**電通二乙微處理器實驗 實驗結報**

|  |  |  |  |
| --- | --- | --- | --- |
| **實驗名稱** | **音樂教室** | | |
| **組別** |  | **組員** | **李嘉誠 05052446** |

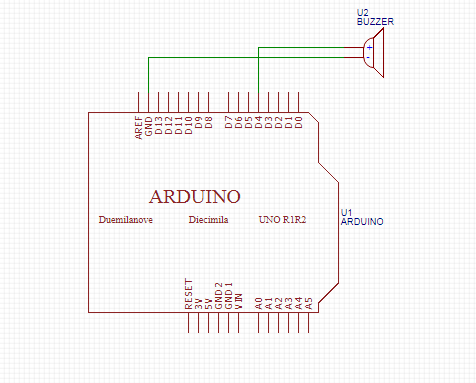
1. **實驗目的**

* 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**

**電路圖**

****

**程式碼**

#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

#define NOTE\_D0 -1

#define NOTE\_D1 294

#define NOTE\_D2 330

#define NOTE\_D3 350

#define NOTE\_D4 393

#define NOTE\_D5 441

#define NOTE\_D6 495

#define NOTE\_D7 556

#define NOTE\_DL1 147

#define NOTE\_DL2 165

#define NOTE\_DL3 175

#define NOTE\_DL4 196

#define NOTE\_DL5 221

#define NOTE\_DL6 248

#define NOTE\_DL7 278

#define NOTE\_DH1 589

#define NOTE\_DH2 661

#define NOTE\_DH3 700

#define NOTE\_DH4 786

#define NOTE\_DH5 882

#define NOTE\_DH6 990

#define NOTE\_DH7 112

#define WHOLE 1

#define HALF 0.5

#define QUARTER 0.25

#define EIGHTH 0.25

#define SIXTEENTH 0.625

int tune[] =

{

NOTE\_D0,NOTE\_D0,NOTE\_D0,NOTE\_D6,NOTE\_D7,NOTE\_DH1,NOTE\_D7,NOTE\_DH1,NOTE\_DH3,NOTE\_D7,NOTE\_D7,NOTE\_D7,NOTE\_D3,NOTE\_D3,

NOTE\_D6,NOTE\_D5,NOTE\_D6,NOTE\_DH1,NOTE\_D5,NOTE\_D5,NOTE\_D5,NOTE\_D3,NOTE\_D4,NOTE\_D3,NOTE\_D4,NOTE\_DH1,

NOTE\_D3,NOTE\_D3,NOTE\_D0,NOTE\_DH1,NOTE\_DH1,NOTE\_DH1,NOTE\_D7,NOTE\_D4,NOTE\_D4,NOTE\_D7,NOTE\_D7,NOTE\_D7,NOTE\_D0,NOTE\_D6,NOTE\_D7,

NOTE\_DH1,NOTE\_D7,NOTE\_DH1,NOTE\_DH3,NOTE\_D7,NOTE\_D7,NOTE\_D7,NOTE\_D3,NOTE\_D3,NOTE\_D6,NOTE\_D5,NOTE\_D6,NOTE\_DH1,

NOTE\_D5,NOTE\_D5,NOTE\_D5,NOTE\_D2,NOTE\_D3,NOTE\_D4,NOTE\_DH1,NOTE\_D7,NOTE\_D7,NOTE\_DH1,NOTE\_DH1,NOTE\_DH2,NOTE\_DH2,NOTE\_DH3,NOTE\_DH1,NOTE\_DH1,NOTE\_DH1,

NOTE\_DH1,NOTE\_D7,NOTE\_D6,NOTE\_D6,NOTE\_D7,NOTE\_D5,NOTE\_D6,NOTE\_D6,NOTE\_D6,NOTE\_DH1,NOTE\_DH2,NOTE\_DH3,NOTE\_DH2,NOTE\_DH3,NOTE\_DH5,

NOTE\_DH2,NOTE\_DH2,NOTE\_DH2,NOTE\_D5,NOTE\_D5,NOTE\_DH1,NOTE\_D7,NOTE\_DH1,NOTE\_DH3,NOTE\_DH3,NOTE\_DH3,NOTE\_DH3,NOTE\_DH3,

NOTE\_D6,NOTE\_D7,NOTE\_DH1,NOTE\_D7,NOTE\_DH2,NOTE\_DH2,NOTE\_DH1,NOTE\_D5,NOTE\_D5,NOTE\_D5,NOTE\_DH4,NOTE\_DH3,NOTE\_DH2,NOTE\_DH1,

NOTE\_DH3,NOTE\_DH3,NOTE\_DH3,NOTE\_DH3,NOTE\_DH6,NOTE\_DH6,NOTE\_DH5,NOTE\_DH5,NOTE\_DH3,NOTE\_DH2,NOTE\_DH1,NOTE\_DH1,NOTE\_D0,NOTE\_DH1,

