**電通二乙微處理器實驗 實驗結報**

|  |  |  |  |
| --- | --- | --- | --- |
| **實驗名稱** | Lab 06 – 音樂教室 | | |
| **組別** | **30** | **組員** | **03053613馮永軒 03050841傅冠珅** |

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**

**1.**

**#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\_G4, NOTE\_G4, NOTE\_E4, NOTE\_D4, NOTE\_E4, NOTE\_D4, NOTE\_C4,**

**NOTE\_E4, NOTE\_D4, NOTE\_C4, NOTE\_A3, NOTE\_G3, NOTE\_A3, NOTE\_G3,**

**NOTE\_A3, NOTE\_A3, NOTE\_C4, NOTE\_A3, NOTE\_C4, NOTE\_D4, NOTE\_E4,**

**NOTE\_D4, NOTE\_D4, NOTE\_G4, NOTE\_G4, NOTE\_E4, NOTE\_D4, NOTE\_C4,**

**};**

**int noteDurations[] = {**

**6, 16, 8, 8, 8, 8, 4,**

**6, 16, 8, 8, 8, 8, 4,**

**6, 16, 8, 8, 8, 8, 4,**

**6, 16, 8, 8, 8, 8, 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 \* 1.30);**

**}**

**}**

**void setup(){**

**}**

**void loop(){**

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

**delay(2000);**

**}**

**2.**

**#include <Keypad.h>**

**#include "pitches.h"**

**const byte ROWS=4;**

**const byte COLS=4;**

**int duration=200;**

**const byte Buzzer=13;**

**const int rhythm=500;**

**char index;**

**int SoundLeng;**

**char song\_tone;**

**char ch;**

**char keys[ROWS][COLS]={**

**{'1','2','3','A'},**

**{'4','5','6','B'},**

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

**{'\*','0','#','D'}**

**};**

**byte rowPins[ROWS]={7,8,9,10};**

**byte colPins[COLS]={3,4,5,6};**

**int**

**pitch[]={0,NOTE\_C4,NOTE\_D4,NOTE\_E4,NOTE\_F4,NOTE\_G4,NOTE\_A4,NOTE\_B4,**

**NOTE\_C5,NOTE\_D5,NOTE\_E5,NOTE\_F5,NOTE\_G5,NOTE\_A5,NOTE\_B5};**

**float song[] = {5,0.5, 1+7,0.5, 2+7,0.5, 3+7,0.5, 2+7,0.5, 3+7,0.5,**

**4+7,0.5, 5+7,0.5, 5+7,0.5, 5+7,0.5, 4+7,0.5,**

**3+7,0.5,**

**2+7,0.5, 2+7,1, 0,0.5, 5,0.5, 1+7,0.5, 2+7,0.5,**

**3+7,0.5,**

**2+7,0.5, 3+7,0.5, 4+7,0.5, 5+7,0.5, 5+7,0.5,**

**5+7,0.5,**

**6+7,0.5, 5+7,1, 2+7,0.5, 3+7,0.5, 1+7,1, 1+7,0.5,**

**6,0.5,**

**2+7,0.5, 2+7,0.5, 3+7,0.5, 3+7,0.5, 1+7,1, 5+7,0.5,**

**1+7,0.5, 5,0.5, 1+7,0.5, 7,0.5, 1+7,0.5, 0,1,**

**1+7,0.5,**

**6,0.5, 2+7,0.5, 2+7,0.5, 3+7,0.5, 3+7,0.5, 4+7,0.5,**

**4+7,0.5,**

**4+7,0.5, 3+7,0.5, 2+7,0.5, 1+7,0.5, 1+7,1, -1};**

**Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS,**

**COLS);**

**void setup() {**

**pinMode(Buzzer,OUTPUT);**

**Serial.begin(9600);**

**}**

**void loop() {**

**char key = keypad.getKey();**

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

