**BỘ GIÁO DỤC VÀ ĐÀO TẠO TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**---------------------------------------**



**Báo cáo bài tập lớn**

|  |
| --- |
| **XÂY DỰNG CÁC HỆ THỐNG NHÚNG** |
|  |
|  |
| **ĐỀ TÀI: HỆ THỐNG ĐO NHỊP TIM THỜI GIAN THỰC** |
| Giảng viên hướng dẫn: TS.GVC. Nguyễn Xuân Sâm  Lớp: D14CQCP01-N  Thành viên: Trần Hoàng Minh N14DCCN116  Nguyễn Ngọc Bảo Thy N14DCCN145  Trần Mạnh Hùng N14DCCN030 |
| **Hồ Chí Minh, Tháng 04/2018** |

MỤC LỤC

[**CHƯƠNG 1. PHÂN TÍCH BÀI TOÁN** 3](#_Toc512764813)

[1.1. Khảo sát và phân tích bài toán: 3](#_Toc512764814)

[1.1.1 Các cách đo nhịp tim: 3](#_Toc512764815)

[1.1.2 Mục đích: 6](#_Toc512764816)

[1.1.3 Các công nghệ nhúng có mặt trên thị trường: 7](#_Toc512764817)

[1.2. Lựa chọn giải pháp: 11](#_Toc512764818)

[1.2.1. Giải pháp công nghệ: 11](#_Toc512764819)

[1.2.2. Giải pháp thiết kế: 11](#_Toc512764820)

[1.2.3. Các yêu cầu: 13](#_Toc512764821)

[1.2.4. Giới hạn cho hệ thống: 14](#_Toc512764822)

[1.2.5. Một vài Lưu ý về hệ thống: 14](#_Toc512764823)

[**CHƯƠNG 2. THIẾT KẾ HỆ THỐNG** 15](#_Toc512764824)

[2.1. Sơ đồ tổng quát: 15](#_Toc512764825)

[2.2. Sơ đồ đặc tả: 15](#_Toc512764826)

[2.3. Các module trong hệ thống: 16](#_Toc512764827)

[2.3.1. Khối Sensor: 16](#_Toc512764828)

[2.3.2. Khối Xử lý: 17](#_Toc512764829)

[2.3.2. Khối Network: 17](#_Toc512764830)

[2.3.3. Khối lưu trữ 19](#_Toc512764831)

[2.3.4. Khối hiển thị 19](#_Toc512764832)

[**CHƯƠNG 3: XÂY DỰNG HỆ THỐNG** 20](#_Toc512764833)

[3.1. Thiết kế phần cứng: 20](#_Toc512764834)

[3.2. Thiết kế phần mềm: 20](#_Toc512764835)

[3.3. Mạch thi công thực tế 27](#_Toc512764836)

[KẾT LUẬN 29](#_Toc512764837)

[Các trang web tham khảo: 29](#_Toc512764838)

# CHƯƠNG 1. PHÂN TÍCH BÀI TOÁN

## Khảo sát và phân tích bài toán:

**Nhịp tim** là số lần tim đập trên mỗi phút. Nó phụ thuộc vào từng cá nhân, tuổi tác; trọng lượng cơ thể; trạng thái hoạt động như ngồi yên hay di chuyển; các bệnh lý mắc phải; các thuốc đang sử dụng, thậm chí nhiệt độ không khí cũng có thể ảnh hưởng tới nhịp tim. Một yếu tố gắn liền với chúng ta hàng ngày và có tác động đến nhịp tim một cách rõ ràng, dễ nhận biết nhất đó chính là cảm xúc. Khi chúng ta bị kích thích hay sợ hãi, vui mừng hay lo lắng đều có thể làm tăng nhịp tim.

### 1.1.1 Các cách đo nhịp tim:

Đo nhịp tim bằng ngón tay :

* Cách tìm nhịp tim: Áp sát mặt trong của 2 ngón tay bên này vào mặt trong của cổ tay bên kia – chỗ có những nếp gấp cổ tay (hai tay ngược nhau). Bấm nhẹ vào đó cho đến khi bạn cảm thấy mạch đập. Nếu cần thiết, có thể di chuyển ngón tay xung quanh đó cho đến khi bạn cảm thấy nhịp đập.



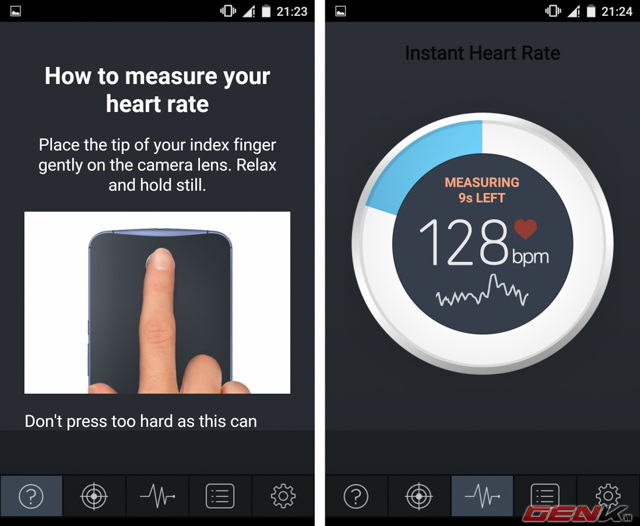
Đo nhịp tim bằng ống nghe y tế:

* Cách tìm nhịp tim: Áp hai nút ống nghe vào hai lỗ tai, trục dọc của nút tai phải khớp với trục dọc của ống tai ngoài.
* Chọn màng hay chuông tùy theo âm mà ta muốn nghe.
* Khi nghe bằng màng cần ấn chặt màng trên da.
* Khi sử dụng chuông chỉ cần ấn nhẹ lên da.



Đo nhịp tim bằng thiết bị điện thoại với ứng dụng hỗ trợ đo nhịp tim:

* Cơ chế hoạt động của phần mềm là quan sát sự thay đổi màu sắc trên đầu ngón tay, sau đó ứng dụng sẽ dùng các thuật toán riêng để đưa ra kết quả nhịp tim gần đúng nhất với người dùng. VD: Instant Heart Rate, Runtastic Heart Rate Monitor, Heart Rate Plus.



Máy đo nhịp tim chuyên dụng trong y học:

* Máy đo khí máu và nhịp tim đo độ bão hòa oxy máu theo mạch đập hoạt động dựa trên cơ sở phép đo quang phổ kế (sắc ký) và phép đo xung động kế (xung động ký).
* Phép đo sắc ký hoạt động dựa trên cơ sở độ hấp thụ ánh sáng của Hemoglobin (Hb) và Oxyhemoglobin (HbO2) khác nhau đối với 2 bước sóng khác nhau của ánh sáng đỏ (660nm) và ánh sáng hồng ngoại (905nm).
* Phép đo xung động ký hoạt động dựa trên cơ sở độ hấp thu ánh sáng truyền qua mô thay đổi có tính chu kỳ do sự thay đổi thể tích máu giữa kỳ tâm thu và tâm trương.



Hệ thống đo nhịp tim kết hợp database realtime:

* Có khả năng lưu trữ dữ liệu đã đo được lên server, đồng thời nhận được thông qua nền tảng web và điện thoại. Vẽ biểu đồ thống kê theo BPM (Beat per minute) realtime, điều chỉnh độ nhạy sensor thông qua ThresHold, đưa ra nhận xét về sức khỏe và tình trạng tim mạch.

### 1.1.2 Mục đích:

* Hiện nay, các thiết bị đo từ thủ công đến tự động đều chỉ dừng lại ở mức đo và đưa ra chỉ số nhịp tim mà không lưu lại hoặc theo dõi tình trạng sau mỗi lần đo.
* Việc đưa dữ liệu đo đạc được lên server sẽ dễ dàng lưu trữ và là nguyên liệu để phân tích đánh giá tình trạng sức khỏe. Việc theo dõi nhịp tim thường xuyên và duy trì sẽ giúp đánh giá tình trạng tốt hơn và chính xác hơn.
* Ngày nay, các ứng dụng điện thoại và trang web ngày càng phổ biến và linh hoạt hơn. Do đó, việc lấy dữ liệu nhịp tim và hiển thị trên các nền tảng này sẽ trực quan và dễ dàng hơn.