NOTE\_DH2,NOTE\_DH1,NOTE\_DH2,NOTE\_DH2,NOTE\_DH5,NOTE\_DH3,NOTE\_DH3,NOTE\_DH3,NOTE\_DH3,NOTE\_DH6,NOTE\_DH6,NOTE\_DH5,NOTE\_DH5,

NOTE\_DH3,NOTE\_DH2,NOTE\_DH1,NOTE\_DH1,NOTE\_D0,NOTE\_DH1,NOTE\_DH2,NOTE\_DH1,NOTE\_DH2,NOTE\_DH2,NOTE\_D7,NOTE\_D6,NOTE\_D6,NOTE\_D6,NOTE\_D6,NOTE\_D7

};//这部分就是整首曲子的音符部分，用了一个序列定义为tune，整数

float duration[]=

{

1,1,1,0.5,0.5, 1+0.5,0.5,1,1, 1,1,1,0.5,0.5,

1+0.5,0.5,1,1, 1,1,1,1, 1+0.5,0.5,1,1,

1,1,0.5,0.5,0.5,0.5, 1+0.5,0.5,1,1, 1,1,1,0.5,0.5,

1+0.5,0.5,1,1, 1,1,1,0.5,0.5, 1+0.5,0.5,1,1,

1,1,1,0.5,0.5, 1,0.5,0.25,0.25,0.25,0.5, 0.5,0.5,0.5,0.25,0.5,1,

0.5,0.5,0.5,0.5,1,1, 1,1,1,0.5,0.5, 1+0.5,0.5,1,1,

1,1,1,0.5,0.5, 1.5,0.5,1,1, 1,1,1,1,

0.5,0.5,1,1,0.5,0.5, 1.5,0.25,0.5,1, 1,1,1,1,

1,1,1,1, 1,1,1,1, 0.5,0.5,1,1,0.5,0.5,

1,0.5,0.5,1,1, 1,1,1,1, 1,1,1,1,

0.5,0.5,1,1,0.5,0.5, 1,0.5,0.25,0.5,1, 1,1,1,0.5,0.5

};//这部分是整首曲子的节拍部分，也定义个序列duration，浮点（数组的个数和前面音符的个数是一样的，一一对应么）

int length;//这里定义一个变量，后面用来表示共有多少个音符

int tonePin=5;//蜂鸣器的pin

void setup()

{

pinMode(tonePin,OUTPUT);//设置蜂鸣器的pin为输出模式

length = sizeof(tune)/sizeof(tune[0]);//这里用了一个sizeof函数， 可以查出tone序列里有多少个音符

}

void loop()

{

for(int x=0;x<length;x++)//循环音符的次数

{

tone(tonePin,tune[x]);//此函数依次播放tune序列里的数组，即每个 音符

delay(400\*duration[x]);//每个音符持续的时间，即节拍duration，是调整时间的越大，曲子速度越慢，越小曲子速度越快，自己掌握吧

noTone(tonePin);//停止当前音符，进入下一音符

}

delay(5000);//等待5秒后，循环重新开始

}

**2.實驗步驟**

* **使用 4x4 鍵盤演奏音樂**

**程式碼**

#include <Keypad.h>

#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

#define melodyPin 3

const byte ROWS = 4; // 4 Rows

const byte COLS = 4; // 4 Columns

int duration = 500;

// 定義 Keypad 的按鍵

char keys[ROWS][COLS] = {

{'7', '8', '9', 'C'},

{'4', '5', '6', 'D'},

{'1', '2', '3', 'E'},

{'0', 'A', 'B', 'F'}};

// 定義 Keypad 連到 Arduino 的接腳

byte rowPins[ROWS] = {10,11,12,13};

// 連到 Keypad 的 4 個 Rows

byte colPins[COLS] = {6,7,8,9};

// 連到 Keypad 的 4 個 Columns

// 建立 Keypad 物件

Keypad keypad = Keypad( makeKeymap(keys), rowPins,colPins, ROWS, COLS );

void setup(){

Serial.begin(9600);

}

void loop(){

// 讀取 Keypad 的輸入

char key = keypad.getKey();

