

Em định xây dựng thuật toán theo kiểu này thầy xem có ổn không ạ. Thầy và các bạn góp ý giúp em với

**Thầy Hoàng Anh:** thuật toán về tinh thần thì tương đối ổn, tuy nhiên như thế này vẫn chưa chạy được, do:

* Các em thiếu chế độ gió (các cấp độ quạt gió khác nhau)
* Tăng hay giảm nhiệt độ về cùng một giá trị thì đều như nhau, các em nên dùng biến lưu giá trị nhiệt độ để biết lúc nào nên tác động vào nút tăng hay giảm với vòng lặp 13 lần như trên
* Chế độ tắt bật nên xếp vào vòng lặp trong cùng thay vì ngoài cùng như trên
* Bổ sung vào mảng RAW các giá trị nằm ở cuối hiển thị tương ứng chế độ, nhiệt độ, mức quạt, và trạng thái tắt/bật. Việc này sẽ không ảnh hưởng tới tín hiệu IR, tôi sẽ giải thích các em sau

Đây là chuoeng trình bọn em thiết kế theo cái thuật toán trên đã được thêm chế độ Wind

int LED1 = 3; //khai báo nút ON/OFF

int LED2 = 4; //khai báo nút MODE

int LED3 = 5; //khai báo nút UP

int LED4 = 6; //khai báo nút DOWN

int LED5 = 7; //khai báo nút WIND

void setup() {

// put your setup code here, to run once:

pinMode(LED1, OUTPUT);

pinMode(LED2, OUTPUT);

pinMode(LED3, OUTPUT);

pinMode(LED4, OUTPUT);

pinMode(LED5, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

char i,j,k,m;

for(i=1;i<=2;i++) //nut ON OFF

{

digitalWrite(LED1, HIGH); //nhấn ON

delay(1000);

digitalWrite(LED1, LOW); //nhả nút ON

delay(1000);

for(j=1;j<=4;j++) //nut MODE

{

digitalWrite(LED2, HIGH); //nhấn MODE

delay(1000);

digitalWrite(LED2, LOW); //nhả MODE

delay(1000);

for(k=16;k<=29;k++) //nut TANG

{

digitalWrite(LED3, HIGH); //nhấn UP

delay(1000);

digitalWrite(LED3, LOW); //nhả UP

delay(1000);

if(k==29)

{

for(m=29;m>=16;m--) //nut GIAM

{

digitalWrite(LED4, HIGH); //nhấn DOWN

delay(1000);

digitalWrite(LED4, LOW); //nhả DOWN

delay(1000);

}

}

}

}

}

digitalWrite(LED5, HIGH); //nhấn WIND

delay(1000);

digitalWrite(LED5, LOW); //nhả WIND

delay(1000);

}

Thầy xem giúp bọn em

**Thầy Hoàng Anh:** Các em chèn phần code thu IR mã raw vào đi, đồng thơi quy định các chế độ hoạt động (ví dụ: chế độ auto là 0, sưởi là 1, mát là 2, gió là 3…) rồi chèn vào mảng raw thu được. Phần này Mười đã làm rồi, hy vọng các em giúp đỡ lẫn nhau để công việc hiệu quả

int opto1nutONOFF = 3;

int opto2nutMODE = 4;

int opto3nutWIND = 5;

int opto4nutUP = 6;

int opto5nutDOWN = 7;

char giatri = 0;

byte j;

int t=300; //khoảng thời gian trễ

void setup() {

// put your setup code here, to run once:

for(char i=3;i<8;i++) //khai báo đầu ra là output

{

pinMode(i, OUTPUT);

}

}

void loop() {

// put your main code here, to run repeatedly:

if(giatri<1) //giá trị để ngừng lấy mã

{

byte nhietdo;

Lấy dữ liệu IR chế độ OFF 1 lần duy nhất và hiện thị ra serial Monitor

for(int mode=0;mode<4;mode++) //các chế độ 0: auto, 1: sưởi, 2: làm mát , 3: chế độ quạt gió

