

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH  
KHOA KỸ THUẬT ĐIỆN TỬ 2

---

# **ĐỒ ÁN**

# **TỐT NGHIỆP ĐẠI HỌC**

**CHUYÊN NGÀNH: KỸ THUẬT ĐIỆN - ĐIỆN TỬ**  
**HỆ ĐẠI HỌC CHÍNH QUY**  
**NIÊN KHÓA: 2014-2019**

*Đề tài:*

## **MẠCH HẸN GIỜ PHÁT THÔNG TIN THEO THỜI GIAN THỰC**

**Mã số đề tài: 19- N14DCDT112**

**Sinh viên thực hiện : HUỖNH ĐỨC LINH**  
**MSSV : N14DCDT112**  
**Lớp : D14CQKD01-N**  
**Giáo viên hướng dẫn : ThS. NGUYỄN LAN ANH**

**11/2018**  
**TP.HCM – 2018**

## MỤC LỤC

<b>LỜI NÓI ĐẦU.....</b>	<b>1</b>
<b>CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI.....</b>	<b>2</b>
1.1 Tổng Quan Về Lĩnh Vực Nghiên Cứu:.....	2
1.2 Mục Tiêu Đề Tài.....	3
1.3 Nhiệm Vụ Và Giới Hạn Của Đề Tài:.....	3
1.4 Phương pháp nghiên cứu: .....	3
<b>CHƯƠNG II: GIỚI THIỆU CÁC MODULE DS1307, DFplayer mini .....</b>	<b>4</b>
2.1 Giới thiệu tổng quát về vi điều khiển ESP8266 NODEMCU: .....	4
2.1.1 Giới thiệu tổng quát.....	4
2.1.2 Giới thiệu sơ bộ về ESP8266 NODEMCU .....	4
2.1.3 Module DS1307.....	14
2.1.3.1 Tổng quát Module RTC DS1307 .....	15
2.2.3.2 Cách lấy thông tin về thời gian và lịch .....	16
2.1.3.3 Chi tiết các thanh ghi .....	16
2.1.3.4 Lập trình Module Ds1307 .....	18
2.1.4 Module DFplayer mini .....	18
2.1.4.1 Giới thiệu về DFplayer.....	18
2.1.4.2 Tính năng cơ bản gồm có:.....	19
2.1.4.3 Mạch có thể được sử dụng trong nhiều ứng dụng:.....	19
2.1.4.4 Chế độ hoạt động .....	20
2.1.4.5 Lập trình cho DFplayer mini.....	22
2.1.5 Màn hình TFT 3.2 và TFT shield LCD module .....	22
2.1.5.1 Màn hình TFT 3.2 có chức năng là gì? .....	22
2.1.5.2 Thông số kỹ thuật.....	23
2.1.5.3 Phương thức giao tiếp. ....	24
2.1.6 Arduino mega 2560 .....	25
2.1.7 Loa 3 w .....	27
2.2. Giao diện người dùng trên màn hình 3.2 TFT .....	27
2.2.1 Các thanh công cụ điều khiển.....	27

2.2.2 Chọn bài hát và phát thông tin theo yêu cầu .....	28
2.3. Giới thiệu tổng quát về Blynk.....	30
2.3.1 App Blynk là gì?.....	30
2.3.2 Tại sao phải dùng App Blynk.....	30
2.4. Hướng dẫn sử dụng App Blynk .....	31
2.4.1 Tải ứng dụng Blynk trên ANDROID hoặc IOS .....	31
2.4.2 Lấy mã AUTH TOKEN .....	31
2.4.3 Cài đặt thư viện Blynk.....	32
2.4.4 Tạo code ví dụ .....	32
2.4.5 Dán mã xác thực .....	34
2.5. Xây dựng app điều khiển bằng qua Blynk.....	35
2.5.1 Gồm những chức năng.....	35
2.5.2 Hẹn giờ phát thông tin .....	36
2.6 Khái niệm cơ bản về giao thức Modbus.....	37
2.6.1 Các chuẩn modbus nào đang được sử dụng phổ biến .....	38
2.6.2 Khái niệm cơ bản về giao thức Modbus RTU .....	39
2.6.3 Cấu trúc bản tin Modbus RTU .....	40
<b>CHƯƠNG III: THIẾT KẾ MÔ HÌNH PHẦN CỨNG VÀ PHẦN MỀM LẬP TRÌNH ỨNG DỤNG .....</b>	<b>47</b>
3.1 Thiết kế mô hình phần cứng và phần mềm lập trình cho hệ thống.....	47
3.1.1 Tìm hiểu phần mềm lập trình Arduino IDE: .....	47
3.1.2 Sơ đồ kết nối.....	49
3.1.3 Sơ đồ giải thuật:.....	50
<b>CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>52</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>53</b>

## MỤC LỤC HÌNH

Hình 2.1: ESP8266 NODEMCU .....	4
Hình 2.2: Sơ đồ ra chân ESP8266 NODEMCU .....	5
Hình 2.3: Tải firmware cho module ESP8266 NODEMCU .....	6
Hình 2.4: Dán đường dẫn file .json vào Arduino IDE .....	7
Hình 2.5: Cài Firmware ESP8266 NODEMCU cho Arduino.....	8
Hình 2.6: Cài đặt gói thư viện cho ESP8266 NODEMCU .....	9
Hình 2.7: Chọn NodeMCU Board trong Aduino IDE .....	9
Hình 2.8: Thiết lập cổng COM port kết nối máy tính với board.....	10
Hình 2.9: Kiểm tra kết nối với chương trình Blink có sẵn.....	11
Hính 2.10: Compile chương trình nạp xuống Kit.....	12
Hình 2.11: Lỗi kết nối với máy tính thường gặp.....	13
Hình 2.23: Update driver để sửa lỗi .....	14
Hình 2.13: Module RTC DS1307.....	14
Hình 2.14: Chip DS1307 .....	15
Hình 2.15: Thanh ghi trong chip DS1307 .....	16
Hình 2.16: Module DFPlayer Mini .....	18
Hình 2.17: Serial mode .....	20
Hình 2.18: Ad key mode .....	21
Hình 2.19: I/O mode.....	21
Hình 2.20: Màn hình TFT 3.2 .....	22
Hình 2.21: Bảng sự thật TFT 3.2.....	24
Hình 2.23: App Blynk .....	30
Hình 2.24 Auth token .....	31
Hình 2.25 Giao diện của App Blynk .....	32
Hình 2.26. Thanh công cụ của Blynk .....	33
Hình 2.27 Thiết lập công cụ trên Blynk .....	33
Hình 2.28. Nạp vào ESP8266.....	34
Hình 2.29: Sơ đồ kết nối modbus .....	38
Hình 2.30: Mô hình OSI.....	40
Hình 2.31: Khung truyền dữ liệu.....	40
Hình 3.1: Phần mềm lập trình Arduino IDE .....	47
Hình 3.2: Sơ đồ kết nối hệ thống.....	49

### LỜI NÓI ĐẦU

Ngày nay việc ứng dụng vi điều khiển, vi xử lý đang ngày càng phát triển rộng rãi và thâm nhập ngày càng nhiều vào các lĩnh vực kỹ thuật và đời sống xã hội. Tuy nhiên ứng dụng cho các hệ thống nhúng ngày nay không đơn giản chỉ dừng lại ở điều khiển đèn nhấp nháy, đếm số người vào/ra, hiển thị dòng thông báo trên matrix led hay điều khiển ON-OFF của động cơ... mà nó ngày càng trở nên phức tạp. Và với xu hướng tất yếu này cùng với sự phát triển mạnh mẽ của công nghệ chế tạo vi mạch, người ta đã tạo ra những vi điều khiển có cấu trúc mạnh hơn, đáp ứng thời gian thực tốt hơn, chuẩn hóa hơn so với các vi điều khiển 8 bit trước đây.

Với sự phát triển mạnh mẽ của khoa học, đặc biệt là ngành điện tử, sự phát minh ra các linh kiện điện tử đã và đang ngày càng đáp ứng được yêu cầu của các hệ thống. Ưu điểm của việc sử dụng các linh kiện điện tử làm cho các hệ thống linh hoạt và đa dạng hơn, giá thành thấp hơn và độ chính xác cao hơn.