// NO\_KEY 代表沒有按鍵被按下

if (key != NO\_KEY){ // 假如有按鍵被按下，就印出按鍵對應的字元

switch (key) {

case '1':

tone(3, NOTE\_C5, duration);

break;

case '2':

tone(3, NOTE\_D5, duration);

break;

case '3':

tone(3, NOTE\_E5, duration);

break;

case '4':

tone(3, NOTE\_F5, duration);

break;

case '5':

tone(3, NOTE\_G5, duration);

break;

case '6':

tone(3, NOTE\_A5, duration);

break;

case '7':

tone(3, NOTE\_B5, duration);

break;

case '8':

tone(3, NOTE\_C4, duration);

break;

case '9':

tone(3, NOTE\_D4, duration);

break;

case 'A':

tone(3, NOTE\_E4, duration);

break;

case 'B':

tone(3, NOTE\_F4, duration);

break;

case 'C':

tone(3, NOTE\_G4, duration);

break;

case 'D':

tone(3, NOTE\_A4, duration);

break;

case 'E':

tone(3, NOTE\_B4, duration);

break;

default:

tone(3, NOTE\_C8, duration);

}

Serial.println(key);

}

}

**3.實驗步驟**

* **發報摩斯電碼**

**程式碼**

#define Do 440

const byte audio = 3;

int dotLen = 150;

int dashLen = 700;

int incomingByte = 0; // 用來儲存收進來的 data byte

void setup() {

Serial.begin(9600);

}

void loop() {

// 檢查是否有資料可供讀取

if (Serial.available() > 0) {

// 讀取進來的 byte

incomingByte = Serial.read();

// 印出收到的資料

Serial.println(incomingByte, DEC);

GetChar(incomingByte);

delay(800);

}

}

void MorseDot() {

tone(audio, Do, dotLen); // start playing a tone

delay(500); // hold in this position

}

// DASH

void MorseDash() {

tone(audio, Do, dashLen); // start playing a tone

delay(800); // hold in this position

}

void GetChar(char tmpChar) {

switch (tmpChar) {

case 'a':

MorseDot();

MorseDash();

break;

case 'b':

MorseDash();

MorseDot();

MorseDot();

MorseDot();

break;

case 'c':

MorseDash();

MorseDot();

MorseDash();

MorseDot();

break;

case 'd':

MorseDash();

MorseDash();

MorseDot();

break;

case 'e':

MorseDot();

break;

case 'f':

MorseDot();

MorseDot();

MorseDash();

MorseDot();

break;

case 'g':

MorseDash();

MorseDash();

MorseDot();

break;

case 'h':

MorseDot();

MorseDot();

MorseDot();

MorseDot();

break;

case 'i':

MorseDot();

MorseDot();

break;

case 'j':

MorseDot();

MorseDash();

MorseDash();

MorseDash();

break;

case 'k':

MorseDash();

MorseDot();

MorseDash();

break;

case 'l':

MorseDot();

MorseDash();

MorseDot();

MorseDot();

break;

case 'm':

MorseDash();

MorseDash();

break;

case 'n':

MorseDash();

MorseDot();

break;

case 'o':

MorseDash();

MorseDash();

MorseDash();

break;

case 'p':

MorseDot();

MorseDash();

MorseDash();

MorseDot();

break;

case 'q':

MorseDash();

MorseDash();

MorseDot();

MorseDash();

break;

case 'r':

MorseDot();

MorseDash();

MorseDot();

break;

case 's':

MorseDot();

MorseDot();

MorseDot();

break;

case 't':

MorseDash();

break;

case 'u':

MorseDot();

MorseDot();

MorseDash();

break;

case 'v':

MorseDot();

MorseDot();

MorseDot();

MorseDash();

break;

case 'w':

MorseDot();

MorseDash();

MorseDash();

break;

case 'x':

MorseDash();

MorseDot();

MorseDot();

MorseDash();

break;

case 'y':

MorseDash();

MorseDot();

MorseDash();

MorseDash();

break;

case 'z':

MorseDash();

MorseDash();

MorseDot();

MorseDot();

break;

case '0':

MorseDash();

MorseDash();

MorseDash();

MorseDash();

MorseDash();

break;

case '1':

MorseDot();

MorseDash();

MorseDash();

MorseDash();

MorseDash();

break;

case '2':

MorseDot();

MorseDot();

MorseDash();

MorseDash();

MorseDash();

break;

case '3':

MorseDot();

MorseDot();

MorseDot();

MorseDash();

MorseDash();

break;

case '4':

MorseDot();

MorseDot();

MorseDot();

MorseDot();

MorseDash();

break;

case '5':

MorseDot();

MorseDot();

MorseDot();

MorseDot();

MorseDot();

break;

case '6':

MorseDash();

MorseDot();

MorseDot();

MorseDot();

MorseDot();

break;

case '7':

MorseDash();

MorseDash();

MorseDot();

MorseDot();

MorseDot();

break;

case '8':

MorseDash();

MorseDash();

MorseDash();

MorseDot();

MorseDot();

break;

case '9':

MorseDash();

MorseDash();

MorseDash();

MorseDash();

MorseDot();

break;

default:

break;

}

}

1. **心得討論**

這次實驗還挺好玩的,感覺像做了一架簡陋版的鋼琴一樣