# PROJECT IOT102

# Tên project : Cảm biến đo, kiểm tra nhịp tim

| Nhóm 8:   * Bùi Tuấn Đạt – HE163093 * Phạm Tiến Hiệu – HE163449 * Nguyễn Đình Minh – HE151002 |
| --- |

## I. Ý tưởng

* Nhịp đập của tim là thông số rất quan trọng trong việc xây dựng 1 bài tập thể dục khoa học. Cảm biến nhịp tim chỉ thường xuất hiện trên các thiết bị đắt tiền như : máy chạy bộ, máy đo trong bệnh viện, smartphone... Giờ đây đã xuất hiện 1 loại cảm biến đo nhịp tim mà chúng ta dễ dàng kết nối với các bo mạch điện tử như Arduino, Raspberry Pi và các MCU khác.
* Cảm biến hoạt động dựa trên 1 sensor quang đo nhịp tim cùng với bộ khuếch đại tín hiệu và lọc nhiễu được nhóm mình chọn sử dụng để triển khai ý tưởng làm và sử dụng cảm biến đo , kiểm tra nhịp tim với những bước cơ bản ban đầu.

## II. Nội dung

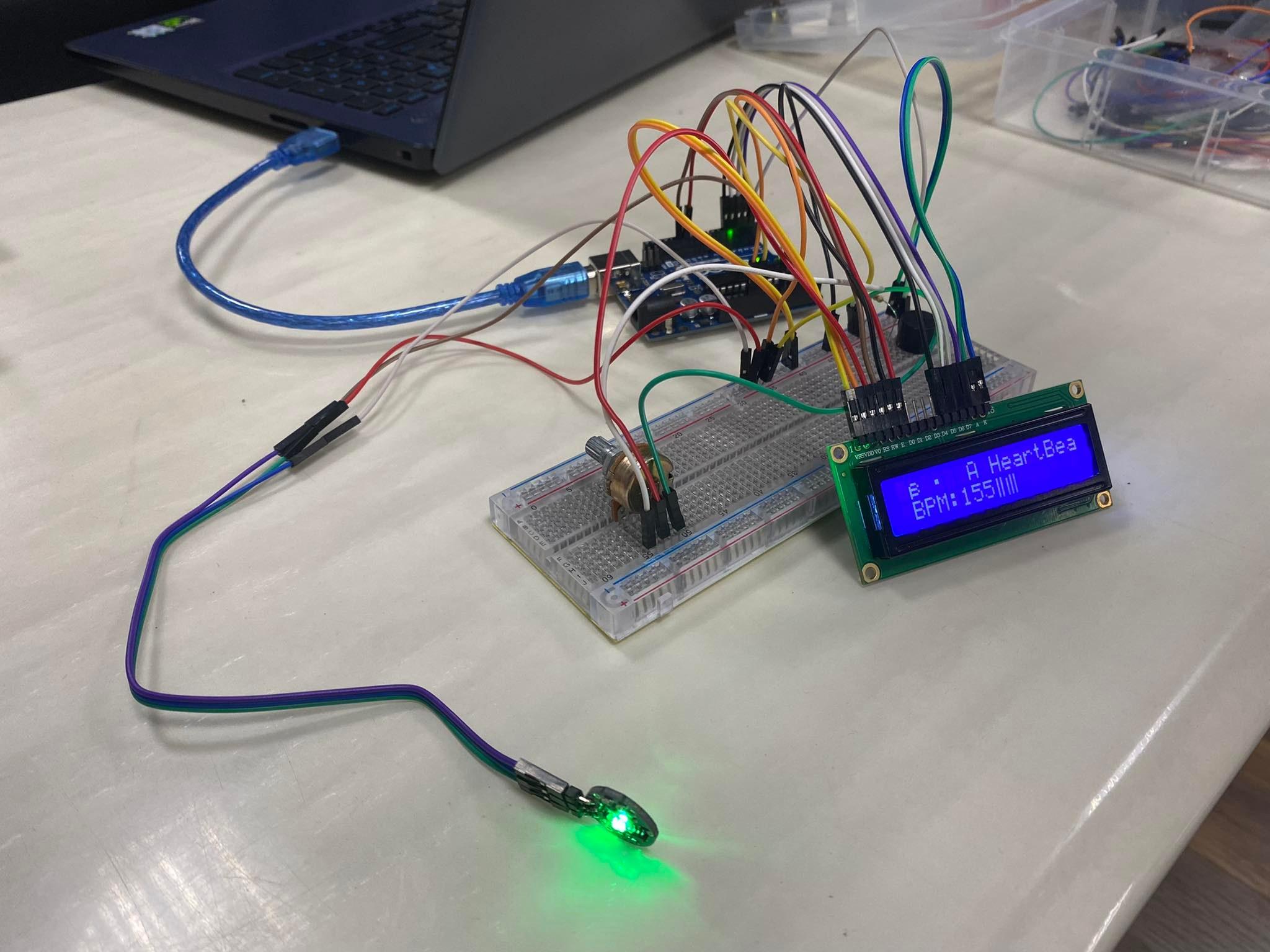
### 1. Linh kiện (Hardware Required )(Bắt buộc phải có)