Sau gần 4 năm học tập và nghiên cứu ở trường, em đã được làm quen với các môn học chuyên ngành. Để áp dụng lý thuyết với thực tế em được giao đề án môn cuối khóa với tên đề tài là “**Mạch hẹn giờ phát thông tin theo thời gian thực**”.

Tuy nhiên do kiến thức chuyên môn còn hạn chế, tài liệu tham khảo có giới hạn nên còn xảy ra nhiều sai sót. Em rất mong thầy và các bạn góp ý bổ sung để bản đồ án của em được hoàn thiện hơn và giúp em hiểu biết hơn trong quá trình học tập tiếp theo.

**Em xin chân thành cảm ơn!**

Huỳnh Đức Linh

### CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

#### 1.1 Tổng Quan Về Lĩnh Vực Nghiên Cứu:

Trong những năm gần đây, các hệ thống điều khiển theo thời gian thực là một trong những lĩnh vực thu hút nhiều sự chú ý trong giới khoa học nghiên cứu về khoa học máy tính. Trong đó, vấn đề điều hành thời gian thực và vấn đề lập lịch là đặc biệt quan trọng. Một trong ứng dụng quan trọng của hệ thống thời gian thực (RTS) đã và đang được ứng dụng rộng rãi hiện nay là các dây chuyền sản xuất tự động, robot, điều khiển các thí nghiệm tự động, trong thiết kế đồng hồ hiển thị thời gian thực...Thế hệ ứng dụng tiếp theo của hệ thống này sẽ điều khiển robot giống con người, hệ thống kiểm soát thông minh trong các nhà máy công nghiệp, điều khiển các trạm không gian.

Và cho ra đời các “Mạch hẹn giờ phát thông tin theo thời gian” được phát minh ngày càng nhiều để hỗ trợ cuộc sống của con người. Đã bao giờ bạn cảm thấy khó chịu khi phải tự bật thông tin vào mỗi buổi sáng hay cài chuông báo thức chưa, thật phức tạp và tốn thời gian, và mạch hẹn giờ phát thông tin phục vụ có nhiều ứng dụng mang lại tiện ích và tiết kiệm tiền bạc cho gia đình bạn. Những sản phẩm này giúp bạn có thể chủ động hơn trong những công việc cần lập trình về giờ giấc, ngoài ra còn giúp bạn tiết kiệm điện năng, dễ dàng cài đặt để thông báo cho các công xưởng vào thời gian chính xác, ...

Cụ thể hơn, ứng dụng này sẽ thông báo những sự kiện quan trọng cho người dùng, nhắc nhở làm một công việc nào đó như đến giờ uống thuốc, tập thể dục, sinh hoạt cá nhân, cập nhật tin tức mới từ internet hoặc gặp gỡ khách hàng v.v...

### **1.2: Mục Tiêu Đề Tài**

- Tìm hiểu nguyên lý mạch giải mã phát thông tin theo thời gian thực và các mô hình điều khiển bằng wifi.
- Tìm hiểu các thiết bị linh kiện sử dụng trong hệ thống như: Module wifi ESP8266, vi điều khiển, các module DFplayer, DS1307...
- Tìm hiểu các phần mềm thiết kế và thực hiện hệ thống: Phần mềm tạo APP, phần mềm lập trình vi điều khiển ...
- Thực hiện thiết kế mạch hẹn giờ phát thông tin.

### **1.3. Nhiệm Vụ Và Giới Hạn Của Đề Tài:**

Với sự hạn chế về kinh nghiệm, kinh phí và thời gian thực hiện nên em xin giới hạn phạm vi thực hiện đề tài là:

- Tìm hiểu nguyên lý mạch giải mã dfplayer và các mô hình điều khiển bằng wifi.
- Tìm hiểu các module cảm biến hiện có trên thị trường hiện nay.
- Tìm hiểu các phần mềm thiết kế và thực hiện hệ thống: Phần mềm tạo APP, phần mềm lập trình vi điều khiển...
- Thực hiện thiết kế mạch hẹn giờ phát thông tin.

### **1.4. Phương pháp nghiên cứu:**

Nghiên cứu dựa vào các nguồn tài liệu tìm kiếm được từ các nghiên cứu và các bài báo trong nước và cả nước ngoài. Nghiên cứu của các sinh viên trong và ngoài nước. Xây dựng mạch hẹn giờ phát thông tin từ các thiết bị có sẵn như vi điều khiển, các module cảm biến.

## **CHƯƠNG II: GIỚI THIỆU CÁC MODULE DS1307, DFplayer mini**

### **2.1 Giới thiệu tổng quát về vi điều khiển ESP8266 NODEMCU:**

#### **2.1.1 Giới thiệu tổng quát**

Module ESP-12 [1, overview] kết hợp với firmware ESP8266 trên Arduino và thiết kế phần cứng giao tiếp tiêu chuẩn đã tạo nên NodeMCU, loại Kit phát triển ESP8266 phổ biến nhất trong thời điểm hiện tại. Với cách sử dụng, kết nối dễ dàng, có thể lập trình, nạp chương trình trực tiếp trên phần mềm Arduino, đồng thời tương thích với các bộ thư viện Arduino sẵn có, NodeMCU là sự lựa chọn hàng đầu cho các bạn muốn tìm hiểu về ESP8266 hiện nay.



***Hình 2.1: ESP8266 NODEMCU***

#### **2.1.2 Giới thiệu sơ bộ về ESP8266 NODEMCU**

Khả năng hoạt động như một modem wifi:

- Có thể quét và kết nối đến một mạng wifi bất kỳ (Wifi Client) để thực hiện các tác vụ như lưu trữ, truy cập dữ liệu từ server.
- Tạo điểm truy cập wifi (Wifi Access Point) cho phép các thiết bị khác kết nối, giao tiếp và điều khiển.
- Là một server để xử lý dữ liệu từ các thiết bị sử dụng internet khác.

Nguồn vào và nguồn ra

ESP8266 NodeMCU nhận nguồn từ cổng micro USB tích hợp sẵn trên mạch, giúp việc nạp code trở nên dễ dàng hơn. Bên cạnh đó, việc cấp nguồn cho module cũng linh động hơn vì bạn có thể sử dụng sạc dự phòng thay cho nguồn từ USB trên máy tính (nguồn cấp tối đa là 5V).



## CHƯƠNG II: GIỚI THIỆU CÁC MODULE DS1307, DFplayer mini

ESP8266 NodeMCU có thể cung cấp nguồn cho tối đa 4 thiết bị: 3 nguồn ra 3.3V và một nguồn từ chân Vin (điện thế bằng điện thế từ cổng micro USB). Khi sử dụng các chân cấp nguồn này, hãy luôn kiểm tra để chắc chắn không cắm nhầm chân dương (trên mạch in là 3v3 và Vin) và chân âm (GND). Tuy nhiên, 3 chân 3.3V đều được bảo vệ, khi cắm ngược cực, module sẽ chỉ nóng lên và dừng hoạt động. Chân Vin thì KHÔNG, cắm ngược cực ở chân này là module bốc khói theo nghĩa đen nhé (kinh nghiệm xương máu).