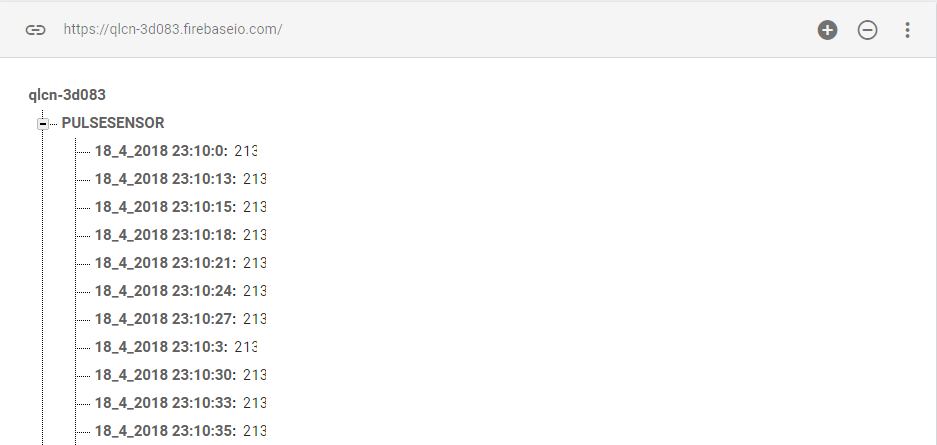
****

Figure 1: Trên Firebase database realtime

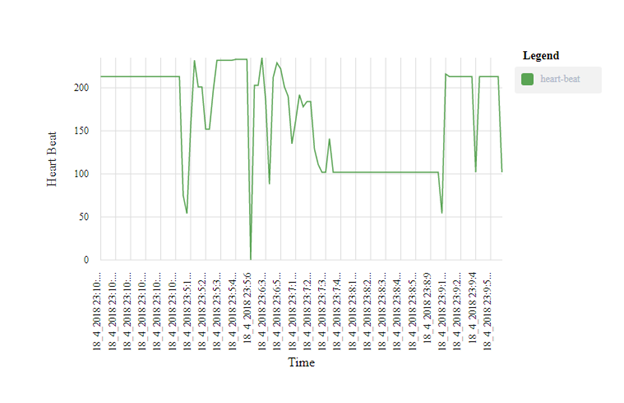
****

Figure 2: Biểu đồ trên nền tảng web

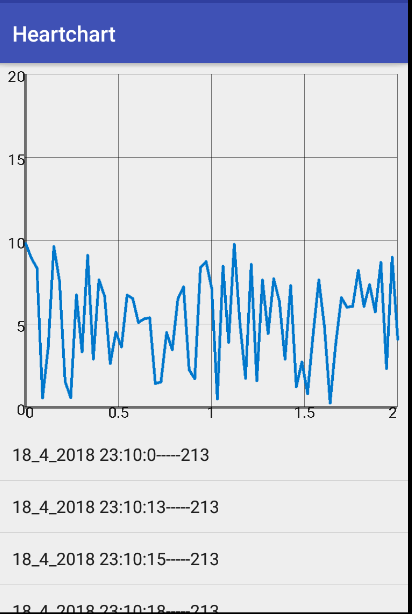
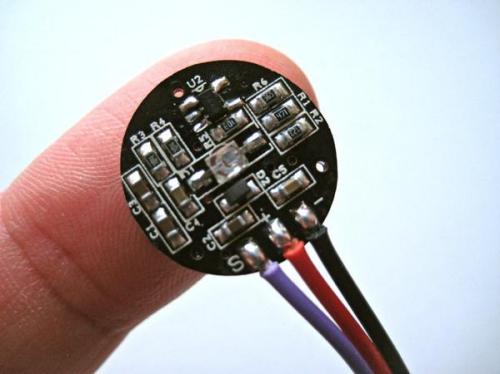
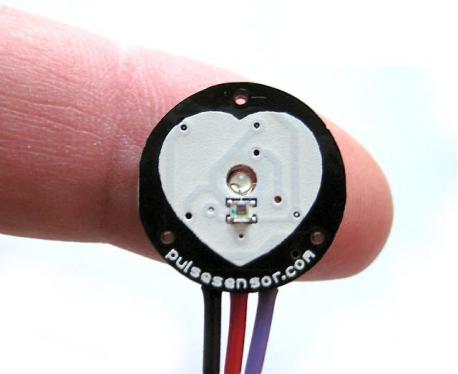


Figure 3: Biểu đồ trên nền tảng mobile

### 1.1.3 Các công nghệ nhúng có mặt trên thị trường:

#### Sensor:

* Hiện nay, trên thị trường đang sử dụng sensor đo nhịp tim phổ biến của pulsesensor.com. Cảm biến nhịp tim dạng quang Pulse Sensor sử dụng nguyên lý đo nhịp tim bằng ánh sáng với kích thước nhỏ gọn và giao tiếp Analog rất dễ sử dụng, phù hợp cho các ứng dụng điện tử y sinh.
* Thông số kỹ thuật:
  + Nguồn : 3~5VDC
  + Dòng tiêu thụ : < 4mA
  + Ngõ ra : Analog.
  + Độ dài dây : 61cm ( 24 inch).
  + Đường kính cảm biến : 1.6 cm ( 0.625 inch).



#### Vi điều khiển:

* Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà người ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3).

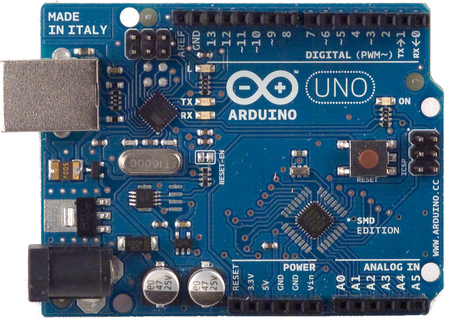
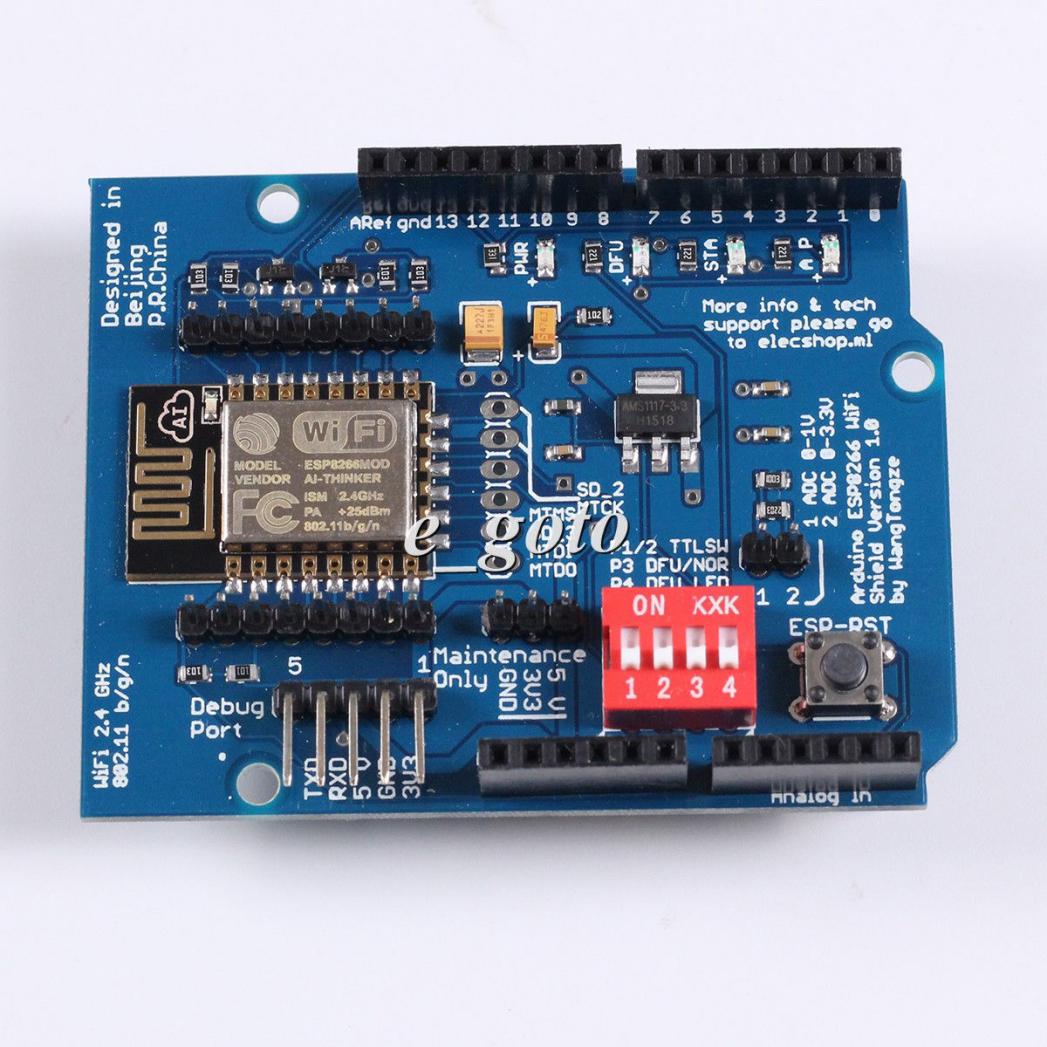