* Arduino Uno R3
* LED Display LCD 16 x 2
* Module Cảm biến Nhịp Tim Pulse Sensor.
* Breadboard.
* Dây nối.
* Còi chip (buzzer)

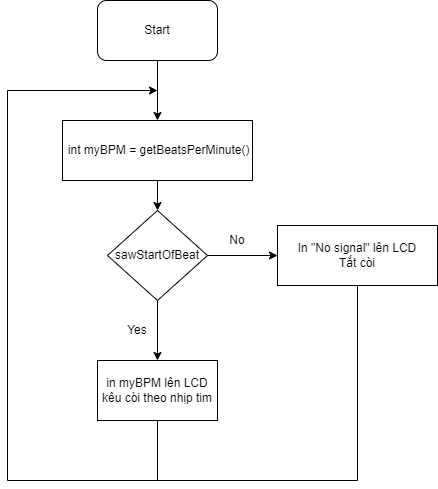
### 2. Mạch nguyên lý (Schematic)

### 3. Mạch kết nối (Circuit )

Mạch mô phỏng hoặc mạch thực tế. (Bắt buộc phải có)



### 4. Lưu đồ thuật toán (Flow chart)

Lưu đồ giải thuật của hàm **void loop** (Bắt buộc phải có)

## III. Kết quả đạt được (Bắt buộc phải có)

* Nguyên lý hoạt động: Đầu tiên ta chạm đầu ngón tay vào cảm biến nhịp tim, khi tim đập và trương lên, mật độ máu trong mạch máu giảm, các chùm tia hồng ngoại chiếu vào ngón tay bị phản xạ nhưng không nhiều, đầu ra của cảm biến nhịp tim sẽ hiểu đây là một xung mức thấp. Khi tim đập và co lại, mật độ máu trong mạch máu tăng, các chùm tia hồng ngoại chiếu vào ngón tay bị phản xạ nhiều hơn, đầu ra của cảm biến nhịp tim sẽ hiểu đây là một xung mức cao. Tuy sự chênh lệch trong sự phản xạ tia hồng ngoại không lớn nhưng Arduino có thể phân biệt và hiểu được đâu là xung mức cao, đâu là xung mức thấp, nhờ đó chúng ta có thể theo dõi nhịp tim bằng cảm biến.
* Kết quả thực hiện : Ứng dụng cảm biến trên sẽ giúp chúng ta hiển thị được thông số BPM (nhịp đập tim của người dùng) trực tiếp và thay đổi trên màn hình LCD được nối tiếp Arduino . Màn hình Serial cũng sẽ in ra: ♥ A HeartBeat Happened ! khi một nhịp đập được phát hiện và tiếng còi sẽ bíp có tốc độ tương tự với mạch đập tim.  
  còn khi không có nhịp đập, màn hình sẽ hiện ra “No signal”.
* IV. Phụ lục

### 1. Link Tinkercad

* Link mô phỏng
* Link video (nếu có)

### 2. Code (phải có)

#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.

#include <LiquidCrystal.h>

//  Variables

const int PulseWire = A0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0

const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.

int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.

// Use the "Gettting Started Project" to fine-tune Threshold Value beyond default setting.

// Otherwise leave the default "550" value.

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

PulseSensorPlayground pulseSensor;  // Creates an instance of the PulseSensorPlayground object called "pulseSensor"

void setup() {

  Serial.begin(9600);          // For Serial Monitor

  pinMode(1, OUTPUT);

  // Configure the PulseSensor object, by assigning our variables to it.

  pulseSensor.analogInput(PulseWire);

  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino's LED with heartbeat.

  pulseSensor.setThreshold(Threshold);

  // Double-check the "pulseSensor" object was created and "began" seeing a signal.

  if (pulseSensor.begin()) {

    Serial.println("We created a pulseSensor Object !");  //This prints one time at Arduino power-up,  or on Arduino reset.

  }

  pinMode(7, OUTPUT);

  lcd.begin(16, 2);

}

void loop() {

  int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that returns BPM as an "int".

  // "myBPM" hold this BPM value now.

  if (pulseSensor.sawStartOfBeat()) {            // Constantly test to see if "a beat happened".

    Serial.println("♥  A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat happened".

  Serial.print("BPM: ");                        // Print phrase "BPM: "

  Serial.println(myBPM);  // Print the value inside of myBPM.

    lcd.setCursor(0, 0);

    lcd.print("A HeartBeat! "); // If test is "true", print a message "a heartbeat happened".

    lcd.setCursor(1, 1);

    lcd.print("BPM: ");                        // Print phrase "BPM: "

    lcd.setCursor(5, 1);

    lcd.println(myBPM);  // Print the value inside of myBPM.

    int val = map(myBPM, 0, 250, 500, 5);

    digitalWrite(7, LOW);

    delay(val);

    digitalWrite(7, HIGH);

    delay(val);

  }

  else {

    digitalWrite(7, LOW);               //  Else, the signal must be below "550", so "turn-off" this LED.

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("No signal");

  }

  delay(20);                    // considered best practice in a simple sketch.

}