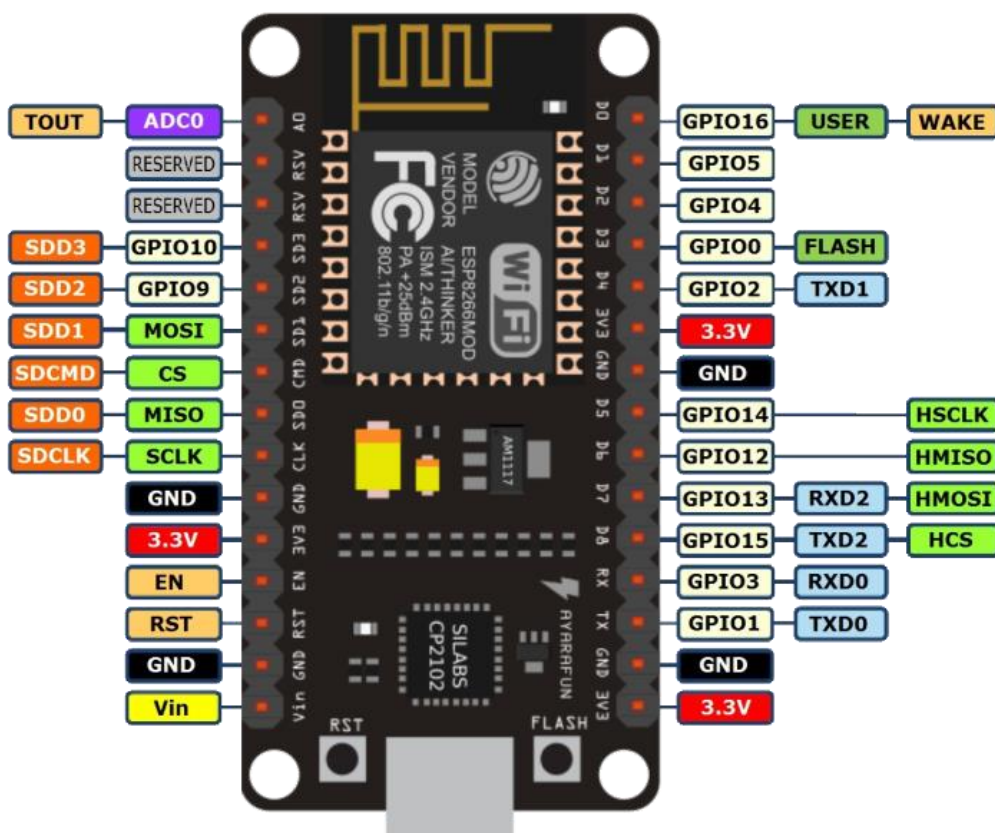
Truyền và nhận tín hiệu

ESP8266 NodeMCU [2] có tổng cộng 13 chân GPIO (General-purpose input/output) – chân có thể truyền/nhận tín hiệu (trên mạch in từ D0 đến D8 và RX, TX, SD2, SD3).

### Bắt đầu sử dụng

Nắm được sơ lược lý thuyết rồi, chúng ta hãy bắt đầu thực hành luôn nhé. Phần này sẽ hướng dẫn bạn cài firmware cho ESP8266 NodeMCU và cài đặt chương trình đầu tiên.

*Bước 1: Bản đồ chân Pin trên ESP8266 NodeMCU*



*Hình 2.2: Sơ đồ ra chân ESP8266 NODEMCU*

### LƯU Ý:

- Module chỉ có thể kết nối tới nguồn tối đa 5V qua cổng micro USB.
- Các chân I/O chỉ có thể giao tiếp với các linh kiện qua điện thế tối đa là 3.3V

### Bước 2: Tải firmware cho module

<https://github.com/esp8266/Arduino#installing-with-boards-manager>

- Continuing
- License and credits

#### Installing with Boards Manager

Starting with 1.6.4, Arduino allows installation of third-party platform packages using Boards Manager. We have packages available for Windows, Mac OS, and Linux (32 and 64 bit).

- Install Arduino 1.6.5 from the [Arduino website](#).
- Start Arduino and open Preferences window.
- Enter [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) into Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install esp8266 platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).

The best place to ask questions related to this core is ESP8266 community forum: <http://www.esp8266.com/arduino>. If you find this forum or the ESP8266 Boards Manager package useful, please consider supporting it with a donation. [paypal](#) [donate](#)

#### Available versions

Stable version updated Nov 30, 2015

Boards manager link: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

Documentation: <http://esp8266.github.io/Arduino/versions/2.0.0/>

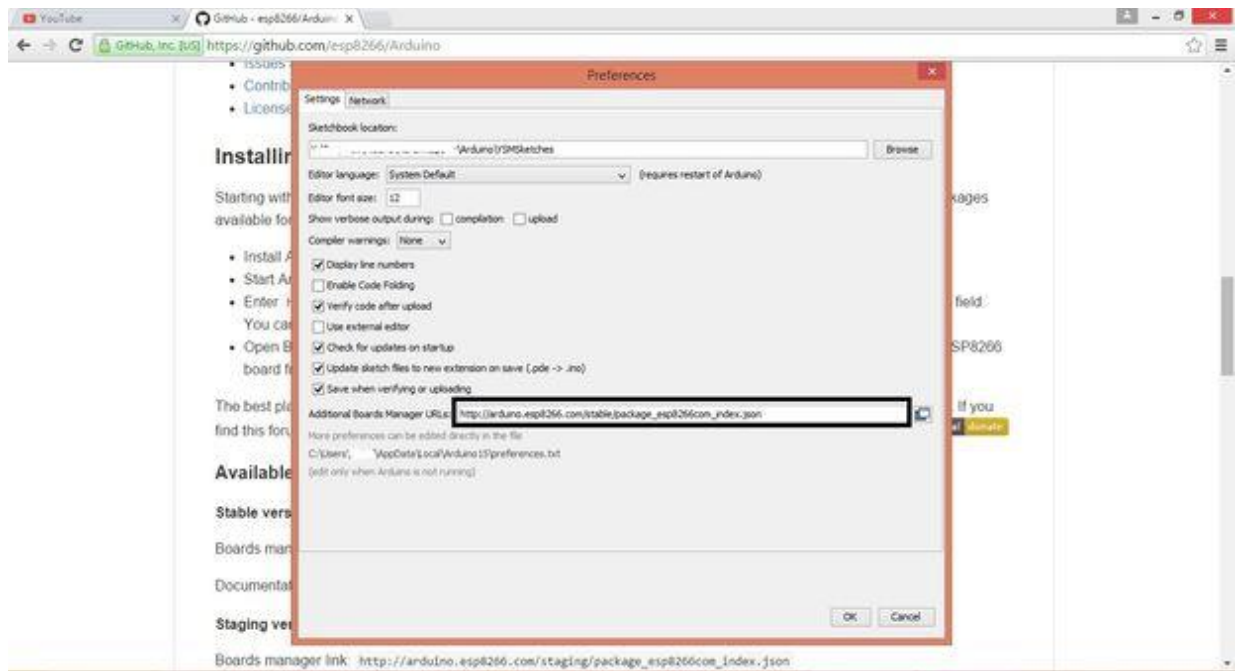
**Hình 2.3: Tải firmware cho module ESP8266 NODEMCU**

Như nội dung trong hình, bạn hãy copy đường dẫn của firmware mới nhất từ trang Github này.

Đường dẫn nhìn như thế này:

