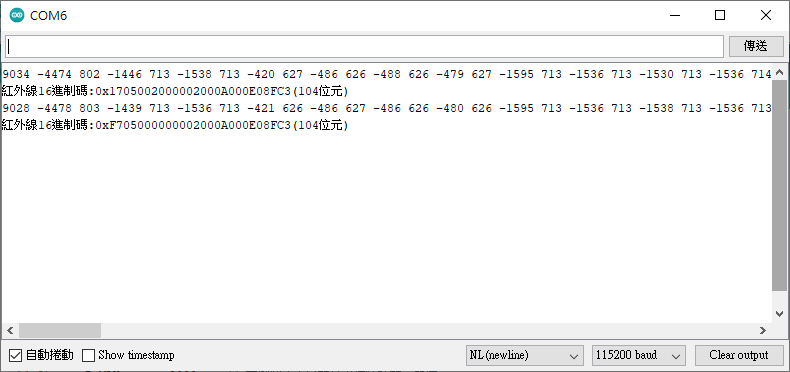
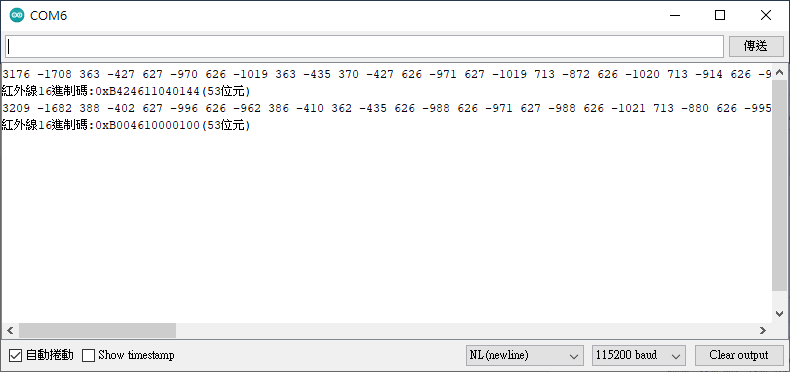
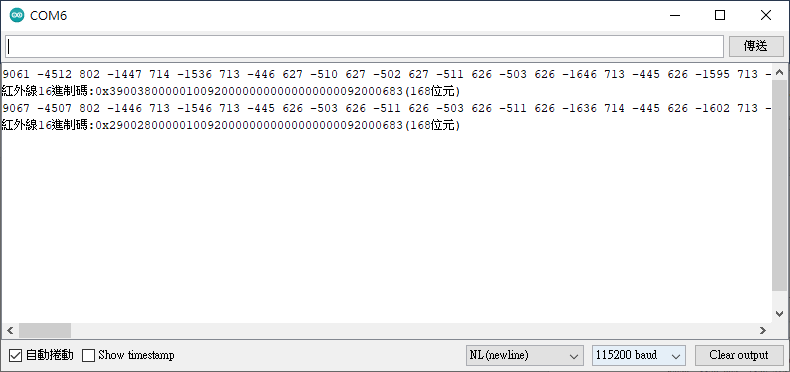
右邊(小白)



中(大白)



左邊(大藍)



程式碼

/\*

\* 紅外線接收解碼程式

\* 目前支援 NEC HERAN HITACHI DAIKIN 格式

\*/

//#define NEC

//#define HERAN

//#define HITACHI

#define DAIKIN

#define CODE\_NUMBER 50 //50\*4=200 最多可以接收 200位元

#if defined (NEC )

#define D\_START 4000 // 判斷為啟始位元最小持續時間，單位 us

#define D\_IDLE 25000 // 一段時間沒變化就進入 Idle，單位 us

#define D\_MAX 2000 // 電壓狀態不變的最大持續時間，單位 us

#define D\_MIN 300 // 電壓狀態不變的最小持續時間，單位 us

#define D\_HI 1000 // 判斷為高電位最小持續時間，單位 us

#endif

#if defined (HERAN )

#define D\_START 4000 // 判斷為啟始位元最小持續時間，單位 us

#define D\_IDLE 25000 // 一段時間沒變化就進入 Idle，單位 us

#define D\_MAX 25000 // 電壓狀態不變的最大持續時間，單位 us

#define D\_MIN 300 // 電壓狀態不變的最小持續時間，單位 us

#define D\_HI 1000 // 判斷為高電位最小持續時間，單位 us

#endif

#if defined (HITACHI )