Figure 4: Thông số kỹ thuật

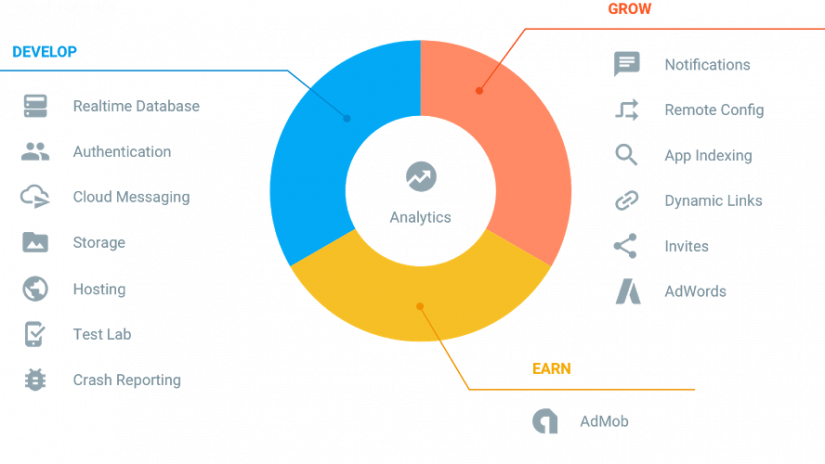
#### Hệ thống mạng:

* Arduino ESP8266 Wifi Shield được thiết kế dạng Shield Arduino nên có thể kết nối dễ dàng với hầu hết các board Arduino hiện nay để thực hiện các ứng dụng với module wifi esp8266 như bật tắt thiết bị qua wifi, thu thập dữ liệu qua wifi, báo động wifi, và hầu hết các ứng dụng IoT hiện nay.
* Thông số kỹ thuật:
  + Arduino ESP8266 Wifi Shield
  + Module chính: Wifi SoC ESP8266 V12E
  + Thiết kế dạng Shield chuẩn chân Arduino.
  + Cấp nguồn từ các chân nguồn của Arduino.
  + Pin Out đầy đủ các chân của ESP8266 V12E.
  + Tích hợp Switch thiết lập chế độ giao tiếp với Arduino, chế độ nạp Firmware.
  + Tích hợp các Led PWR, DFU, AP, STA.



#### Hệ thống Server Database Realtime:

* Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.
* Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database tới tất cả các client theo thời gian thực. Cụ thể hơn là bạn có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.



#### Nền tảng dùng để hiện thị và phân tích dữ liệu:

* Web platform: Angular  
  - Angular là phiên bản tiếp theo của framework phát triển dựa trên JavaScript của Google.  
  - Angular hỗ trợ dễ dàng kết nối với firebase hơn trang web thông thường sử dụng jquery.

- Lấy dữ liệu dưới dạng json hoặc object.

* Mobile platform: Lấy dữ liệu từ Firebase về dạng JSON hoặc Object .Từ dữ liệu đó dùng để vẽ biểu đồ nhịp tim trên điện thoại.

## 1.2. Lựa chọn giải pháp:

### 1.2.1. Giải pháp công nghệ:

* Qua phân tích ở trên, nhóm chúng em đưa ra giải pháp xây dựng hệ thống đo nhịp tim và lưu trữ dữ liệu trên server thời gian thực. Đọc giờ từ Service thời gian thực, hiển thị dữ liệu nhịp tim liên tục và có thể cài đặt được Threshold. Ứng dụng hỗ trợ trên nhiều nền tảng khác nhau như web hoặc smartphone.

### 1.2.2. Giải pháp thiết kế:

* Việc lựa chọn giải pháp thường được xem xét trên nhiều phương diện nhưng quan trọng là giải pháp có khả thi không? Có phù hợp với thực tế và thỏa mãn yêu cầu về kinh tế?
* Hệ thống đo nhịp tim thời gian thực với bộ não điều khiển là ATmega328 và các linh kiện, công nghệ khác: cảm biến đo nhịp tim, ESP8266 wifi shield, Firebase, Angular, Android.
  + **Arduino Uno R3:** với vi điều khiển Atmega328 là loại phổ biến và dễ sử dụng nhất trong các dòng Arduino hiện nay cũng như tương thích với nhiều loại Arduino Shield nhất. Với khả năng lập trình cho các ứng dụng điều khiển phức tạp do được trang bị cấu hình mạnh cho các loại bộ nhớ ROM, RAM và Flash, các ngõ vào ra digital I/O trong đó có nhiều ngõ có khả năng xuất tín hiệu PWM, các ngõ đọc tín hiệu analog và các chuẩn giao tiếp đa dạng như UART, SPI, TWI (I2C).
  + **Cảm biến đo nhịp tim:** hình dạng nhỏ, nhẹ, độ chính xác khá cao cùng giá thành khá thấp sẽ là lựa chọn phù hợp cho hệ thống đo nhịp tim.
  + **ESP8266 wifi shield:** là board mạch ứng dụng, tích hợp module wifi ESP8266-07. Board được thiết kế giúp cho người sử dụng dễ dàng tìm hiểu, lập trình ESP8266, ứng dụng để làm ra những sản phẩm điều khiển được thiết bị từ xa bằng smartphone hay máy tính qua wifi.
  + **Firebase:** Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database tới tất cả các client theo thời gian thực. Cụ thể hơn là bạn có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.
  + **Angular:** Ứng dụng web cho phéplấy dữ liệu từ firebase dưới dạng object và hiển thị biểu đồ nhịp tim theo thời gian thực, tùy chỉnh thông số trên firebase.

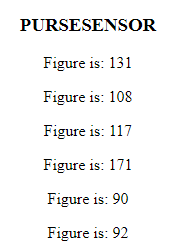


Figure 5: Giá trị BPM lấy được trên web

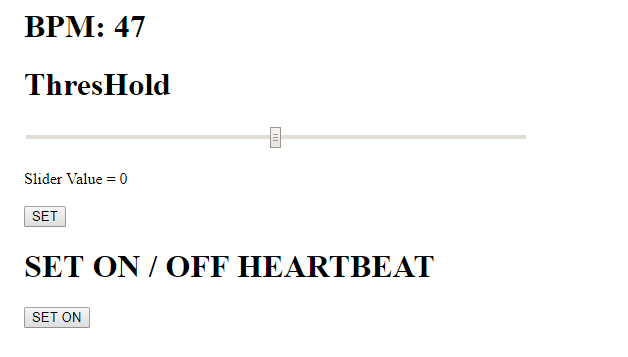


Figure 6: Giao diện Web với ThresHold và bật tắt biểu đồ

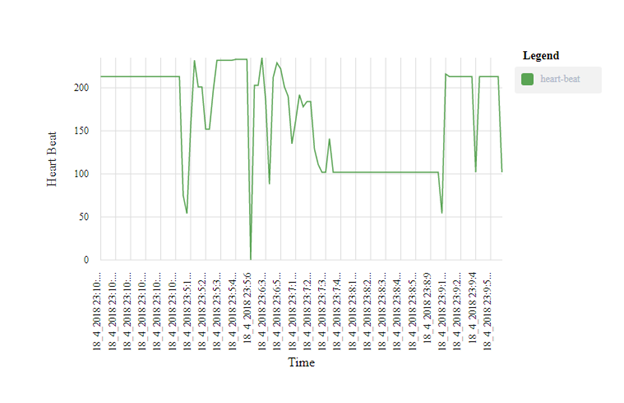
****

Figure 7: Biểu đồ nhịp tim trên giao diện Web

* + **Android:** ứng dụng trên mobile cho phép lấy dữ liệu từ firebase dưới dạng object (key,value) và hiển thị biểu đồ theo thời gian thực.