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

### Bước 3: Dán đường dẫn của file .json đó vào Arduino IDE



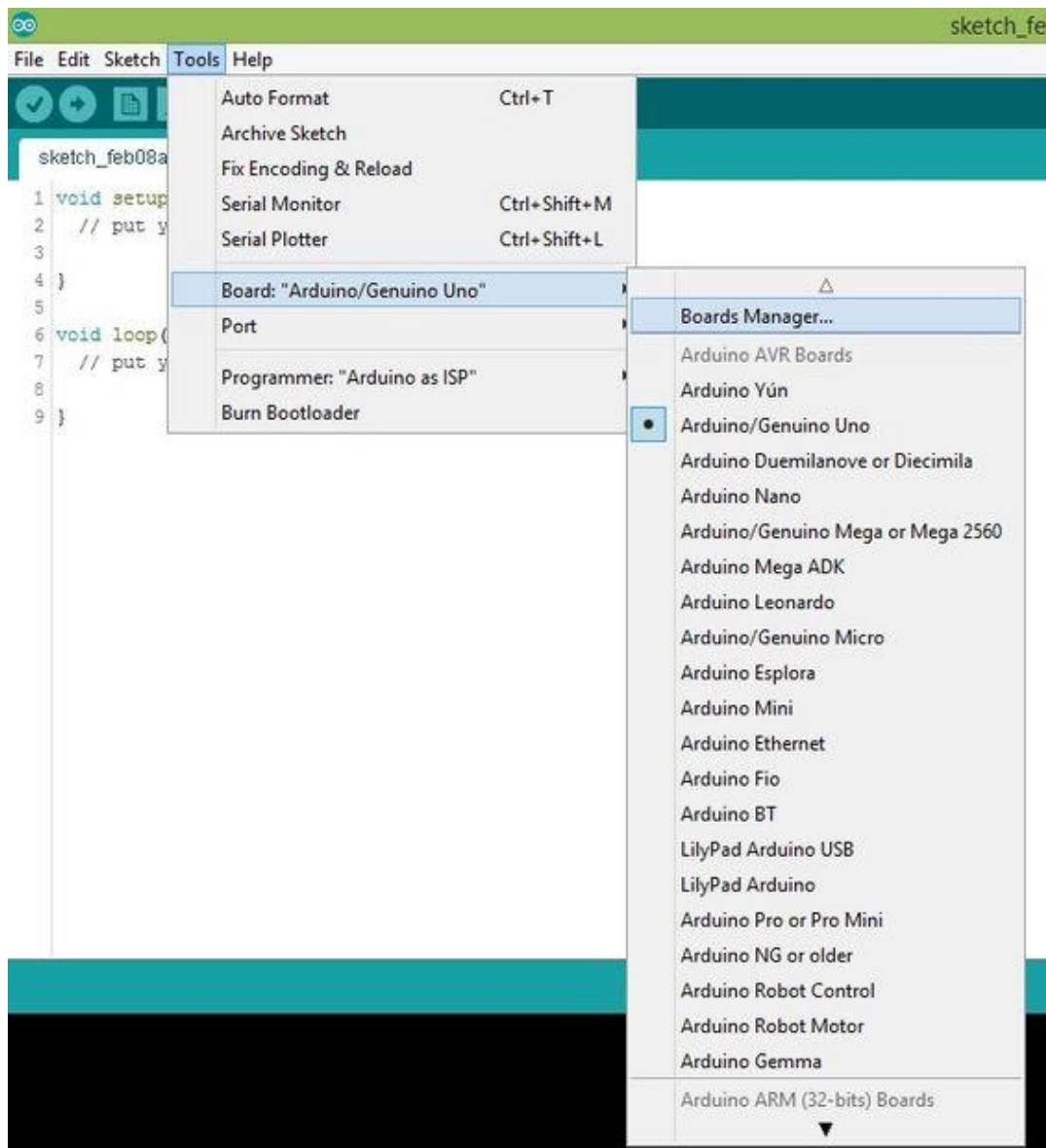
**Hình 2.4: Dán đường dẫn file .json vào Arduino IDE**

Vào File -> Preference -> dán đường dẫn vào Additional Boards Manager URLs -> OK -> khởi động lại IDE.

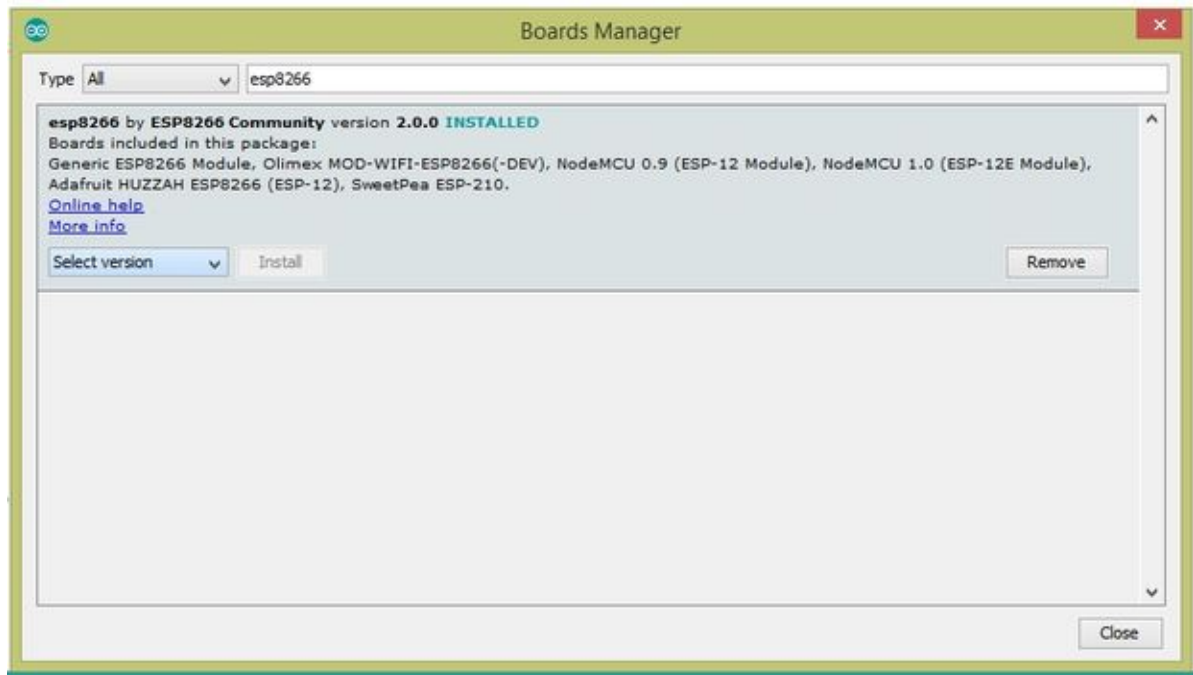
Trong trường hợp máy bạn chưa cài Arduino IDE (môi trường lập trình cho các mạch Arduino). Hãy tải ở link này:

- Tải trực tiếp từ Windows Store: <https://www.microsoft.com/store/apps/9nblggh4rsd8>
  - Tải file cài đặt: [https://www.arduino.cc/download\\_handler.php](https://www.arduino.cc/download_handler.php)
- Hoặc vào trang download của Arduino và tải phiên bản bạn mong muốn.

### Bước 4: Cài đặt Firmware ESP8266 cho Arduino



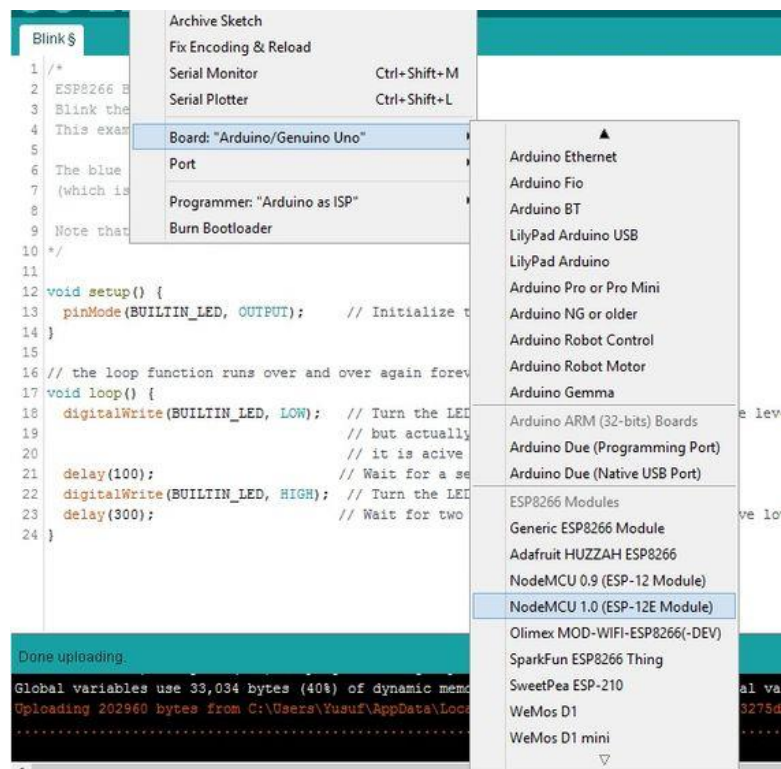
*Hình 2.5: Cài Firmware ESP8266 NODEMCU cho Arduino*



