作業三 LCM控制實驗文字走馬燈

資工三乙/林采昕/408262143 2021/11/05

Q1 何謂脈衝寬度調變 (PWM -Pulse Width Modulation)其應用 為何 ?

脈寬調變會將數位訊號轉變成類比訊號，改變保持開啟與關閉的時間長度。

Q2 Clock (時脈 ) 是一週而復始的訊號，請解釋頻率 (frequency)1MHz的定

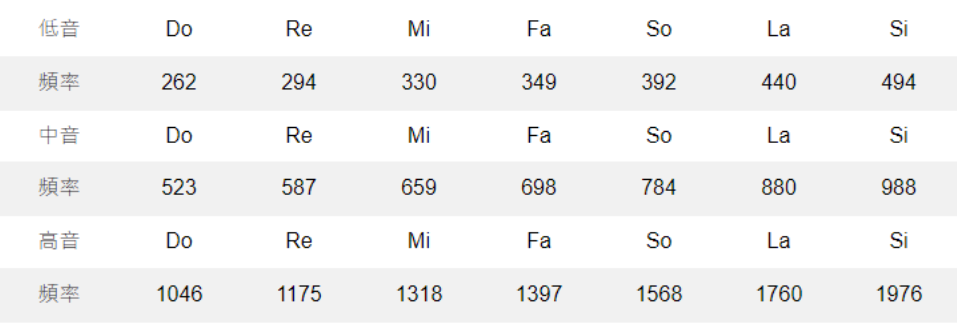
義 ? 何謂 cycle time? 何謂 duty cycle?

頻率 : 每一秒週期性事件發生的次數。

cycle time : 執行一個 clock 所需要的時間。

duty cycle : 在一串理想的脈衝序列中，代表1的正脈衝的持續時間與脈衝總周期的比值。

Q3: 請寫出 七音符 高音、中音、低音 及其對應頻率表

程式碼:

#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

byte Row=0, Col=0;

int i = 0;

int speakerPin = D3;

// 依照簡譜的順序，填入代表的音符，空白代表休止符

char notes[] = "cfefgfgaabadggfffedefccfefgfgabadggfffedeffaCagfefgfedcfaCagfefgcffgaabadggfefccfccdcczczccdcczczcc ";

int tmp[105][3]; //tmp[音符][音高]

// 決定每個音階的拍子

double beats[] = {0.5, 1, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 1, 0.5, 0.5, 2, 0.5, 0.5, 1, 0.5, 0.5, 1, 0.5, 0.5, 1, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 0.5, 0.5, 1,0.5,0.5,1,0.5,0.5,2,0.5,0.5,1,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,1,0.5,0.5,1,0.5,0.5,2,0.5,0.5,0.5,2,0.5,0.5,0.5,0.5,0.5,0.5,0.5,2,2,2,0.5,0.5,1,0.5,0.5,0.5,0.5,1,0.5,0.5,0.5,0.5,0.5,1,0.5,0.5,0.5,0.5,1,0.5,0.5,0.5,0.5,0.5};

// 利用 sizeof()，算出總共要多少音符

int length = sizeof(notes);

int tempo = 600;

int duration = 300;

int buttonState1,buttonState2,buttonState3,buttonState4 = 0;

void setup() {

pinMode(speakerPin, OUTPUT);

pinMode(BT1, INPUT);

pinMode(BT2, INPUT);

pinMode(BT3, INPUT);

pinMode(BT4, INPUT);

pinMode(10, INPUT); //R1: S1,S2,S3,S4 (1,2,3,A)

pinMode(11, INPUT\_PULLUP); //R2: S5,S6,S7,S8 (4,5,6,B)

pinMode(12, INPUT\_PULLUP); //R3: S9, S10, S11,S12 (7,8,9,C)

pinMode(13, INPUT\_PULLUP); //R4: (\*,0,#,D)

pinMode(A0, OUTPUT); //A1, C1: S1,S5,S9 (1,4,7,\*)

pinMode(A1, OUTPUT); //A2, C2: S2,S6,S10 (2,5,8,0)

pinMode(A2, OUTPUT); //A3, C3: S3,S7,S11 (3,6,9,#)

pinMode(A3, OUTPUT); //A4, C4, S4,S8,S12 (\*,0, #,D) // (A, B, C, D) is correct.