#define D\_START 1500 // 判斷為啟始位元最小持續時間，單位 us

#define D\_IDLE 25000 // 一段時間沒變化就進入 Idle，單位 us

#define D\_MAX 2000 // 電壓狀態不變的最大持續時間，單位 us

#define D\_MIN 300 // 電壓狀態不變的最小持續時間，單位 us

#define D\_HI 1000 // 判斷為高電位最小持續時間，單位 us

#endif

#if defined (DAIKIN )

#define D\_START 1500 // 判斷為啟始位元最小持續時間，單位 us

#define D\_IDLE 20000 // 一段時間沒變化就進入 Idle，單位 us

#define D\_MAX 2000 // 電壓狀態不變的最大持續時間，單位 us

#define D\_MIN 300 // 電壓狀態不變的最小持續時間，單位 us

#define D\_HI 1000 // 判斷為高電位最小持續時間，單位 us

#endif

byte startBit = 0; // 是否出現啟始位元

const int IR\_rec\_pin = 3; // IR 接收器輸出腳位

boolean IRstate = LOW; // IR 接收器輸出腳位狀態

boolean IRstate\_last = LOW; // IR 接收器輸出腳位狀態(上一次)

unsigned long int time\_last = 0; // 上一次 IRstate 變化的時間

unsigned long int hTime = 0; // 高電位時間

unsigned long int lTime = 0; // 低電位時間

boolean isIdle = true; // 是否在等待 IR 訊號模式Idle）

int number = 0; // 接收多少位元

unsigned long int factor = 1; // 2進位加權

byte code[CODE\_NUMBER] ; // 紅外線傳送碼

byte index =0; // code陣列索引

void setup( ) {

Serial.begin( 115200 );

pinMode( IR\_rec\_pin, INPUT ); // 設定針腳 I/O 模式

IRstate = digitalRead( IR\_rec\_pin ); // 取得腳位元狀態初始值

IRstate\_last = IRstate;

Serial.println( "紅外線接收解碼程式" );

}

void loop( ) {

IRRecCode();

delayMicroseconds( 5 );

}

void IRRecCode() {

IRstate = digitalRead( IR\_rec\_pin );// 讀取腳位狀態

if ( (micros( ) - time\_last >= D\_IDLE) && !isIdle && number>0 ) { // 時間間隔大於設定的時間，且原本的狀態為接收中狀態

if(number>0){

Serial.print(hTime);

Serial.print(" ");

Serial.print(micros( ) - time\_last);

Serial.println("");

Serial.print("紅外線16進制碼:0x");

for(int i=index-1;i>=0;i--){

Serial.print(code[i], HEX);

}

Serial.print("(");

Serial.print(number);

Serial.println( "位元)" );

}

isIdle = true; //進入等待狀態

startBit=0;

number=0;

index=0;

}

//狀態改變

if ( IRstate != IRstate\_last ) {

long int timeNow = micros( ); // 記錄目前時間

long int dTime = timeNow - time\_last; // 與上一次腳位變化的時間差

if ( (dTime < D\_MAX) && (dTime > D\_MIN) && !isIdle) {

//判斷低電位的時間間隔

if (IRstate == LOW) {

Serial.print(hTime);

Serial.print(" ");

Serial.print(0-dTime);

Serial.print(" ");

//如果是1，加上其位元的權數，不是1就不用加了

if (dTime > D\_HI) {

code[index] += factor;

}

factor = factor \* 2;

number++;

if(number%4==0){

factor=1;

index++;

}

}else{

//HIGH

hTime=dTime;

//Serial.print("h");

}

}

// 記錄此次時間

time\_last = timeNow;

//判斷啟始位元訊號

if(isIdle && dTime > D\_START && startBit==0 && IRstate == HIGH){

//高電位一段時間

startBit=1;

hTime=dTime;

}else if(isIdle && dTime > D\_START && startBit==1 && IRstate == LOW){

//低電位一段時間

isIdle = false; //進入接收中狀態

lTime=dTime;

startBit=0;

//啟始位元

for(int i=0;i<CODE\_NUMBER;i++){

code[i] = 0;

}

number = 0;

factor = 1;

index = 0;

Serial.print(hTime);

Serial.print(" -");

Serial.print(lTime);

Serial.print(" ");

}

}

// 記錄此次狀態

IRstate\_last = IRstate;

}//end IRRecCode

參考: https://www.block.tw/blog/arduino-irr/