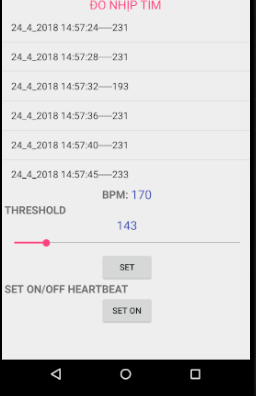


Figure 8: Giao diện Mobile với Threshhold,BPM, và bật tắt biểu đồ cùng với listview hiển thị dữ liệu nhịp tim từ server

Vì vậy giải pháp thiết kế hệ thống đo nhịp tim thời gian thực dùng các linh kiện trên có nhiều ưu thế hơn so với những giải pháp khác. Đồng thời đảm bảo được yêu cầu về kinh tế.

### 1.2.3. Các yêu cầu:

Với hệ thống đo nhịp tim thời gian thực đòi hỏi các yêu cầu:

• Hiển thị đúng thời gian: ngày, tháng, năm, giờ, phút, giây.

• Điều chỉnh và thay đổi được Threshold.

• Đảm bảo đúng về thời gian thực.

• Khả năng thực thi:Thời gian đáp ứng, độ chính xác, độ tin cậy…

• Đảm bảo về kích thước và trọng lượng cho phép.

• Độ an toàn, khả năng chống lại sự phá hoại hay xâm nhập…

### 1.2.4. Giới hạn cho hệ thống:

• Sử dụng nguồn điện 5V.

• Kích thước phù hợp với người sử dụng.

• Hệ thống nhỏ gọn.

• Hệ thống lưu được dữ liệu lên Server.

• Làm việc trong điều kiện môi trường bình thường

### 1.2.5. Một vài Lưu ý về hệ thống:

* Nếu xuất hiện một vài rắc rối về hiển thị nhịp tim, hãy đảm bảo rằng sensor nhịp tim được áp vào da gần mạch máu một lực vừa đủ: không quá mạnh, không quá yếu. Sử dụng lực quá mạnh sẽ làm nhịp tim biến mất, lực quá yếu sẽ bị nhiễu do môi trường xung quanh.
* Nếu hệ thống đọc quá nhiều BPM, hoặc bị nhiễu, hãy thử điều chỉnh giá trị ThresHold. Giá trị ThresHold có thể nằm trong khoảng 0 – 1000, nhưng hãy cố gắng chỉ điều chỉnh mỗi 5 hoặc 10 đơn vị. Giảm ThresHold sẽ gia tăng độ nhạy, tăng ThresHold sẽ giảm độ nhạy của cảm biến.

# CHƯƠNG 2. THIẾT KẾ HỆ THỐNG

## 2**.1. Sơ đồ tổng quát:**

Khối hiển thị

Kết nối mạng

Khối lưu trữ

Sensor

Khối network

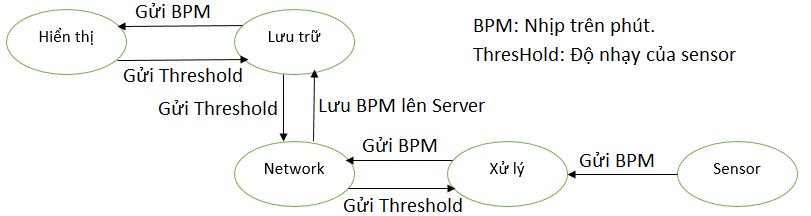
Giao tiếp Serial

Khối xử lý

Kết nối mạng

* Sensor: cảm biến đo nhịp tim.
* Khối xử lý: Arduino Uno R3 với Atmega328.
* Khối network: ESP8266 wifi shield.
* Khối lưu trữ: Firebase với database realtime.
* Khối hiển thị: nền tảng web với Angular, nền tảng mobile với Android

## 2.2. Sơ đồ đặc tả:



## 2.3. Các module trong hệ thống:

### 2.3.1. Khối Sensor:

* Đầu tiên đặt sensor đo nhịp tim vào tay, sensor sẽ dùng quang kế để cảm nhận nhịp tim và biến đổi sang tín hiệu digital gửi sang khu xử lý. Dữ liệu được giới hạn từ 0 đến 1000 và là dữ liệu thô chưa được xử lý.

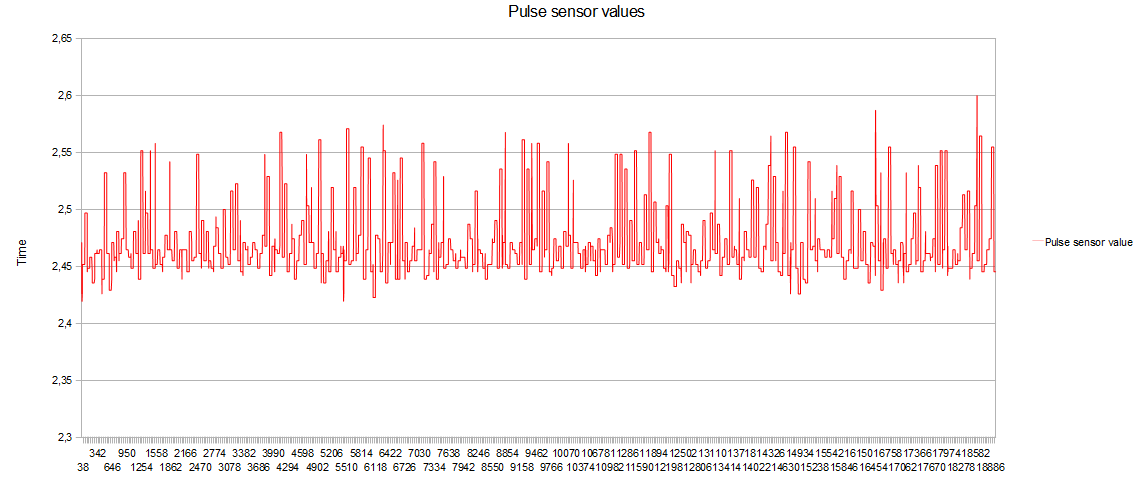


Figure 9: Giá trị raw lấy từ sensor

* Sensor sử dụng điện áp 5v gồm 3 chân: data, vcc, gnd. Trong đó chân vcc cắm vào chân cấp điện 5v, chân gnd cắm vào chân gnd của Arduino, chân data cắm vào cổng A0 ở analog in của Arduino

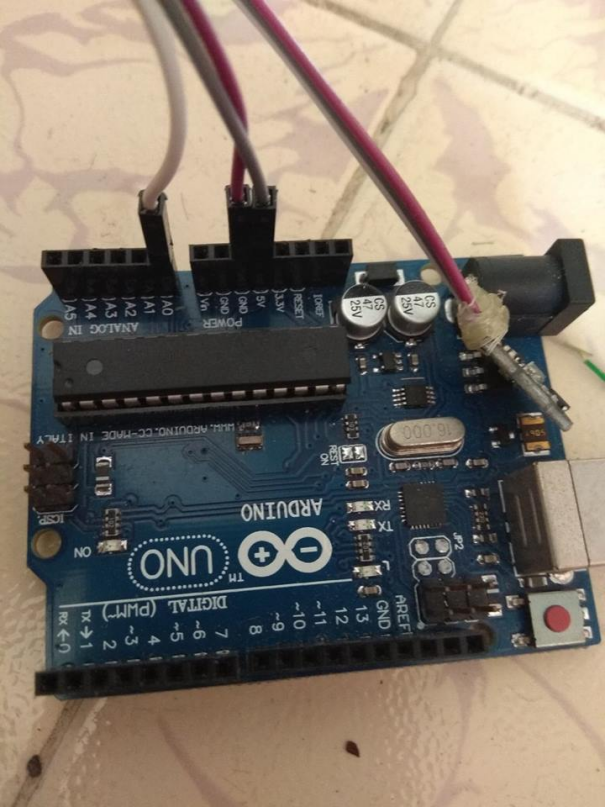


Figure 10: Arduino và các chân cắm

### 2.3.2. Khối Xử lý:

* Khối điều khiển trung tâm sử dụng vi điều khiển Atmega328, qua chương trình đã lập trình được nạp cho chip, vi điều khiển sẽ điều khiển việc đọc dữ liệu từ sensor, hiển thị dữ liệu lên máy tính qua cổng Serial và gửi dữ liệu sang khối network.