**Hình 2.6 Cài đặt gói thư viện cho ESP8266 NODEMCU**

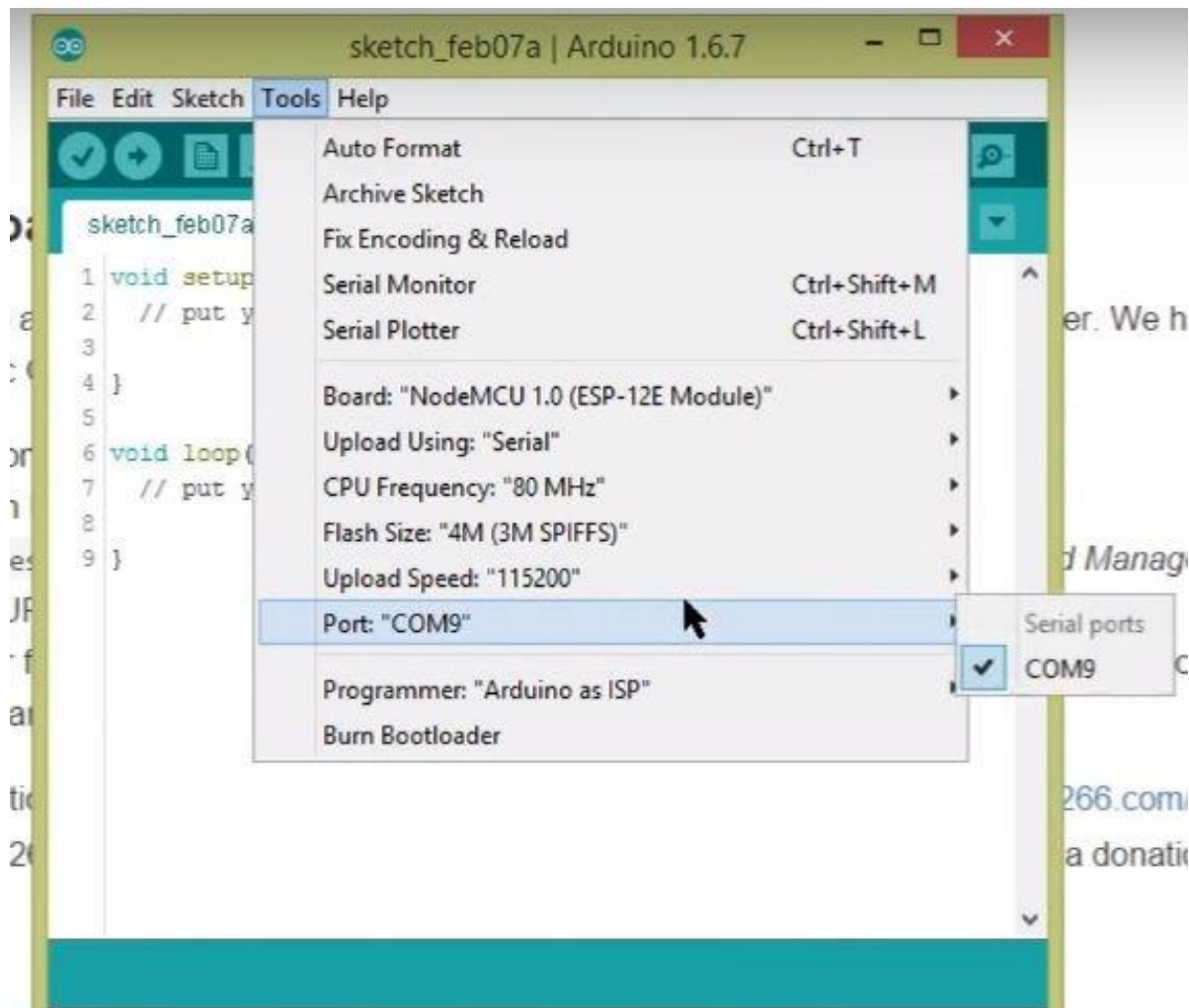
Vào Tools -> Boards Manager -> tìm tên “esp8266” -> Install -> Khởi động lại IDE

### Bước 5: Chọn NodeMCU Board trong Aduino IDE



**Hình 2.7: Chọn NodeMCU Board trong Aduino IDE**

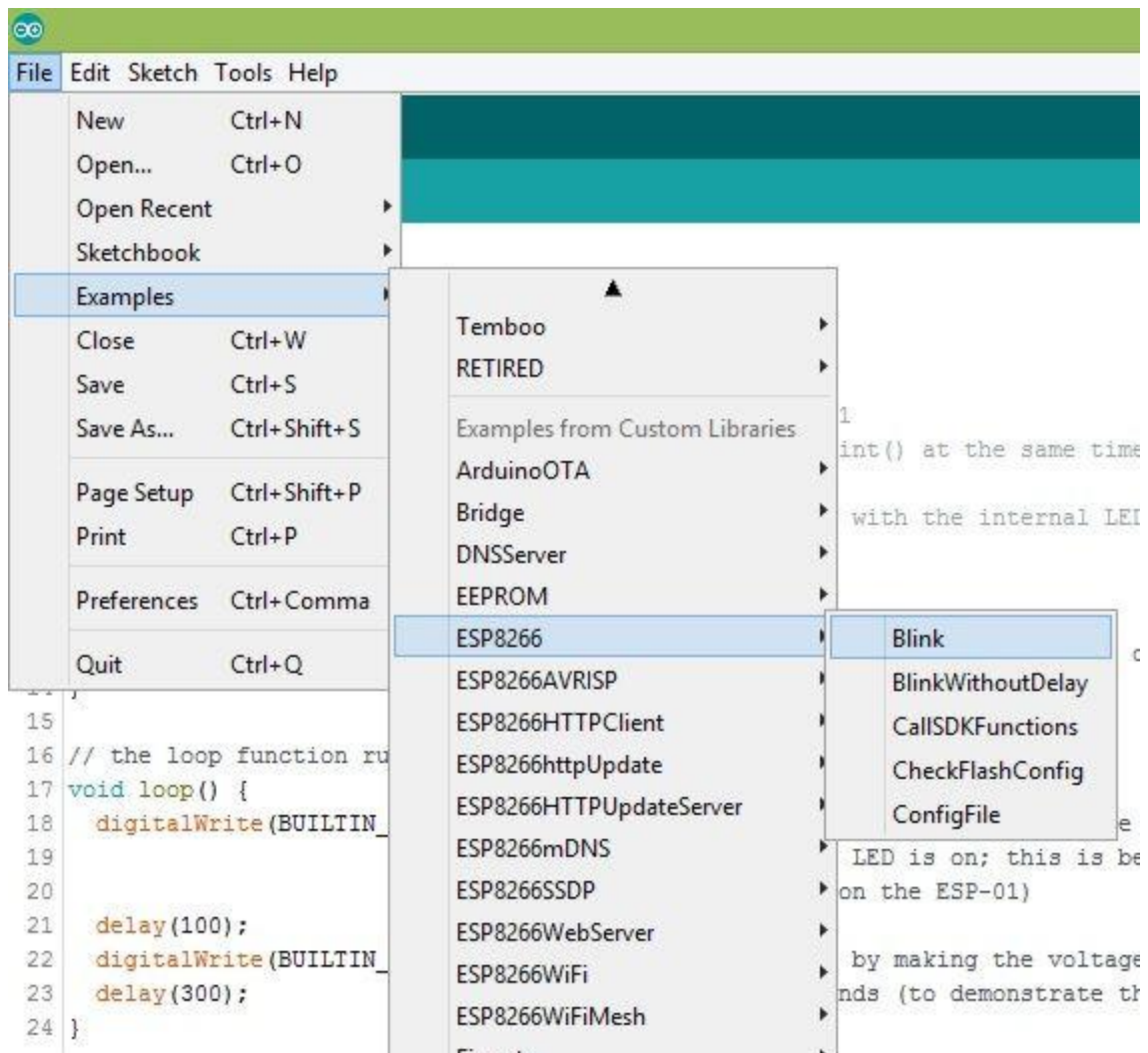




*Hình 2.8: Thiết lập cổng COM port kết nối máy tính với board*

Vào Tools -> Board -> kéo tìm và chọn **NodeMCU 1.0 (ESP-12E Module)** và thiết đặt COM Port kết nối máy tính của Board, vậy là bạn đã sẵn sàng để chạy chương trình đầu tiên.