//Pin left to right :R1 R2 R3 R4 C1 C2 C3 C4 digitalWrite(A0,HIGH);

digitalWrite(A1,HIGH);

digitalWrite(A2,HIGH);

digitalWrite(A3,HIGH);

lcd.begin(16, 2); // start the library

}

// 放入高、中、低、的七個音符的頻率 tones[音高][音符]

int tones[3][8] = { {261, 294, 330, 349, 392, 440, 494}, {523, 587, 659, 698, 784, 880, 988}, {1046, 1175, 1318, 1397, 1568, 1760, 1976} };

int minn = 2000;

int n = 0; // n : 紀錄音高

void loop() {

byte keyindex=0;

buttonState1 = digitalRead(BT1);

buttonState2 = digitalRead(BT2);

buttonState3 = digitalRead(BT3);

buttonState4 = digitalRead(BT4);

if ((keyscan()==true) && (keyindex=(Row-1)\*4+Col)){

delay(500);

if(keyindex == 4){

lcd.clear();

lcd.setCursor(10,0);

lcd.print(" High");

n = 2;

}

else if(keyindex == 8){

lcd.clear();

lcd.setCursor(10,0);

lcd.print("Middle");

n = 1;

}

else if(keyindex == 12){

lcd.clear();

lcd.setCursor(10,0);

lcd.print(" Low");

n = 0;

}

else {

lcd.setCursor(0,0);

tmp[i][1] = n;

// 紀錄音符

if(keyindex == 1) {

lcd.print("Do");

tmp[i][0] = 0;

}

if(keyindex == 2){

lcd.print("Re");

tmp[i][0] = 1;

}

if(keyindex == 3) {

lcd.print("Mi");

tmp[i][0] = 2;

}

if(keyindex == 5) {

lcd.print("Fa");

tmp[i][0] = 3;

}

if(keyindex == 6) {

lcd.print("So");

tmp[i][0] = 4;

}

if(keyindex == 7) {

lcd.print("La");

tmp[i][0] = 5;

}

if(keyindex == 9) {

lcd.print("Si");

tmp[i][0] = 6;

}

tone(speakerPin,tones[tmp[i][1]][tmp[i][0]], duration);

lcd.setCursor(0,1);

lcd.print(tones[tmp[i][1]][tmp[i][0]]);

// delay(duration);

// noTone(speakerPin);

// delay(duration);

i++;

}

}

if(buttonState1 == LOW) {

lcd.clear();//清空lcd面板上的字

lcd.setCursor(0,1);

lcd.print("Recording");

i = 0; //從第0個開始記錄

minn = 2000;

}

else if(buttonState2 == LOW) {

lcd.clear();//清空lcd面板上的字

lcd.setCursor(0,1);

lcd.print("No Recording");

// 紀錄結束時是第幾個音，避免按了BTN2後還記錄到其他音符。

if(minn > i) minn = i;

}

else if(buttonState3 == LOW) {

lcd.clear();//清空lcd面板上的字

lcd.setCursor(0,1);

lcd.print("Playing");

for(int j = 0; j < minn; j++){

tone(speakerPin,tones[tmp[j][1]][tmp[j][0]], duration);

delay(duration);

noTone(speakerPin);

delay(duration);

}

}

else if(buttonState4 == LOW) {

lcd.clear();//清空lcd面板上的字

lcd.setCursor(0,0);

lcd.print("");

lcd.setCursor(0,1);

lcd.print("Playing a Song.");

// 利用 for 來播放我們設定的歌曲，一個音一個音撥放

for (int i = 0; i < length; i++) {

// 如果是空白的話，不撥放音樂

if (notes[i] == ' ') {

delay(beats[i] \* tempo); // rest

} else {

// 呼叫 palyNote() 這個 function，將音符轉換成訊號讓蜂鳴器發聲

playNote(speakerPin,notes[i], beats[i] \* tempo);

}

// 每個音符之間的間隔，這邊設定的長短會有連音 or 段音的效果

delay(tempo/10);

}

}

}

void playNote(int OutputPin, char note, unsigned long duration) {

// 音符字元與對應的頻率由兩個矩陣表示

char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C', 'D', 'E', 'F', 'G', 'A', 'B','z'};

int tones[] = { 523, 587, 659, 698, 784, 880, 988,1046, 1175, 1318, 1397, 1568, 1760, 1976, 440};