**switch(key){**

**case '0':**

**tone(Buzzer,NOTE\_B5,duration);**

**break;**

**case '1':**

**tone(Buzzer,NOTE\_C4,duration);**

**break;**

**case '2':**

**tone(Buzzer,NOTE\_D4,duration);**

**break;**

**case '3':**

**tone(Buzzer,NOTE\_E4,duration);**

**break;**

**case '4':**

**tone(Buzzer,NOTE\_F4,duration);**

**break;**

**case '5':**

**tone(Buzzer,NOTE\_G4,duration);**

**break;**

**case '6':**

**tone(Buzzer,NOTE\_A4,duration);**

**break;**

**case '7':**

**tone(Buzzer,NOTE\_B4,duration);**

**break;**

**case '8':**

**tone(Buzzer,NOTE\_C5,duration);**

**break;**

**case '9':**

**tone(Buzzer,NOTE\_D5,duration);**

**break;**

**case 'A':**

**tone(Buzzer,NOTE\_E5,duration);**

**break;**

**case 'B':**

**tone(Buzzer,NOTE\_F5,duration);**

**break;**

**case 'C':**

**tone(Buzzer,NOTE\_G5,duration);**

**break;**

**case 'D':**

**tone(Buzzer,NOTE\_A5,duration);**

**break;**

**case 'E':**

**tone(Buzzer,NOTE\_B5,duration);**

**break;**

**case 'F':**

**tone(Buzzer,NOTE\_B5,duration);**

**break;**

**}**

**}**

**if(Serial.read()=='S')**

**{**

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

**Serial.println("Sing");**

**index=0;**

**while (1) {**

**song\_tone = char(song[index]);**

**index++;**

**SoundLeng = int(song[index]\*rhythm);**

**index++;**

**if (song\_tone==-1)**

**break;**

**else if (song\_tone==0)**

**delay(SoundLeng\*1.3);**

**else {**

**tone(Buzzer, pitch[song\_tone], SoundLeng);**

**delay(SoundLeng\*1.3);**

**}**

**}**

**}**

**}**

**3.** **char\*morse[]={"01","1000","1010","100",**

**"0","0010","110","0000",**

**"00","0111","101","0110",**

**"11","10","111","0110",**

**"1101","010","000","1",**

**"001","0001","11","1001",**

**"1011","1100"};**

**const byte Buzzer=13;**

**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,400,100);**

**delay(100);**

**}**

**else{**

**tone(Buzzer,400,300);**

**delay(300);**

**}**

**ptr++;**

**delay(100);**

**}**

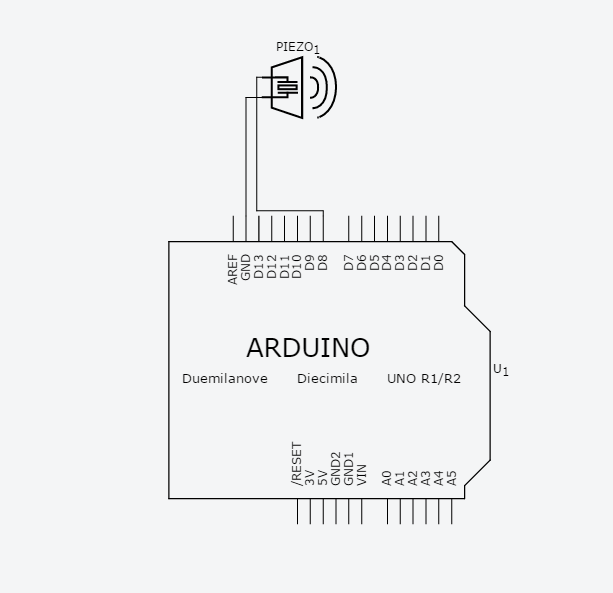
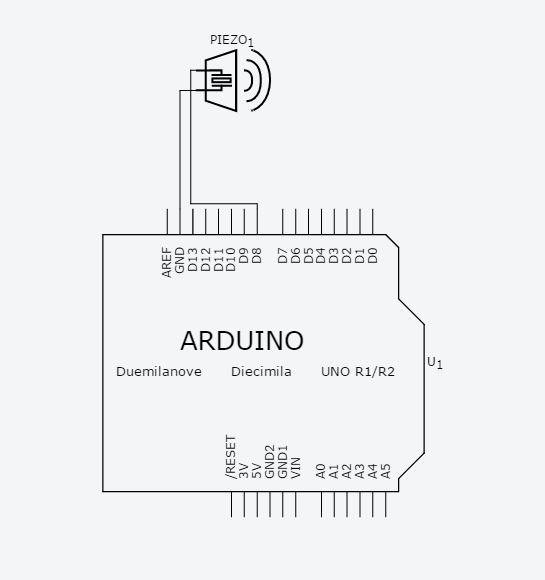
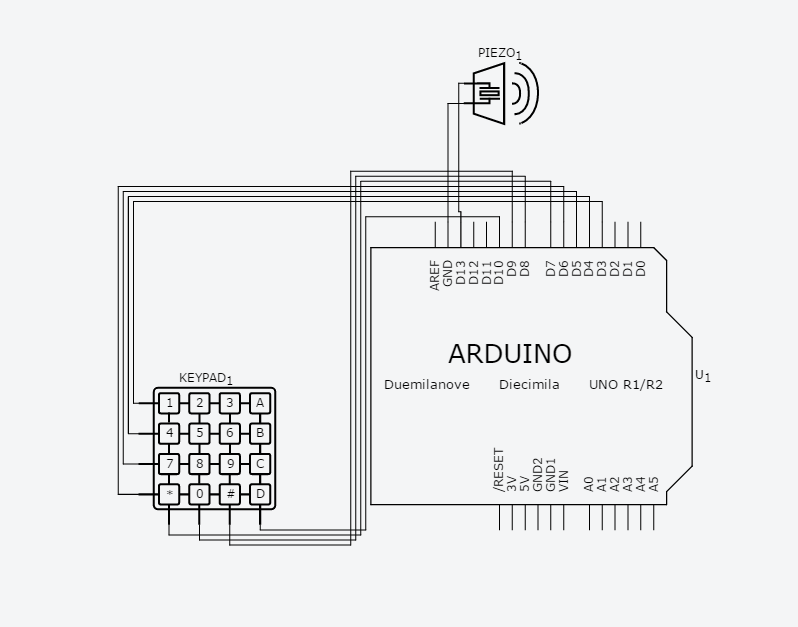
**delay(300);**

**}**

**}**

**}**

1. **實驗結果及分析**

****

1. **心得討論**

**這次的實驗多虧了助教與同學的幫忙才能順利完成**

1. **修正電路圖**
2. **修正程式碼**