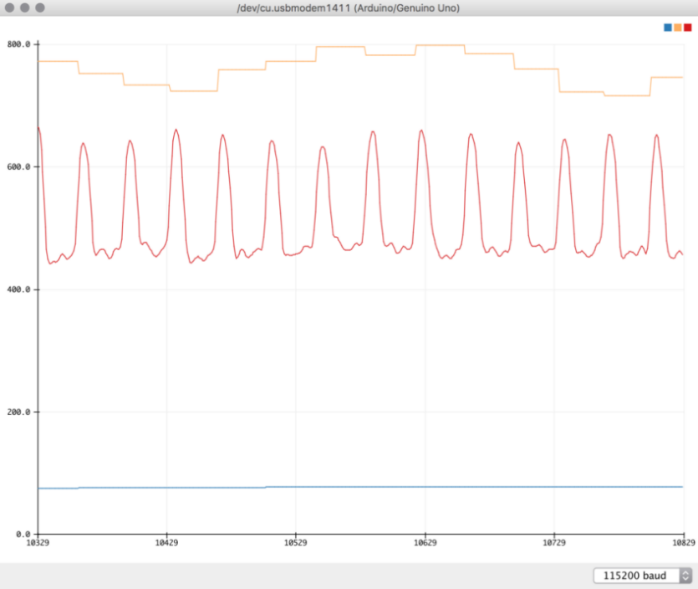


Figure 11: Hình ảnh sau khi xử lý dữ liệu raw từ bộ thư viện

### 2.3.2. Khối Network:

* ESP8266 dùng chân cắm Tx, Rx, nguồn 5v và chân Gnd để nạp code và giao tiếp với Arduino. Ngoài ra hộp div dùng để tùy chỉnh các chế độ như flash, Serial communication,…

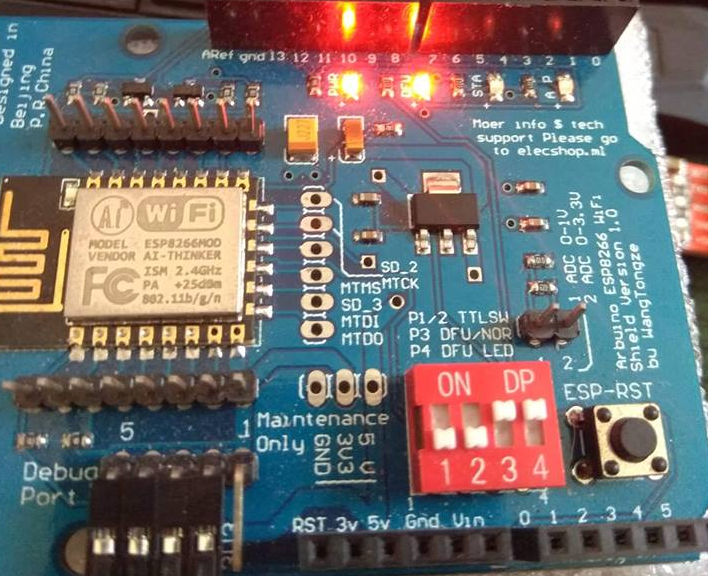


Figure 12: Chế độ nạp code với các switch tương ứng Down Down Up Up và đèn DFU sẽ bật sáng.

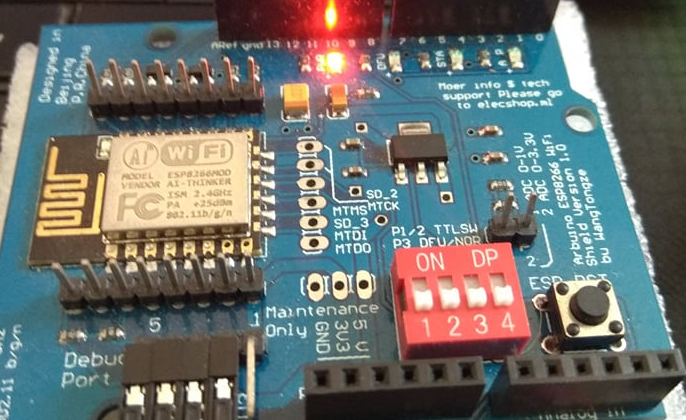


Figure 13: Với 4 switch tương ứng đều Down sẽ chuyển sang trạng thái giao tiếp với Arduino

* Để có thể nạp code, cần dùng một USART để cắm vào cổng USB của máy tính. Cần chú ý việc cắm các chân Tx, Rx với sơ đồ chéo để đảm bảo đầu gửi sẽ đc gửi tới đầu nhận và ngược lại.

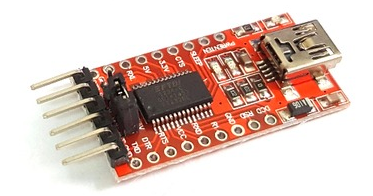


Figure 14: Mạch Chuyển USB UART TTL FT232RL

* Khối network sử dụng module ESP8266 wifi shield với chuẩn 802.11b/g/n, qua chương trình đã lập trình được nạp cho chip, ESP8266 sẽ gửi dữ liệu từ Arduino uno r3 lên Firebase bằng bộ thư viện Firebase, đồng thời nhận dữ liệu từ Firebase về Threshold và gửi sang cho Arduino.

### 2.3.3. Khối lưu trữ

* Database được lưu dưới dạng cây và gồm các key theo thời gian thực. Tất cả bao gồm 2 key chính là PURSESENSOR và ThresHold. Firebase sẽ gửi dữ liệu theo loại json nên được rất nhiều nền tảng hỗ trợ.
* Ngoài ra Firebase còn bảo mật với các rule siết chặt.

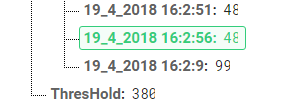
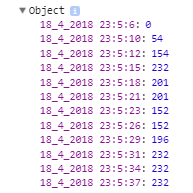


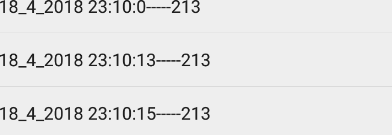
Figure 15: Dữ liệu đang được lưu

### 2.3.4. Khối hiển thị

* Angular**:** Hiển thị dưới dạng object gồm key và value.



* Android: Hiển thị dữ liệu dưới dạng object gồm key và value



# CHƯƠNG 3: XÂY DỰNG HỆ THỐNG

## 3.1. Thiết kế phần cứng:

* Hệ thống sẽ gồm 4 khối như đã phân tích ở chương hai, ta sẽ có sơ đồ nguyên lý như sau:

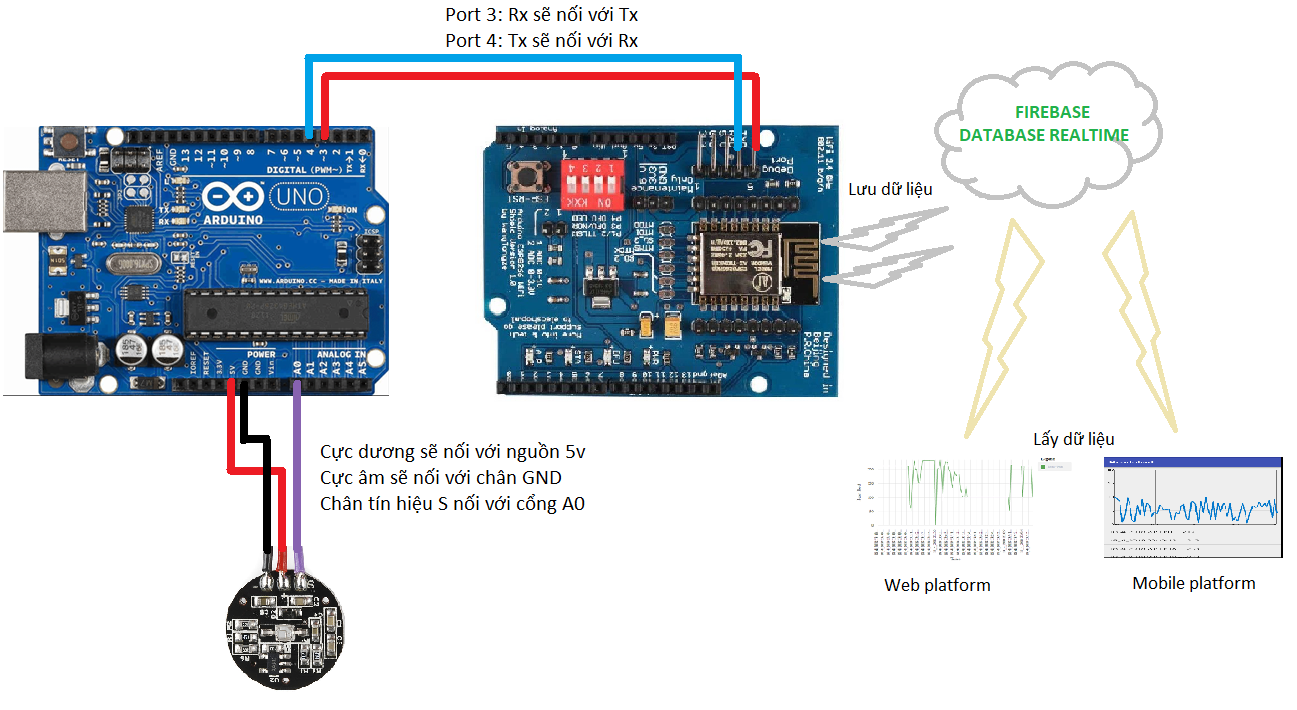
****

Figure 16: Mô hình hệ thống đo nhịp tim thời gian thực

## 3.2. Thiết kế phần mềm:

* Phần mềm cần thực hiện các chức năng chính như sau :
  + Hiển thị nhịp tim theo thời gian thực:
    - Arduino uno r3: khi bắt đầu chạy, Arduino đọc tín hiệu từ sensor nhịp tim dùng bộ thư viện “PulseSensorPlayground” để xử lý dữ liệu thô thành dạng BPM có thể sử dụng được và gửi dữ liệu cho ESP8266. Đồng thời arduino còn nhận giá trị Threshold dùng để chỉnh độ nhạy sensor từ ESP8266 và thiết lập cho Threshold của thư viện.
    - ESP8266: khi bắt đầu chạy, ESP8266 thiết lập các giá trị ban đầu và kết nối đến wifi đã được cài đặt. Khi wifi đã sẵn sàng, ESP8266 sẽ nhận giá trị được gửi từ Arduino và kiểm tra dữ liệu có rỗng không, nếu không ESP8266 sẽ gửi request nhận thời gian hiện tại từ Service trong bộ thư viện NTPtimeESP để lưu trữ lên Firebase. Nếu trạng thái được phép lưu (isOn = 1) ESP8266 sẽ push dữ liệu gồm Time + data lên Firebase liên tục, nếu không (isOn = 0) sẽ chỉ có data được lưu ở biến “BPM” tạm thời trên Firebase để hiển thị.
* Cài đặt Threshold: khi các thiết bị hiển thị có sự thay đổi Threshold, dữ liệu sẽ được thay đổi trên Firebase. ESP8266 sẽ liên tục nhận dữ liệu Threshold về và chuyển cho Arduino để sử dụng. Arduino nhận được Threshold sẽ cài đặt cho Threshold của bộ thư viện.

**CHƯƠNG TRÌNH**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//ARDUINO UNO R3

#include <SoftwareSerial.h> //Khai báo thư viện tạo cổng kết nối ảo

SoftwareSerial mySerial(3, 4); //Dùng cổng PWM 3 là Rx, 4 là Tx giao tiếp với

//ESP8266 và dùng cổng mặc định để debug

//trên máy tính

#define USE\_ARDUINO\_INTERRUPTS true //Cho phép ngắt trên arduino để

//dừng lệnh hiện tại và nhảy vào lệnh trong ngắt

#include <PulseSensorPlayground.h> //Khai báo thư viện đo nhịp tim

PulseSensorPlayground pulseSensor; //khởi tạo struct

const int PulseWire = 0; //Khai báo cổng analog Pin 0 làm cổng lấy tín hiệu

const int LED13 = 13; //Led trên Arduino R3 để show nhịp tim

int Threshold = 550; //Ngưỡng để xác định nhịp tim sẽ được đếm

const String KEY\_PULSE = "PULSESENSOR"; //Tên lưu trên Firebase

void setup() {

Serial.begin(115200); //Bắt đầu baud rate Serial 115200

mySerial.begin(115200);//Bắt đầu baud rate serial giao tiếp với ESP8266 115200

pulseSensor.analogInput(PulseWire); //Cài đặt analogInput là cổng 2

pulseSensor.blinkOnPulse(LED13); //Nháy đèn khi có nhịp tim tại led 13

pulseSensor.setThreshold(Threshold); //Cài đặt threshold

if (pulseSensor.begin()) { //Khi sensor nhịp tim được khởi động sẽ in ra màn hình

Serial.println("We created a pulseSensor Object !");

}

}

void loop() {

int myBPM = pulseSensor.getBeatsPerMinute(); //Lấy dữ liệu nhịp tim trên phút

if (pulseSensor.sawStartOfBeat()) {//Nếu "thấy" nhịp tim thì xuất ra màn hình

Serial.println("♥ A HeartBeat Happened ! ");

Serial.print("BPM: ");

Serial.println(myBPM);

Serial.print("Threshold: ");

Serial.println(Threshold);

}

sendESP((String)myBPM); //Gửi dữ liệu sang ESP8266 gửi lên Firebase

String data = receiveESP(); //Nhận dữ liệu Threshold về để điều chỉnh lại

if (data!="") { //Kiểm tra dữ liệu rỗng

Threshold = data.toInt(); //Ép kiểu int để set cho Threshold

pulseSensor.setThreshold(Threshold); //Cài đặt threshold

}

}

void sendESP(String data) { //Hàm gửi dữ liệu sang ESP8266

mySerial.println(data); //Gửi và delay 1s

delay(1000);

}

String receiveESP() { //Hàm nhận dữ liệu từ ESP8266

String buff = ""; //Tạo biến chuỗi để nhận từng kí tự thêm vào

if (mySerial.available()) { //Kiểm tra cổng kết nối sẵn sàng

while (mySerial.available()) {//Lặp đến khi cổng kết nối hết nhận dữ liệu

char data = (char)mySerial.read();//Đọc kí tự từ cổng kết nối

buff += data; //Lưu vào biến String đệm

}

}

return buff; //Trả kết quả

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//ESP8266 Wifi shield v1.0

//ESP8266 Wifi shield v1 dùng board NodeMCU 1.0

//Upload speed 115200

#include <NTPtimeESP.h> //Thư viện kết nối service thời giàn

#include <ESP8266WiFi.h> //Thư viện cho esp8266 wifi shield

#include <FirebaseArduino.h>//Thư viện giao tiếp với Firebase

#define FIREBASE\_HOST "qlcn-3d083.firebaseio.com" //Đường dẫn CSDL Firebase

#define FIREBASE\_AUTH "DqRKMGNqF5HjBQSVqDBN7FDFSA4OrcPBaGQL82RT" //Token dùng cho việc xác thực

#define WIFI\_SSID "VIETTEL\_GPON" //Tên wifi dùng để kết nối

#define WIFI\_PASSWORD "hoithangduy" //Mật khẩu wifi

#define SAVE\_PURSE\_KEY "PURSESENSOR" //Key lưu vĩnh viễn giá trị BPM trên Firebase

#define SET\_PURSE\_KEY "BPM" //Key lưu tạm thời giá trị BPM trên Firebase

#define THRESHOLD\_KEY "ThresHold" //Key dùng để thiết lập ngưỡng xác định interbeat interval

#define ISON\_KEY "isOn" //Key bật/tắt việc có lưu vĩnh viễn giá trị BPM hay không

NTPtime NTPch("ch.pool.ntp.org");//Server NTP

strDateTime dateTime; //Struct dateTime dùng để lấy thời gian thực vẽ biểu đồ

int isOn; //Biến bật/tắt việc có lưu vĩnh viễn giá trị BPM hay không

void setup() {

Serial.begin(115200); //Bắt đầu Serial kết nối Arduino r3 mức 115200

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //Khởi chạy kết nối wifi

Serial.print("Connecting"); //In trạng thái đang kết nối

while (WiFi.status() != WL\_CONNECTED) {//Khi chưa được kết nối in '.' và delay 500

Serial.print(".");

delay(500);

}

Serial.println();