// 播放音符對應的頻率

for (int i = 0; i < 14; i++) {

if (names[i] == note) {

tone(OutputPin,tones[i], duration);

//下方的 delay() 及 noTone ()，測試過後一定要有這兩行，整體的撥放出來的東西才不會亂掉，可能是因為 Arduino 送出tone () 頻率後會馬上接著執行下個指令，不會等聲音播完，導致撥出的聲音混合而亂掉

delay(duration);

noTone(OutputPin);

}

}

}

bool keyscan( ){

Row=0;Col=0;

bool keypressed = false;

//scan col1

digitalWrite(A0, LOW);

digitalWrite(A1, HIGH);

digitalWrite(A2, HIGH);

digitalWrite(A3, HIGH);

delayMicroseconds(100);

//Read keys in row.1

if(digitalRead(10)==LOW) {

digitalWrite(A0, HIGH);

Col=1;Row=1;

keypressed = true;

return(keypressed);

}

//Read keys in row.2

if(digitalRead(11)==LOW) {

digitalWrite(A0, HIGH);

Col=1;Row=2;

keypressed = true;

return(keypressed);

}

//Read keys in row.3

if(digitalRead(12)==LOW) {

digitalWrite(A0, HIGH);

Col=1;Row=3;

keypressed = true;

return(keypressed);

}

//Read keys in row.4

if(digitalRead(13)==LOW) {

digitalWrite(A0, HIGH);

Col=1;Row=4;

keypressed = true;

return(keypressed);

}

//scan col 2

digitalWrite(A0, HIGH);

digitalWrite(A1, LOW);

digitalWrite(A2, HIGH);

digitalWrite(A3, HIGH);

delayMicroseconds(100);

//Read keys in row.1

if(digitalRead(10)==LOW) {

digitalWrite(A1, HIGH);

Col=2;Row=1;

keypressed = true;

return(keypressed);

}

//Read keys in row.2

if(digitalRead(11)==LOW) {

digitalWrite(A1, HIGH);

Col=2;Row=2;

keypressed = true;

return(keypressed);

}

//Read keys in row.3

if(digitalRead(12)==LOW) {

digitalWrite(A1, HIGH);

Col=2;Row=3;

keypressed = true;

return(keypressed);

}

//Read keys in row.4

if(digitalRead(13)==LOW) {

digitalWrite(A1, HIGH);

Col=2;Row=4;

keypressed = true;

return(keypressed);

}

//scan col 3

digitalWrite(A0, HIGH);

digitalWrite(A1, HIGH);

digitalWrite(A2, LOW);

digitalWrite(A3, HIGH);

delayMicroseconds(100);

//Read keys in row.1

if(digitalRead(10)==LOW) {

digitalWrite(A2, HIGH);

Col=3;Row=1;

keypressed = true;

return(keypressed);

}

//Read keys in row.2

if(digitalRead(11)==LOW) {

digitalWrite(A2, HIGH);

Col=3;Row=2;

keypressed = true;

return(keypressed);

}

//Read keys in row.3

if(digitalRead(12)==LOW) {

digitalWrite(A2, HIGH);

Col=3;Row=3;

keypressed = true;

return(keypressed);

}

//Read keys in row.4

if(digitalRead(13)==LOW) {

digitalWrite(A2, HIGH);

Col=3;Row=4;

keypressed = true;

return(keypressed);

}

//scan col 4

digitalWrite(A0, HIGH);

digitalWrite(A1, HIGH);

digitalWrite(A2, HIGH);

digitalWrite(A3, LOW);

delayMicroseconds(100);

//Read keys in row.1

if(digitalRead(10)==LOW) {

digitalWrite(A3, HIGH);

Col=4;Row=1;

keypressed = true;

return(keypressed);

}

//Read keys in row.2

if(digitalRead(11)==LOW) {

digitalWrite(A3, HIGH);

Col=4;Row=2;

keypressed = true;

return(keypressed);

}

//Read keys in row.3

if(digitalRead(12)==LOW) {

digitalWrite(A3, HIGH);

Col=4;Row=3;

keypressed = true;

return(keypressed);

}

//Read keys in row.4

if(digitalRead(13)==LOW){

digitalWrite(A3, HIGH);

Col=4;Row=4;

keypressed = true;

return(keypressed);

}

return(false);

}