### Bước 6: LED Blink – kết nối đèn Led có sẵn trên module



**Hình 2.9: Kiểm tra kết nối với chương trình Blink có sẵn**



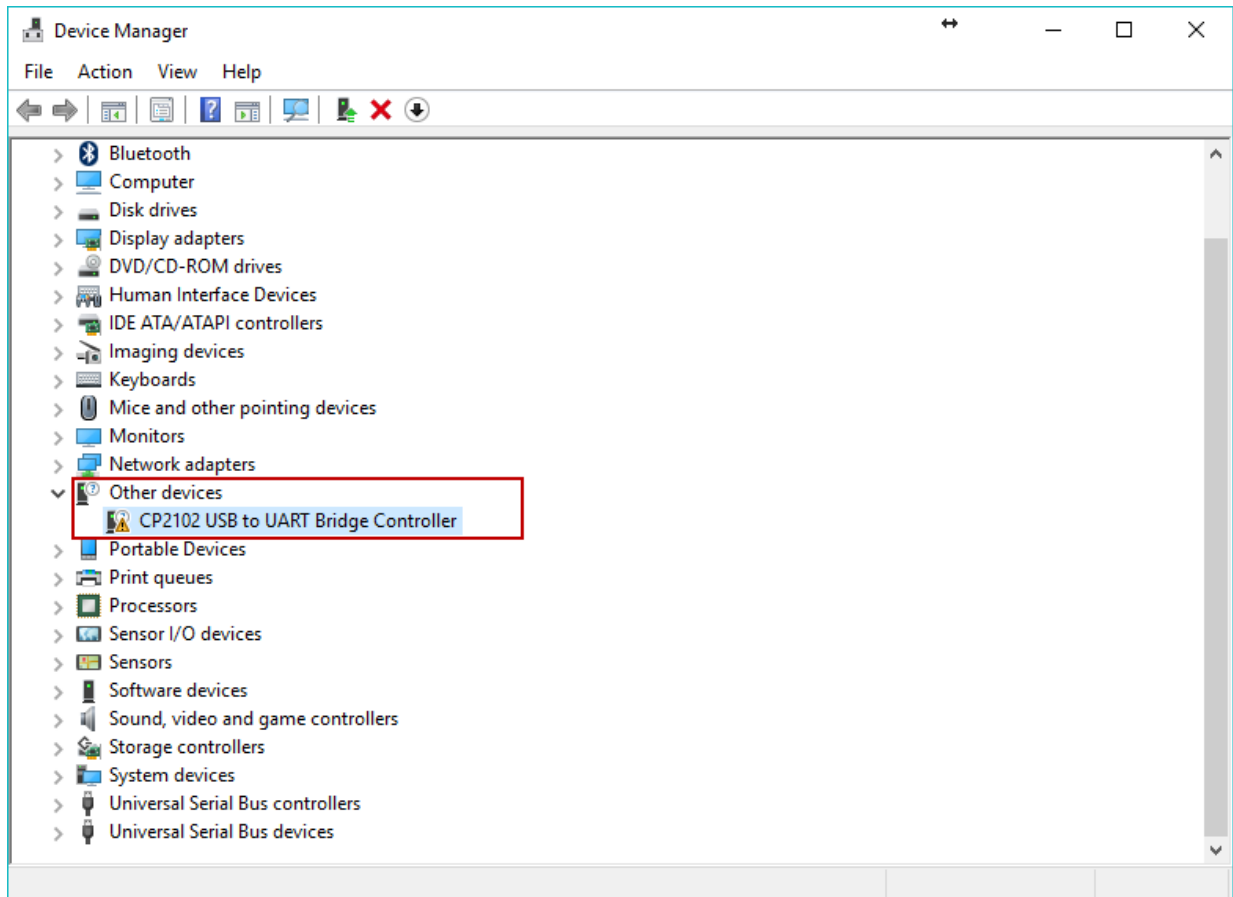
*Hình 2.10: Compile chương trình nạp xuống Kit*

Phần code sẽ như thế này, sau đó chọn nút Upload, chờ IDE nạp code vào module và xem thành quả.

### **Máy tính của bạn không kết nối được với ESP8266 NodeMCU?**

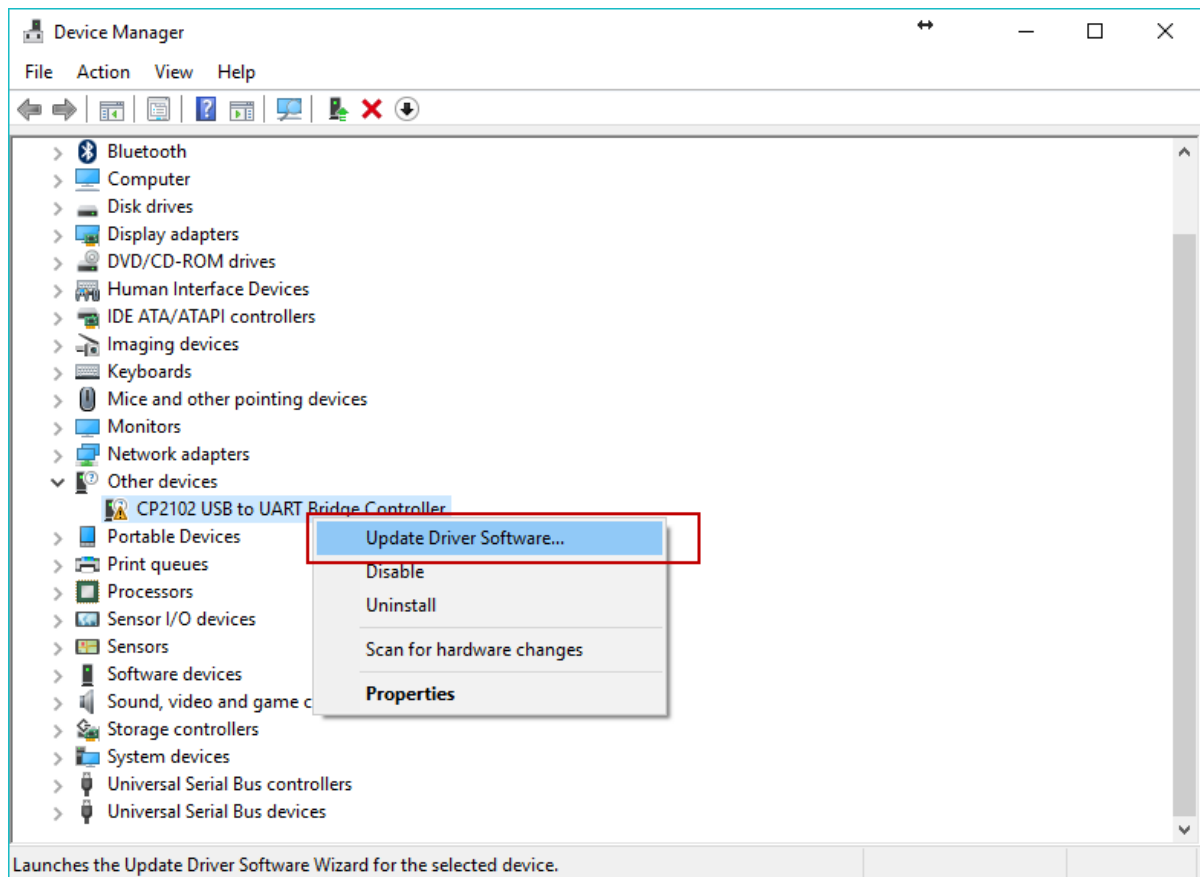
Trong trường hợp đã cắm module vào máy tính nhưng không thấy tín hiệu gì, hãy kiểm tra lại driver đã nhận hay chưa bằng cách vào Device Manager





*Hình 2.11: Lỗi kết nối với máy tính thường gặp*

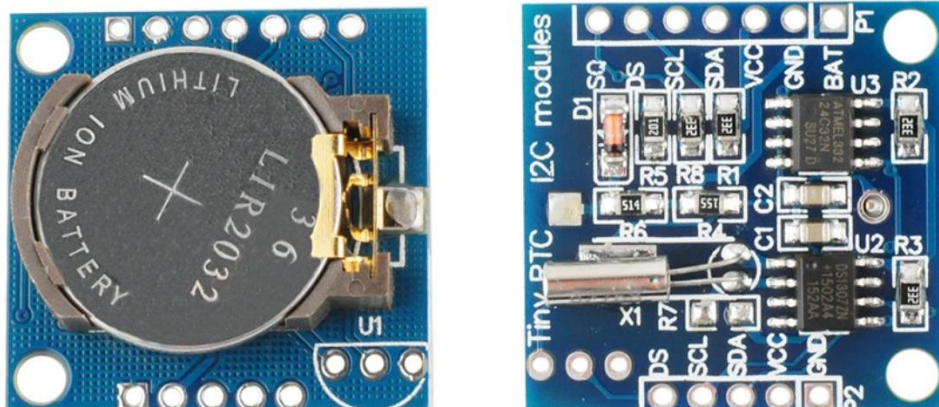
Nếu bạn thấy có dấu chấm than như hình nghĩa là chưa có driver. Để khắc phục, hãy tải driver rồi giải nén. Sau đó click phải chuột chọn Update Driver Software



*Hình 2.23: Update driver để sửa lỗi*

Rồi chọn Browse my computer for driver software và tìm đến thư mục chứa driver để cài đặt. Sau khi cài driver xong, kiểm tra lại trong Device Manager thấy không còn dấu chấm than là OK.

