

Phân tích và giải thích đoạn code Arduino

Đoạn code bạn cung cấp là một chương trình Arduino thực hiện chức năng đo huyết áp và nhịp tim, hiển thị thông tin lên màn hình LCD và kiểm soát các thiết bị như motor và van (valve). Dưới đây là phần giải thích chi tiết.

1. Tổng quan chương trình

1. Chức năng chính:

- Đo huyết áp tâm thu (systolic) và tâm trương (diastolic).
- Đo nhịp tim (heart rate).
- Hiển thị kết quả đo trên màn hình LCD 16x2.
- Điều khiển motor và valve để bơm và xả khí.

2. Cấu trúc chính:

- Chương trình sử dụng **state machine** (máy trạng thái) để chuyển đổi giữa các trạng thái: startState, inflate1State, inflate2State, deflateState, displayState, và resetState.
- Sử dụng Timer để tạo ngắt định kỳ và thực hiện các chức năng như đo lường áp suất, đếm thời gian.

3. Phần cứng được sử dụng:

- Motor và valve: điều khiển bơm/xả khí.
 - Nút nhấn: bt_start, bt_stop, bt_resume.
 - Màn hình LCD: hiển thị thông tin.
 - Bộ chuyển đổi ADC: đo tín hiệu từ cảm biến áp suất.
 - Arduino Timer: xử lý ngắt định kỳ.
-

2. Phân tích từng phần chi tiết

Phần đầu - Khai báo và cấu hình ban đầu

```
#include <avr/interrupt.h>
```

```
#include <LiquidCrystal.h>
```

- avr/interrupt.h: Thư viện xử lý ngắt cho vi điều khiển AVR (Arduino).
- LiquidCrystal.h: Thư viện điều khiển màn hình LCD 16x2.

#define on HIGH

#define off LOW

- Định nghĩa on và off cho các thiết bị điều khiển (motor, valve).

Khai báo trạng thái (State Definitions):

#define startState 0

#define inflate1State 1

#define inflate2State 2

#define deflateState 3

#define displayState 4

#define resetState 5

- Các trạng thái của chương trình được định nghĩa. Mỗi trạng thái tương ứng với một giai đoạn trong quá trình đo huyết áp.

#define Sys_Measure 6

#define Sys_Cal 7

#define Rate_Measure 8

#define dias_Measure 9

#define dias_Cal 10

- Trạng thái đo lường và tính toán:
 - Sys_Measure: Đo tín hiệu áp suất tâm thu.
 - Sys_Cal: Tính toán giá trị tâm thu.
 - Rate_Measure: Đo nhịp tim.
 - dias_Measure: Đo tín hiệu áp suất tâm trương.
 - dias_Cal: Tính toán giá trị tâm trương.

Khai báo phần cứng:

#define valve A4

#define motor A3

#define bt_start A5

#define bt_stop A6

```
#define bt_resume A7
```

```
#define ADC0 A0
```

```
#define ADC1 A1
```

- Các chân của Arduino được gán với các thiết bị:
 - valve: Điều khiển valve xả khí.
 - motor: Điều khiển motor bơm khí.
 - bt_start, bt_stop, bt_resume: Các nút nhấn điều khiển.
 - ADC0, ADC1: Đọc tín hiệu cảm biến áp suất.

Biến toàn cục:

```
unsigned char currentState;
```

```
unsigned int timepress0, timepress1, timepress2, timelcd;
```

```
float DC_gain;
```

```
unsigned char meas_state;
```

```
unsigned int timing, timerate, timerun_dias, timecount, timedeflate, timedisplay;
```

```
float maxpressure, pressure, accum_data, press_data;
```

```
unsigned char count, stop_count;
```

- Các biến lưu trạng thái chương trình (currentState, meas_state) và thông tin đo lường (pressure, systolic, diastolic, pulse_per_min).