Serial.print("connected: "); //In trạng thái đã kết nối

Serial.println(WiFi.localIP()); //In Ip local được kết nối

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH); //Khởi chạy kết nối tới Firebase dựa

//vào đường dẫn host và token xác thực

isOn = 0; //khởi tạo isOn đang tắt

}

void loop() {

String data = receiveUNO(); //Hàm nhận dữ liệu sensor được gửi từ Arduino R3

if (data!="") { //Kiểm tra dữ liệu có rỗng hay không

dateTime = NTPch.getNTPtime(7.0, 0);//Lấy thời gian thực: tham số đầu là Timezone

//(Múi giờ VN là 7)

//Tham số sau là DayLightSaving: 1 với thời //gian, 2 là thời gian US (VN không sử dụng)

isOn = receiveFirebase(ISON\_KEY); //Nhận dữ liệu isOn từ Firebase

if (isOn==1) { //Nếu isOn là đang bật thì lưu BPM lên Firebase

saveFirebase(SAVE\_PURSE\_KEY, data, dateTime); //Hàm gửi dữ liệu tới Firebase gồm

//data và thời gian thực

}

setFirebase(SET\_PURSE\_KEY, data); //Lưu tạm thời giá trị BPM trên Firebase

}

sendUNO((String)receiveFirebase(THRESHOLD\_KEY));//Nhận dữ liệu từ Firebase với //key là "ThresHold" và gửi sang cho Arduino R3 để set ngưỡng xác định //nhịp tim

}

void saveFirebase(String key, String data, strDateTime dateTime) { //Hàm gửi dữ liệu //cho Firebase

if (!dateTime.valid) return; //Kiểm tra thời gian thực đúng trước khi tiến hành gửi

String st = (String)dateTime.day + "\_" + (String)dateTime.month + "\_" + (String)dateTime.year + " " + (String)dateTime.hour + ":" + (String)dateTime.minute + ":" + (String)dateTime.second; //Tạo chuỗi thời //gian thực để gửi

Firebase.setInt(key + "/" + st, data.toInt()); //Gửi lên Firebase với tham số đầu là

//đường dẫn để lưu, hàm này ghi đè dữ liệu nếu trùng dữ liệu trên Firebase

}

void setFirebase(String key, String data) {//Hàm lưu tạm thời giá trị BPM trên Firebase

Firebase.setInt(key, data.toInt()); //Ghi đè giá trị BPM lên key "BPM" trên Firebase

}

int receiveFirebase(String key) {//Nhận dữ liệu từ Firebase dùng keyword để xác định dữ //liệu cần lấy

return Firebase.getInt(key); //Nhận về kiểu Int và return ra

}

void sendUNO(String data) { //Hàm gửi dữ liệu sang Arduino R3

Serial.println(data); //Gửi và delay 1s

delay(1000);

}

String receiveUNO() { //Hàm nhận dữ liệu từ Arduino R3

String buff = ""; //Tạo biến chuỗi đệm để nhận từng kí tự thêm vào

if (Serial.available()) { //Kiểm tra cổng kết nối sẵn sàng

while(Serial.available()) { //Lặp đến khi cổng kết nối hết nhận được dữ liệu

char data = (char)Serial.read();//Đọc kí tự từ cổng kết nối và ép sang kiểu char

buff += data; //Lưu vào biến String đệm

}

}

return buff; //Trả kết quả

}

* **Web platform:**
  + Source code web sử dụng Angular: https://github.com/nguyenngocbaothy/Firebase-angular-PULSESENSOR
  + Hướng dẫn chạy Angular:
    - 1. Cài nodejs.
    - 2. Gõ lệnh cài angular CLI: npm install -g @angular/cli
    - 3. Vào thư mục chứa các file đã tãi từ github gõ: npm install, sẽ tạo ra file node\_modules.
    - 4. Gõ “ng serve” và đợi server build xong.
    - 5. Lên trình duyệt web gõ: localhost:4200 là chạy.