### 2.1.3 Module DS1307

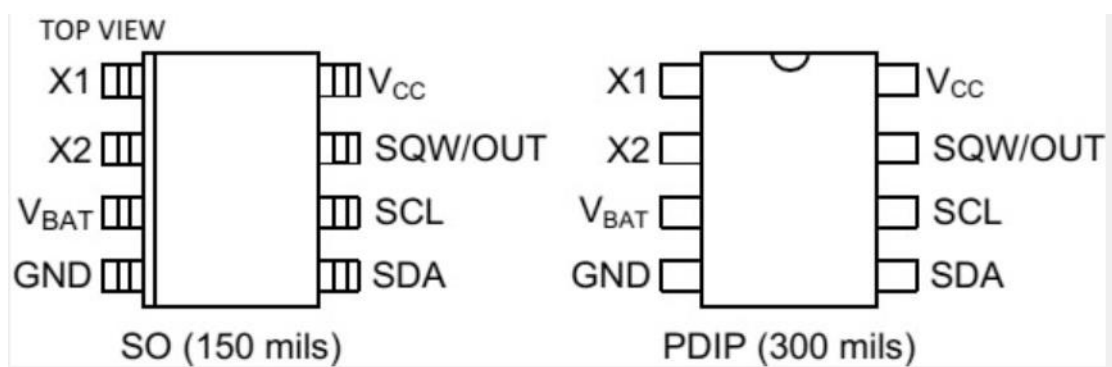


*Hình 2.13: Module RTC DS1307*

### 2.1.3.1 Tổng quát Module RTC DS1307

- DS1307 [3] là chip thời gian thực hay RTC (Read time clock). Đây là một IC tích hợp cho thời gian bởi tính chính xác về thời gian tuyệt đối cho thời gian: Thứ, ngày, tháng, năm, giờ, phút, giây. DS1307 là chế tạo bởi Dallas. Chip này có 7 thanh ghi 8 bit mỗi thanh ghi này chứa: Thứ, ngày, tháng, năm, giờ, phút, giây. Ngoài ra DS1307 còn chứa 1 thanh ghi điều khiển ngõ ra phụ và 56 thanh ghi trống các thanh ghi này có thể dùng như là RAM. DS1307 được đọc thông qua chuẩn truyền thông I2C nên do đó để đọc được và ghi từ DS1307 thông qua chuẩn truyền thông này. Do nó được giao tiếp chuẩn I2C nên cấu tạo bên ngoài nó rất đơn giản.

Sơ đồ chân:



Hình 2.14: Chip DS1307

Trong đó:

Chân	Tên	Chức năng
1	X1	Kết nối đến thạch anh 32.768Khz làm nguồn dao động cho chip
2	X2	
3	VBat	Kết nối đến cực dương của Pin dự phòng, có điện áp tiêu chuẩn khoảng 3V
4	GND	Kết nối đến mass
5	SDA	Chân dữ liệu khi kết nối đến bus I2C
6	SCL	Chân nhận xung clock đồng bộ khi kết nối bus I2C
7	SQW/OUT	Ngõ xuất xung vuông, tần số có thể lập trình để thay đổi từ 1Hz, 4Khz, 8 Khz, 32 Khz
8	VCC	Nguồn cấp chính, khoảng 5VDC

### 2.2.3.2 Cách lấy thông tin về thời gian và lịch

Bảng 1 là địa chỉ của các thanh ghi RAM và thanh ghi RTC của DS 1307. Thanh ghi RTC có địa chỉ từ 00-07h. Các thanh ghi RAM chiếm địa chỉ từ 08h - 3Fh.

Các thông tin về thời gian và lịch thu được bằng cách đọc các byte trong thanh ghi tương ứng. Thời gian và lịch được thiết lập hoặc khởi tạo bằng cách viết các byte thanh ghi thích hợp. Lưu ý dữ liệu lưu thời gian chứa trong thanh ghi đều theo định dạng BCD. Thứ trong tuần thay đổi tại lúc nửa đêm, lưu dưới dạng con số (Ví dụ 1 là Chủ Nhật, 2 là Thứ hai, 3 là thứ 3 ...). Khi lần đầu tiên IC được cấp nguồn các thanh ghi thời gian và lịch reset về 01/01/00 01 00:00:00. (MM/DD/YY DOW HH:MM:SS).

00h	GIÂY
01h	PHÚT
02h	GIỜ
03h	THỨ
04h	NGÀY
05h	THÁNG
06h	NĂM
07h	THANH GHI ĐIỀU KHIỂN
08h-3Fh	VÙNG RAM LƯU TRỮ 56 x 8

*Hình 2.15: Thanh ghi trong chip DS1307*

### 2.1.3.3 Chi tiết các thanh ghi

Đây là các Timekeeper registers của DS1307, chúng ta sẽ dựa vào bảng này để read/write IC DS1307 qua I2C.

## CHƯƠNG II: GIỚI THIỆU CÁC MODULE DS1307, DFplayer mini

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

\*0 luôn luôn đọc là 0

### Bảng 1: Các thanh ghi lưu giữ thời gian

#### Thanh ghi control

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

**Bit 7: Output Control (OUT)** thanh ghi điều khiển ngõ ra. Nó ảnh hưởng đến chân số 7 (chân SQW/OUT). Khi sóng vuông xuất ra chân này bị disable, nếu bit OUT=1 khi đó chân số 7 ở mức cao, ngược lại bit OUT=0 khi đó chân số 7 ở mức thấp.

**Bit 6:** Luôn luôn đọc bằng 0

**Bit 5:** Luôn luôn đọc bằng 0

**Bit 4: Square-Wave Enable (SQWE):** khi bit này được thiết lập bằng 1, cho phép xuất ra xung vuông tại chân số 7, tần số của của xung vuông phụ thuộc vào bit RS0 và RS1. Khi tần số sóng vuông được thiết lập là 1 Hz. Các thanh ghi thời gian được cập nhật tại cạnh xuống của xung vuông. Khi lần đầu khởi tạo cấp nguồn đến thiết bị, bit này bằng 0.

**Bits 1 and 0:** Lựa chọn tốc độ (RS [1:0]). Những bit này điều khiển tần số của tần số sóng vuông, khi sóng vuông được cho phép.

RS1	RS0	SÓNG VUÔNG NGÕ RA SQW/OUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

**Thanh ghi giây (địa chỉ 00h):** 4 bit thấp chứa hàng đơn vị, 3 bit cao chứa hàng chục của giây. Ngoài ra bit thứ 7 có tên là CH, nếu bit này được thiết lập bằng 1 thì đồng hồ không hoạt động. Vì vậy phải thiết lập bit này bằng không ngay từ đầu.

**Thanh ghi phút (địa chỉ 01h):** 4 bit thấp chứa hàng đơn vị, 3 bit cao chứa hàng chục của phút. Ngoài ra bit thứ 7 luôn bằng 0.

**Thanh ghi giờ (địa chỉ 02h):** 4 bit thấp của thanh ghi này chứa hàng đơn vị của giờ, bit thứ 6 quy định chế độ 12 h (bit 6 =1) hoặc 24 h (bit 6=0). Nếu ở chế độ 24h thì bit 4 và bit 5 quy định hàng chục của giờ. Nếu chế độ 12h thì bit 4 quy định hàng chục, bit 5 khi đó quy định (PM: buổi chiều hoặc AM: Buổi sáng

**Thanh ghi thứ (địa chỉ 03h):** ba bit đầu quy định thứ trong tuần (Ví dụ 1 là Chủ Nhật, 2 là Thứ hai, 3 là thứ 3 ...). Các bit còn lại luôn bằng 0.