{

digitalWrite(opto2nutMODE, HIGH); //nhấn MODE

delay(t);

Lấy dữ liệu IR và hiện thị ra serial Monitor

Chèn trạng thái vào mảng RAW

digitalWrite(opto2nutMODE, LOW); //nhả MODE

delay(t);

for(char wind=0;wind<=3;wind++) //các chế độ tốc độ gió khác nhau

{

digitalWrite(opto3nutWIND, HIGH); //nhấn WIND

delay(t);

Lấy dữ liệu IR và hiện thị ra serial Monitor

Chèn trạng thái vào mảng RAW

digitalWrite(opto3nutWIND, LOW); //nhả WIND

delay(t);

for(nhietdo=17;nhietdo<=30;nhietdo++)

{

digitalWrite(opto4nutUP, HIGH); //nhấn UP

delay(t);

Lấy dữ liệu IR và hiện thị ra serial Monitor

Chèn trạng thái vào mảng RAW

digitalWrite(opto4nutUP, LOW); //nhả UP

delay(t);

digitalWrite(opto1nutONOFF, HIGH); //nhấn OFF

delay(t);

digitalWrite(opto1nutONOFF, LOW); //nhả OFF

delay(t);

digitalWrite(opto1nutONOFF, HIGH); //nhấn ON

delay(t);

Lấy dữ liệu IR và hiện thị ra serial Monitor

Chèn trạng thái vào mảng RAW

digitalWrite(opto1nutONOFF, LOW); //nhả ON

delay(t);

if(nhietdo==30) //nếu nhiệt độ bằng 30 thì giảm nhiệt độ xuống

{

for(j=29;j>=16;j--)

{

digitalWrite(opto5nutDOWN, HIGH); //nhấn DOWN

delay(t);

digitalWrite(opto5nutDOWN, LOW); //nhả DOWn

delay(t); }

}

}

}

}

}

giatri = 1; //dừng lấy mã

}

Thầy xem giuos em

**Định**: Lệnh delay sẽ pause chương trình lại, tức là lúc đó chương trình sẽ treo nên không thể nhận được tín hiệu . Mình đề nghị các bạn tham khảo link này để thay thế:

<https://learn.adafruit.com/multi-tasking-the-arduino-part-1/using-millis-for-timing>

Mình sẽ giải thích ngắn gọn:

Đây là 1 chương trình nháy LED.

Thay vì sử dụng delay thì chương trình sẽ dùng 1 bộ đếm thời gian để theo dõi vòng lặp, mỗi lần đèn LED chuyển đổi giá trị thì biến thời gian lúc đó sẽ được lưu lại. Mỗi lần chạy vòng lặp sẽ kiểm tra thời gian hiện tại, nếu đã đủ lâu kể từ lần đổi giá trị lớn nhất thì đèn LED sẽ lại được đổi giá trị và thời gian cập nhật gần nhất sẽ được lưu lại.

Về mặt ý tưởng thì dùng lệnh delay giống như mình bật đèn, sau đó đi ngủ rồi hẹn đồng hồ báo thức sau 1 tiếng thì dậy tắt đen đi. Còn xài bộ đếm thời gian thì mình bật đèn, sau đó mỗi phút đều nhìn đồng hồ, nếu thời gian kể từ lần cuối cùng bật đèn đã bằng hoặc hơn 1 tiếng thì sẽ tắt đi.