Phần setup()

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    pinMode(A5, INPUT);
```

```
    pinMode(A6, INPUT);
```

```
    pinMode(A7, INPUT);
```

```
    pinMode(motor, OUTPUT);
```

```
    pinMode(valve, OUTPUT);
```

```

lcd.begin(16, 2);

lcd.print("READY");

...
}

```

- Khởi tạo các thiết bị:
 - Giao tiếp Serial với tốc độ 9600 baud.
 - Cấu hình các chân INPUT (nút nhấn) và OUTPUT (motor, valve).
 - Khởi tạo LCD 16x2.

Cấu hình Timer

```

cli(); // Tắt ngắt toàn cục

TCCR1B |= (1 << WGM12) | (1 << CS11) | (1 << CS10); // Prescale = 64

OCR1A = 249; // Tạo ngắt mỗi 1ms

TIMSK1 = (1 << OCIE1A); // Cho phép ngắt Timer1

sei(); // Bật ngắt toàn cục

```

- Cấu hình **Timer1** để tạo ngắt mỗi 1ms.

Phần loop() - Vòng lặp chính

```

void loop() {

  switch(currentState)

  {

    case startState:

      start_state();

      break;

    case inflate1State:

      inflate1_state();

      break;

    case inflate2State:

      inflate2_state();

```

```

        break;

    case deflateState:
        deflatestate();
        break;

    case displayState:
        display_state();
        break;

    case resetState:
        reset_state();
        break;
}
}

```

- Sử dụng **state machine** để chuyển đổi giữa các trạng thái.

Ngắt Timer - ISR

```

ISR (TIMER1_COMPA_vect)
{
    if(digitalRead(bt_start)) timepress0++;
    if(analogRead(bt_stop) >= 900) timepress1++;
    if(analogRead(bt_resume) >= 900) timepress2++;
    ...
}

```

- **Ngắt Timer1:**
 - Đếm thời gian nhấn nút (timepress0, timepress1, timepress2).
 - Cập nhật các biến thời gian (timecount, timedeflate).

Các trạng thái chính

1. Trạng thái startState:

```

void start_state(void) {
    if (digitalRead(bt_start) && (timepress0 > 30)) {
        lcd.clear();
        lcd.print("Inflating");
        currentState = inflate1State;
        digitalWrite(valve, on);
        digitalWrite(motor, on);
    }
}

```

- Đợi nhấn nút bt_start để chuyển sang trạng thái bơm khí (inflate1State).

2. Trạng thái inflate1State và inflate2State:

```

void inflate1_state(void) {
    if (pressure >= maxpressure) {
        currentState = deflateState;
        digitalWrite(motor, off);
    }
}

```

- **inflate1State:** Bơm khí cho đến khi đạt áp suất tối đa.
- **inflate2State:** Tính toán áp suất từ tín hiệu ADC.

3. Trạng thái deflateState:

```

void deflatestate(void) {
    if (currentState == deflateState) {
        pressuremeasure();
    }
}

```

- Xả khí và đo huyết áp trong quá trình này.

4. Trạng thái displayState:

```
void display_state(void) {  
    lcd.clear();  
    lcd.print("Sys:");  
    lcd.print(systolic);  
    lcd.setCursor(0, 1);  
    lcd.print("Dias:");  
    lcd.print(diastolic);  
}
```

- Hiển thị kết quả đo trên LCD.

Các hàm đo lường

1. Hàm đo tín hiệu:

```
void read_adc(int Channel) {  
    data = analogRead(Channel);  
    adc_data = ((float)data / 1023) * Vref;  
}
```

- Đọc tín hiệu ADC và chuyển đổi sang đơn vị áp suất.

2. Hàm tính toán huyết áp:

```
void syscal(void) {  
    systolic = (press_data / DC_gain) * 6000;  
}
```

- Tính toán huyết áp tâm thu từ tín hiệu ADC.

3. Kết luận

- **Chương trình:** Quản lý toàn bộ quy trình đo huyết áp và nhịp tim, từ bơm khí, xả khí đến tính toán và hiển thị kết quả.

- **Thiết bị:** Motor, valve, cảm biến áp suất, màn hình LCD, nút nhấn.
- **Cải thiện:** Thêm xử lý lỗi, kiểm tra ngắt hoặc tối ưu hóa logic trạng thái.

Nếu

bạn cần giải thích sâu hơn từng phần hoặc chỉnh sửa, hãy cho tôi biết!