOTE\_G7 3136**

**#define NOTE\_GS7 3322**

**#define NOTE\_A7 3520**

**#define NOTE\_AS7 3729**

**#define NOTE\_B7 3951**

**#define NOTE\_C8 4186**

**#define NOTE\_CS8 4435**

**#define NOTE\_D8 4699**

**#define NOTE\_DS8 4978**

**int melody[] = { NOTE\_A6, NOTE\_B6, NOTE\_A6,NOTE\_FS6, NOTE\_G6, NOTE\_A6, NOTE\_G6,NOTE\_E6, NOTE\_D6, NOTE\_D7, NOTE\_D6,NOTE\_,**

**NOTE\_G6, NOTE\_A6, NOTE\_G6,NOTE\_E6, NOTE\_FS6, NOTE\_G6, NOTE\_FS6,NOTE\_D6, NOTE\_E6, NOTE\_A6, NOTE\_A5,NOTE\_,**

**NOTE\_A6, NOTE\_B6, NOTE\_A6,NOTE\_FS6, NOTE\_G6, NOTE\_A6, NOTE\_G6,NOTE\_E6, NOTE\_D6, NOTE\_D7, NOTE\_D6,NOTE\_,**

**NOTE\_FS6, NOTE\_D6, NOTE\_B5,NOTE\_D6, NOTE\_B5, NOTE\_GS4, NOTE\_A5,NOTE\_A6, NOTE\_A5, NOTE\_,**

**NOTE\_A6, NOTE\_B6, NOTE\_A6,NOTE\_FS6, NOTE\_G6, NOTE\_A6, NOTE\_G6,NOTE\_E6, NOTE\_D6, NOTE\_D7, NOTE\_D6,NOTE\_,**

**NOTE\_G6, NOTE\_A6, NOTE\_G6,NOTE\_E6, NOTE\_FS6, NOTE\_G6, NOTE\_FS6,NOTE\_D6, NOTE\_E6, NOTE\_A6, NOTE\_A5,NOTE\_,**

**NOTE\_A6, NOTE\_B6, NOTE\_A6,NOTE\_FS6, NOTE\_G6, NOTE\_A6, NOTE\_G6,NOTE\_E6, NOTE\_D6, NOTE\_D7, NOTE\_D6,NOTE\_,**

**NOTE\_FS6, NOTE\_D6, NOTE\_B5,NOTE\_D6, NOTE\_B5, NOTE\_GS4, NOTE\_A5,NOTE\_A6, NOTE\_A5, NOTE\_,**

**NOTE\_E6, NOTE\_G6, NOTE\_FS6,NOTE\_A6, NOTE\_G6, NOTE\_FS5, NOTE\_E6,NOTE\_D6, NOTE\_CS6, NOTE\_E6, NOTE\_G6,NOTE\_,**

**NOTE\_FS6, NOTE\_A6, NOTE\_G6,NOTE\_B6, NOTE\_A6, NOTE\_G6, NOTE\_FS6,NOTE\_E6, NOTE\_D6, NOTE\_FS6, NOTE\_A6,NOTE\_,**

**NOTE\_B6, NOTE\_A6, NOTE\_A6,NOTE\_G6, NOTE\_G6, NOTE\_FS6, NOTE\_FS6,NOTE\_E6, NOTE\_E6, NOTE\_G6, NOTE\_B6,NOTE\_,**

**NOTE\_A6, NOTE\_FS6, NOTE\_CS6,NOTE\_D6, NOTE\_G6, NOTE\_E6, NOTE\_B5,NOTE\_CS6, NOTE\_D6, NOTE\_E7, NOTE\_D6,NOTE\_};**

**int noteDurations[] = { 8,8,8,8,8,8,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4,**

**4,8,8,4,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4,**

**4,8,8,4,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4,**

**8,8,8,8,8,8,8,8,4,4,4,4 };**

**void play(int \*melody, int \*noteDurations, int num){**

**for(int note = 0; note < num; note++){**

**int noteDuration = 3000 / noteDurations[note];**

**tone(8, melody[note], noteDuration);**

**delay(noteDuration \* 0.75);**

**}**

**}**

**void setup(){**

**}**

**void loop(){**

**play(melody, noteDurations, sizeof(melody) / sizeof(int));**

**delay(2000);**

**}**

**Arduino 利用4x4 keyboard 演奏一段音樂**

**#define do1 261**

**#define re1 294**

**#define me1 329**

**#define fa1 349**

**#define so1 392**

**#define la1 440**

**#define si1 493**

**#define do2 523**

**#define re2 587**

**#define me2 659**

**#define fa2 698**

**#define so2 784**

**#define la2 880**

**#define si2 988**

**#define do3 1046**

**#define re3 1175**

**int tune[] = {do1,re1,me1,fa1,so1,la1,si1,do2,re2,me2,fa2,so2,la2,si2,do3,re3};**

**#include<Keypad.h>**

**const byte KEY\_ROWS = 4;**

**const byte KEY\_COLS = 4; // 定義 Keypad 的按鍵**

**char keys[KEY\_ROWS][KEY\_COLS] = {**

**{'F', 'E', 'D', 'C'},**

**{'B', '3', '6', '9'},**

**{'A', '2', '5', '8'},**

**{'0', '1', '4', '7'}}; // 定義 Keypad 連到 Arduino 的接腳**

**byte colPins[KEY\_COLS] = {6,5,4,3}; // 按鍵模組，行1~4接腳**

**byte rowPins[KEY\_ROWS] = {10,9,8,7}; // 按鍵模組，列1~4接腳**

**Keypad keypad = Keypad(makeKeymap(keys), rowPins,colPins,KEY\_ROWS, KEY\_COLS ); // 建立 Keypad 物件**

**void setup(){**

**Serial.begin(9600);**

**}**

**void loop(){**

**char key = keypad.getKey(); //讀取 Keypad**

**if(key != NO\_KEY) {**

**if(key>='A'&&key<='F') {**

**Serial.println(key);**

**tone(13, tune[key-'A'+10]);**

**delay(100);**

**noTone(13);**

**}**

**else {**

**Serial.println(key);**

**tone(13, tune[key-'0']);**

**delay(100);**

**noTone(13);**

**}**

**}**

**}**

**摩斯密碼**

**char \*morse[]={**

**"01","1000","1010","100", //A~D**

**"0","0010","110","0000", //E~H**

**"00","0111","101","0100", //I~L**

**"11","10","111","0110", //M~P**

**"1101","010","000","1", //Q~T**

**"001","0001","11","1001", //U~X**

**"1011","1100" //Y~Z**

**};**

**const byte Buzzer=5;**

**char chr,index;**

**char \*ptr;**

**void setup() {**

**pinMode(Buzzer,OUTPUT);**

**Serial.begin(9600);**

**}**

**void loop() {**

**if( Serial.available() ){**

**chr=Serial.read();**

**Serial.println(chr);**

**if( (chr-'A')>=0 && (chr-'Z')<=0 ){**

**index=chr-'A';**

**ptr=morse[index];**

**while(\*ptr!='\0'){**

**if(\*ptr=='0'){**

**tone(Buzzer,440,100);**

**delay(100);**

**}**

**else{**

**tone(Buzzer,440,300);**

**delay(300);**

**}**

**ptr++;**

**delay(100);**

**}**

**delay(300);**

**}**

**else**

**delay(700);**

**}**

**}**