// These variables store the flash pattern

// and the current state of the LED

int ledPin = 13; // the number of the LED pin

int ledState = LOW; // ledState used to set the LED

unsigned long previousMillis = 0; // will store last time LED was updated

long OnTime = 250; // milliseconds of on-time

long OffTime = 750; // milliseconds of off-time

void setup()

{

// set the digital pin as output:

pinMode(ledPin, OUTPUT);

}

void loop()

{

// check to see if it's time to change the state of the LED

unsigned long currentMillis = millis();

if((ledState == HIGH) && (currentMillis - previousMillis >= OnTime))

{

ledState = LOW; // Turn it off

previousMillis = currentMillis; // Remember the time

digitalWrite(ledPin, ledState); // Update the actual LED

}

else if ((ledState == LOW) && (currentMillis - previousMillis >= OffTime))

{

ledState = HIGH; // turn it on

previousMillis = currentMillis; // Remember the time

digitalWrite(ledPin, ledState); // Update the actual LED

}

/\*

\* IRremote: IRrecvDump - dump details of IR codes with IRrecv

\* An IR detector/demodulator must be connected to the input RECV\_PIN.

\* Version 0.1 July, 2009

\* Copyright 2009 Ken Shirriff

\* http://arcfn.com

\* JVC and Panasonic protocol added by Kristian Lauszus (thanks to zenwheel and other people at the original blog post)

\*/

/\*import processing.serial.\*;

Serial mySerial;

PrintWriter output;\*/

#include <IRremote.h>

//Serial mySerial;

//PrintWriter output;

int RECV\_PIN = 11;

int opto1nutONOFF = 3;

int opto2nutMODE = 4;

int opto3nutWIND = 5;

int opto4nutUP = 6;

int opto5nutDOWN = 7;

int opto6DKLED = 8;

byte j;

byte giatri = 0;

int t=2000;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

void setup()

{

Serial.begin(9600);

irrecv.enableIRIn(); // Start the receiver

/\*mySerial = new Serial( this, Serial.list()[0], 9600 );

output = createWriter( "data.txt" );\*/

for(byte i=3;i<=8;i++) //khai báo đầu ra là output

{

pinMode(i, OUTPUT);

}

}

// Dumps out the decode\_results structure.

// Call this after IRrecv::decode()

// void \* to work around compiler issue

//void dump(void \*v) {

// decode\_results \*results = (decode\_results \*)v

void dump(decode\_results \*results) {

int count = results->rawlen;

/\*if (results->decode\_type == UNKNOWN) {

Serial.print("Unknown encoding: ");

}

else if (results->decode\_type == NEC) {

Serial.print("Decoded NEC: ");

}

else if (results->decode\_type == SONY) {

Serial.print("Decoded SONY: ");

}

else if (results->decode\_type == RC5) {

Serial.print("Decoded RC5: ");

}

else if (results->decode\_type == RC6) {

Serial.print("Decoded RC6: ");

}

else if (results->decode\_type == PANASONIC) {

Serial.print("Decoded PANASONIC - Address: ");

Serial.print(results->panasonicAddress,HEX);

Serial.print(" Value: ");

}

else if (results->decode\_type == JVC) {

Serial.print("Decoded JVC: ");

}\*/

/\*Serial.print(results->value, HEX);

Serial.print(" (");

Serial.print(results->bits, DEC);

Serial.println(" bits)");\*/

Serial.print("Raw (");

Serial.print(count, DEC);

Serial.print("): ");

int t=2000;

/\*output.print("Raw (");

output.print(count, DEC);

output.print("): ");\*/

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

if ((i % 2) == 1) {

Serial.print(results->rawbuf[i]\*USECPERTICK, DEC);

}

else {

Serial.print(-(int)results->rawbuf[i]\*USECPERTICK, DEC);

}

Serial.print(" ");

}

}

void hienthi() //tạo chương trình con hiển thị

{

if (irrecv.decode(&results))

{

//Serial.println(results.value, HEX);

dump(&results);

irrecv.resume(); // Receive the next value

}

}

void loop() {

if(giatri<1)

{

delay(10000);

byte nhietdo=16;

if(nhietdo==16)

