實驗六 鍵盤喇叭與類比訊號輸入繼電器控制

實驗目的： Arduino UNO鍵盤整合，控制喇叭發音，達成電子琴之應用；外接喇叭，類比訊號裝置並且能透過繼電器控制電路。

實驗步驟：

1. 先將Arduino連接一個4x4掃描鍵盤和一個喇叭。當鍵盤按下1時，喇叭會唱出Do，當鍵盤按下2時，喇叭會唱出Re，當鍵盤7按下去唱 Si。
2. 使用串列通訊，當PC傳送S或s時，喇叭會唱出一首歌，歌曲請自訂。
3. 小夜燈設計:完成光敏電阻之輸入端設計(A0輸入)，光線改變時會影響LED暗亮，LED接繼電器電路控制。一種將led當成數位接腳，另一種當成類比輸出可調亮度。



|  |
| --- |
| #include<Keypad.h>  #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  #define KEY\_ROWS 4 // 按鍵模組的列數  #define KEY\_COLS 4 // 按鍵模組的行數  const byte potPin = A0;  int val;  int led=11;  // 依照行、列排列的按鍵字元（二維陣列）  char keymap[KEY\_ROWS][KEY\_COLS] = {  {'F', 'B', '7', '3'}, {'E','A','6', '2'},  {'D', '9', '5', '1'}, {'C', '8', '4','0'}  };  char ch,index;  byte colPins[KEY\_COLS] = {2,3,4,5}; // 按鍵模組，行 1~4 接腳。  byte rowPins[KEY\_ROWS] = {6,7,8,9}; // 按鍵模組，列 1~4 接腳。  const byte Buzzer = 10;  int SoundLeng;  char song\_tone;  const int rhythm=200;  Keypad myKeypad = Keypad(makeKeymap(keymap), rowPins, colPins, KEY\_ROWS,  KEY\_COLS);  int notes4[]={NOTE\_C4,NOTE\_D4,NOTE\_E4,NOTE\_F4,NOTE\_G4,NOTE\_A4,NOTE\_B4};  float  song[]={1,1,2,1,3,1,1,1,1,1,2,1,3,1,1,1,3,1,4,1,5,1,0,1,3,1,4,1,5,1,0,1,5,0.5,6,0.5,5,0.5,  4,0.5,3,1,1,1,5,0.5,6,0.5,5,0.5,4,0.5,3,1,1,1,1,1,5,1,1,1,0,1,1,1,5,1,1,1,0,1,-1};  void setup(){  pinMode(Buzzer,OUTPUT);  Serial.begin(9600);  pinMode(led,OUTPUT);  }  void loop(){  val = analogRead(potPin);  Serial.println(val);  if (val <= 500)  {  digitalWrite(led, LOW);  }  else  {  digitalWrite(led, HIGH);  }  // 透過 Keypad 物件的 getKey()方法讀取按鍵的字元  char key = myKeypad.getKey();  if (key){ // 若有按鍵被按下…  Serial.println(key); // 顯示按鍵的字元  if(key >= '0' && key <= '7'){  tone (Buzzer,notes4[key-'0'],300);  }}  if (Serial.available()){  ch = Serial.read();  if( ch == 'S' || ch == 's'){  index = 0;  while(1){  song\_tone = song[index];  index ++;  SoundLeng = song[index]\*rhythm;  index++;  if(song[index] == -1){  break;  }  else if (song\_tone == 0){  delay(SoundLeng\*1.3);  }  else{  tone ( Buzzer,notes4[song\_tone],SoundLeng);  delay(SoundLeng\*1.3);  }  }  }  }} |