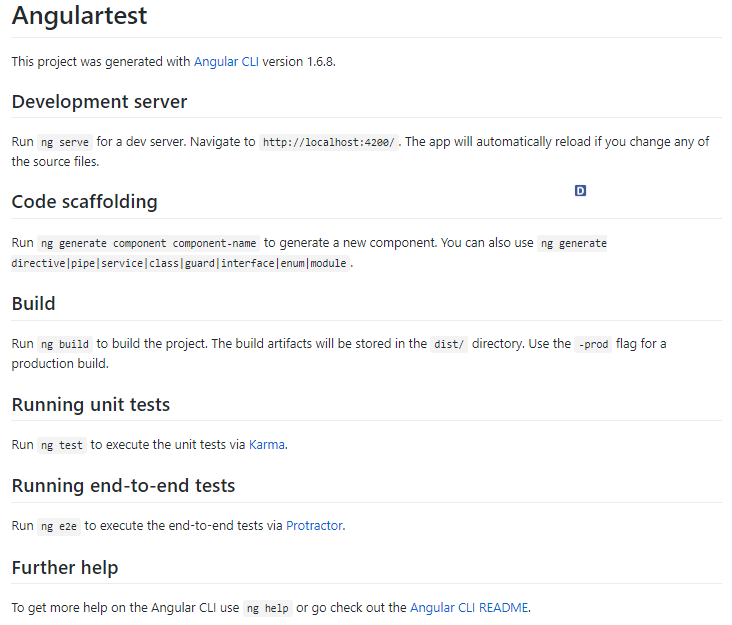
****

Figure 17: Chi tiết trên github

## 3.3. Mạch thi công thực tế

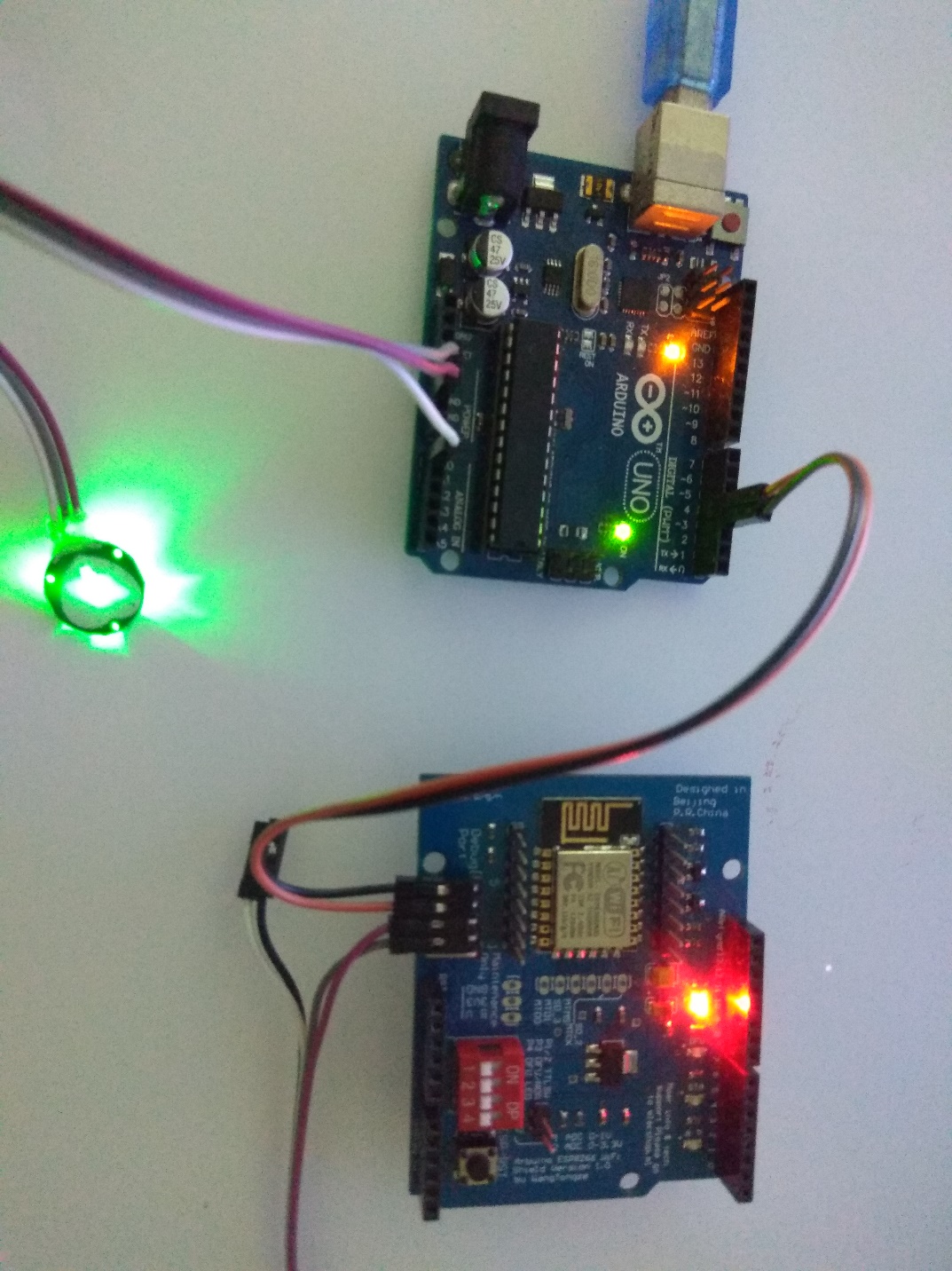
****

Figure 18: Hệ thống đo nhịp tim lắp ráp tách riêng

**Chú thích:** Cách lắp ráp tách riêng này dùng để xem các dữ liệu được xuất ra máy tính. Do khi tách riêng, Arduino sẽ có thể giao tiếp với máy tính thông qua cổng Serial, nếu không sẽ không thể hiển thị được dữ liệu.

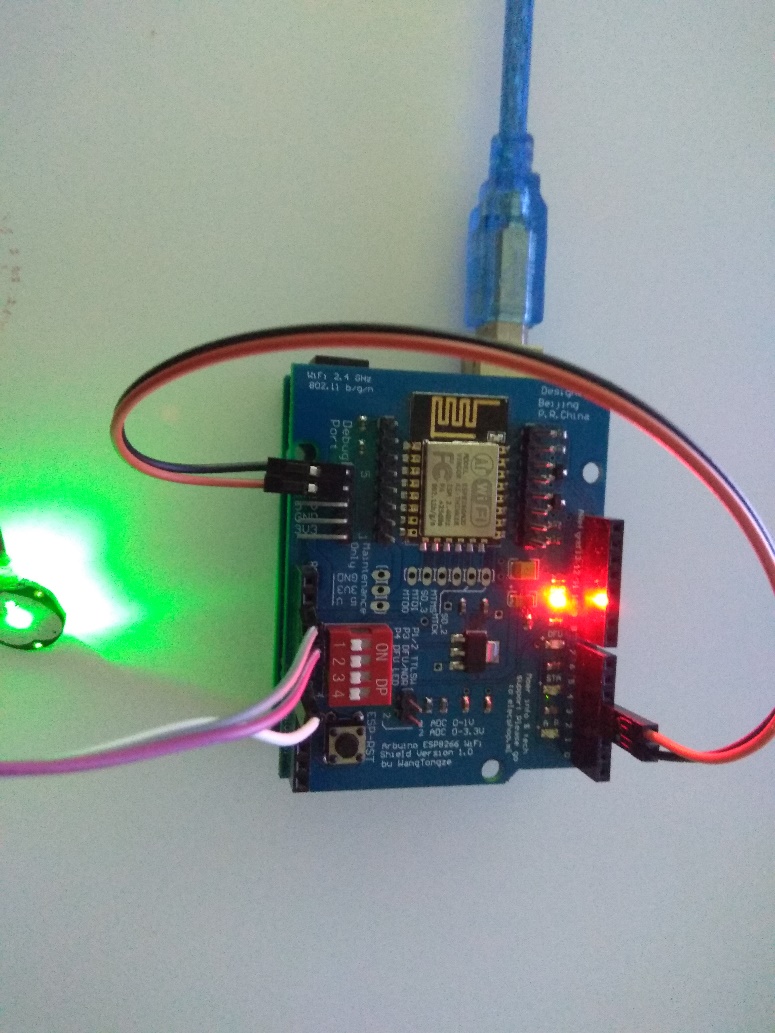


Figure 19: Hệ thống đo nhịp tim lắp ráp kết hợp (nhìn từ trên xuống)

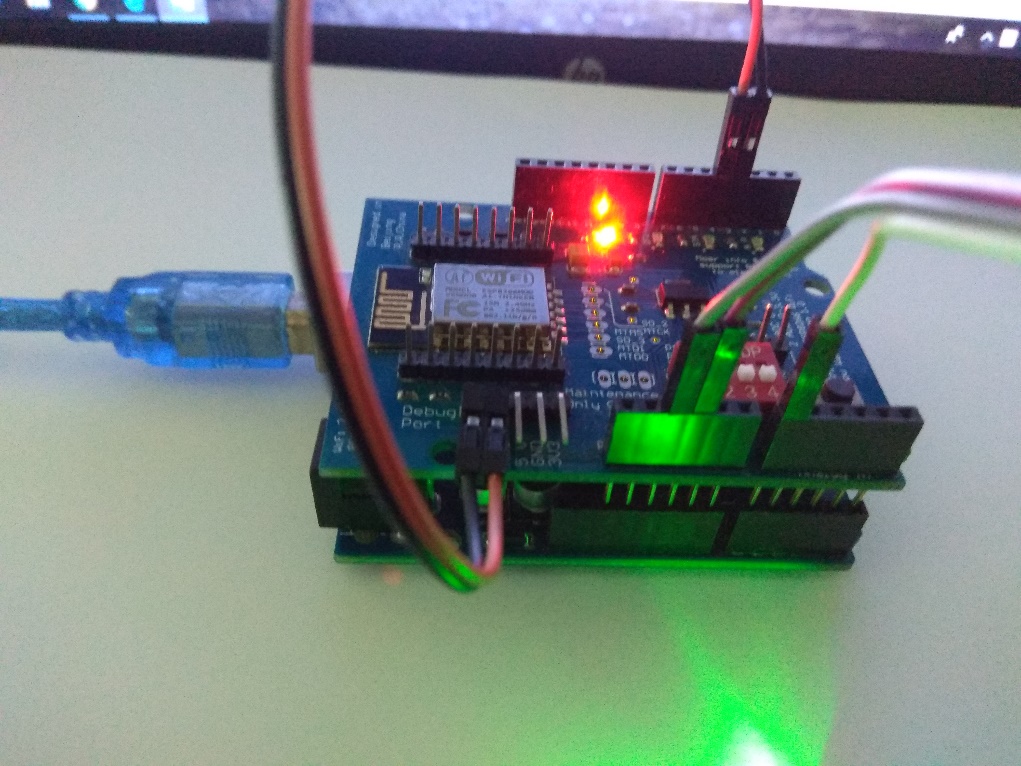
****

Figure 20: Hệ thống đo nhịp tim lắp ráp kết hợp (nhìn từ phía trước)

# KẾT LUẬN

Sau một thời gian tìm hiểu, dưới sự hướng dẫn nhiệt tình của thầy TS.GVC Nguyễn Xuân Sâm, nhóm chúng em đã hoàn thành đề tài “Xây dựng hệ thống đo nhịp tim“ hiển thị trên Web và Mobile.

Hệ thống đo nhịp tim và lưu trữ dữ liệu lên Firebase theo thời gian thực, có thể

tùy chỉnh được Threshold qua smartphone.

Qua đó chúng em được củng cố kiến thức đã học, kết hợp được với thực tiễn, phục vụ cho công việc tương lai rất nhiều.

Chúng em đã cố gắng hết sức để thực hiện đề tài một cách hoàn thiện và nhanh nhất có thể, nhưng do trình độ còn hạn chế nên còn nhiều thiếu sót, đề tài mới dừng lại ở việc làm một hệ thống đo nhịp tim đơn giản.

Cũng với đề tài này, trong tương lai chúng em sẽ phát triển nó ở mức cao hơn, có thể kết hợp chúng với các thiết bị khác để sử dụng trong thực tiễn cuộc sống như tích hợp vào đồng hồ đeo tay, dây chuyền để đo khi đang vận động hoặc tập thể dục…

## Các trang web tham khảo:

* http://arduino.vn/reference/howto
* https://hocarm.org/esp8266-va-firebase/
* https://www.arduino.cc/reference/en/language/functions/communication/serial
* https://pulsesensor.com/pages/code-and-guide
* https://github.com/firebase/firebase-arduino
* Source code web sử dụng Angular: https://github.com/nguyenngocbaothy/Firebase-angular-PULSESENSOR