{

digitalWrite(opto2nutMODE, HIGH); //nhấn MODE

delay(t);hienthi(); //hiển thị mã raw

Serial.print(" 0 1 0 16 "); //in trạng thái của điều hòa

Serial.println(""); //xuống dòng

delay(1000);

digitalWrite(opto2nutMODE, LOW); //nhả MODE

delay(t);

for(byte wind=3;wind>=0;wind--) //các chế độ tốc độ gió khác nhau

{

digitalWrite(opto3nutWIND, HIGH); //nhấn WIND

delay(t);hienthi(); //hiển thị trạng thái

Serial.print(" 0 1 "); //in trạng thái

Serial.print(wind); //in trạng thái mức wind

Serial.print(" 16"); //in nhiệt độ

Serial.println("");

digitalWrite(opto3nutWIND, LOW); //nhả WIND

delay(t);

for(nhietdo=17;nhietdo<=30;nhietdo++)

{

digitalWrite(opto4nutUP, HIGH); //nhấn UP

delay(t);hienthi();

Serial.print("0 1 ");

Serial.print(wind);

Serial.print(" ");

Serial.print(nhietdo);

Serial.println("");

digitalWrite(opto4nutUP, LOW); //nhả UP

delay(t);

digitalWrite(opto1nutONOFF, HIGH); //nhấn OFF

delay(t);hienthi();

Serial.println("");

digitalWrite(opto1nutONOFF, LOW); //nhả OFF

delay(t);

digitalWrite(opto1nutONOFF, HIGH); //nhấn ON

delay(t); hienthi();

Serial.print("1 1 ");

Serial.print(wind);

Serial.print(" ");

Serial.print(nhietdo);

Serial.println("");

digitalWrite(opto1nutONOFF, LOW); //nhả ON

delay(t);

if(nhietdo==30) //nếu nhiệt độ bằng 30 thì giảm nhiệt độ xuống

{

for(j=29;j>=16;j--)

{

if(j!=16)

{

digitalWrite(opto5nutDOWN, HIGH);

delay(t);

digitalWrite(opto5nutDOWN, LOW);

delay(t);

}

else

{

digitalWrite(opto5nutDOWN, HIGH);

delay(t);hienthi();

Serial.print("0 1 ");

Serial.print(wind);

Serial.print(" 16");

Serial.println("");

digitalWrite(opto5nutDOWN, LOW);

delay(t);

digitalWrite(opto1nutONOFF, HIGH); //nhấn OFF

delay(t);

digitalWrite(opto1nutONOFF, LOW); //nhả OFF

delay(t);hienthi();

Serial.println("");

digitalWrite(opto1nutONOFF, HIGH); //nhấn ON

delay(t);hienthi();

Serial.print("1 1 ");

Serial.print(wind);

Serial.print(" 16");

Serial.println("");

digitalWrite(opto1nutONOFF, LOW); //nhả ON

delay(t);

}

}

}

}

}

}

}

giatri = 2;

}

Đây là chương trình em mới sửa dồn 2 con vào 1. Lấy mã của một chế độ mode. Em thấy dùng hàm delay vẫn được có ảnh hưởng gì đâu ạ. Thầy và các bạn xem giúp em.

**Thầy Hoàng Anh:** Nếu mạch chạy vẫn ổn thì ok

**Định:** Thư viện IRremote sử dụng 1 timer interrupt để đọc dữ liệu, đó là lý do tại sao code delay vẫn chạy, vì interrupt này đã dừng chương trình để chuyển sang chạy code nhận dữ liệu, mọi người có thể tham khảo trong link sau,:

<http://www.engblaze.com/microcontroller-tutorial-avr-and-arduino-timer-interrupts/>

Đây là một khái niệm rất quan trong trong lập trình vi xử lý, các bạn nên tìm hiểu rõ ràng để sau này sử dụng. Ngoài ra trong tương lai 1 arduino của chúng ta sẽ được sử dụng để nhận dữ liệu từ nhiều sensor và điều khiển nhiều thiết bị 1 lúc, nên phương pháp pseudo-parallel bằng bộ đếm thời gian cũng rất hay, về cơ bản là nên xài thay thế delay để đảm bảo độ mượt của chương trình.