**Thanh ghi ngày(địa chỉ 04h):** 4 bit đầu lưu hàng đơn vị của ngày, bit 4 và bit 5 quy định hàng chục. Bit 6 và bit 7 luôn luôn bằng 0.

**Thanh ghi tháng(địa chỉ 05h):** 4 bit đầu lưu hàng đơn vị của tháng, bit thứ 4 quy định hàng chục. Các bit còn lại luôn bằng 0.

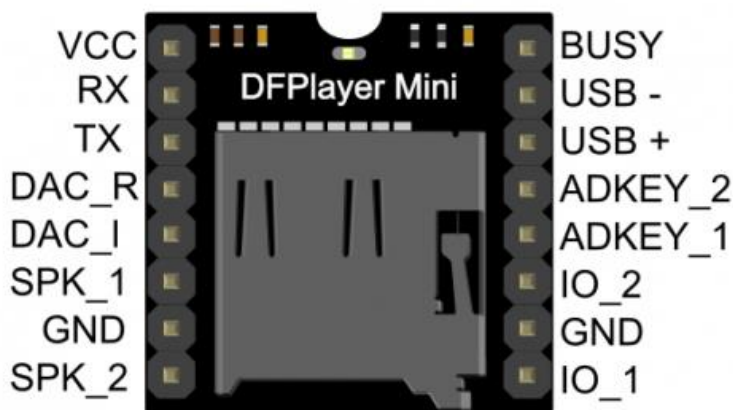
**Thanh ghi năm (địa chỉ 06h):** 4 bit thấp lưu hàng đơn vị và 4 bit cao hàng chục của năm (từ 00 - 99).

### 2.1.3.4. Lập trình Module Ds1307

Chúng ta cần các thư viện giúp ra bóc rạch dữ liệu được phân tích phía trên được dễ dàng. Gồm có rất nhiều thư viện gồm `#include <Wire.h>`, `#include <ds1307.h>`. Bạn chỉ add thư viện vào IDE (phần mềm Arduino). Cách sử dụng ví dụ bạn dùng thư viện bạn chỉ cần gõ lên.

### 2.1.4. Module DFplayer mini

#### 2.1.4.1 Giới thiệu về DFplayer



Hình 2.16: Module DFPlayer Mini

## CHƯƠNG II: GIỚI THIỆU CÁC MODULE DS1307, DFplayer mini

DFPlayer Mini MP3 [4] là mạch phát tập tin âm thanh kiểu máy chơi thông tin MP3. Mạch có thể được sử dụng riêng lẻ chỉ cần pin, loa và nút nhấn hoặc kết hợp với vi điều khiển có giao tiếp chuẩn USART.

### 2.1.4.2 Tính năng cơ bản gồm có:

- Hỗ trợ tập tin mp3, wmv với hầu hết loại tần số lấy mẫu chuẩn.
- Lỗi ra DAC 24 bit âm thanh chuẩn
- Thẻ nhớ định dạng FAT16, 32 lên đến 32GB
- Đọc dữ liệu trong tối đa 100 thư mục, mỗi thư mục 255 bài
- Âm lượng 30 mức với 6 loại EQ
- Có sẵn mạch khuếch đại ra loa công suất 2W

### 2.1.4.3 Mạch có thể được sử dụng trong nhiều ứng dụng:

- Hướng dẫn phân luồng giao thông
- Trạm thông báo, nhắc nhở
- Báo động hoặc cảnh báo ngoài trời, trong nhà
- Quảng báo bán hàng
- Hướng dẫn thông tin, du lịch...

Pin	Description	Note
VCC	Input Voltage	DC3.2~5.0V;Type: DC4.2V
RX	UART serial input	
TX	UART serial output	
DAC_R	Audio output right channel	Drive earphone and amplifier
DAC_L	Audio output left channel	Drive earphone and amplifier
SPK2	Speaker-	Drive speaker less than 3W
GND	Ground	Power GND
SPK1	Speaker+	Drive speaker less than 3W
IO1	Trigger port 1	Short press to play previous (long press to decrease volume)
GND	Ground	Power GND
IO2	Trigger port 2	Short press to play next (long press to increase volume)
ADKEY1	AD Port 1	Trigger play first segment
ADKEY2	AD Port 2	Trigger play fifth segment
USB+	USB+ DP	USB Port
USB-	USB- DM	USB Port
BUSY	Playing Status	Low means playing \High means no



### 2.1.4.4 Chế độ hoạt động

#### 1. Serial mode

Có hỗ trợ trên giao thức serial của cổng com trên máy tính.

Tốc độ board: 9600 bps

Bit Data: 1 bytes

Check Sum: none

Bit Parity: none

CMD	Function Description	Parameters(16 bit)
0x01	Next	
0x02	Previous	
0x03	Specify tracking(NUM)	0-2999
0x04	Increase volume	
0x05	Decrease volume	
0x06	Specify volume	0-30
0x07	Specify EQ(0/1/2/3/4/5)	Normal/Pop/Rock/Jazz/Classic/Base
0x08	Specify playback mode (0/1/2/3)	Repeat/folder repeat/single repeat/ random
0x09	Specify playback source(0/1/2/3/4)	U/TF/AUX/SLEEP/FLASH
0x0A	Enter into standby – low power loss	
0x0B	Normal working	
0x0C	Reset module	
0x0D	Playback	
0x0E	Pause	
0x0F	Specify folder to playback	1~10(need to set by user)
0x10	Volume adjust set	{DH= 1:Open volume adjust } {DL: set volume gain 0~31}
0x11	Repeat play	{1:start repeat play} {0:stop play}

*Hình 2.17: Chế độ Serial*

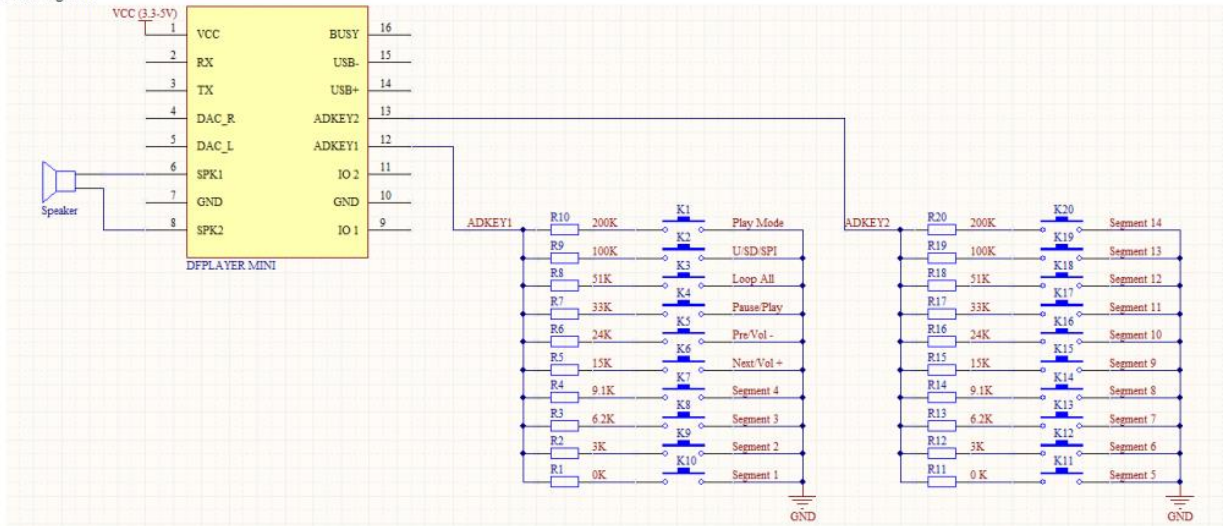


## CHƯƠNG II: GIỚI THIỆU CÁC MODULE DS1307, DFplayer mini

### 2. Ad key mode

Chúng tôi sử dụng các phím mô-đun AD, thay vì phương pháp kết nối bàn phím ma trận truyền thống, đó là tận dụng chức năng MCU AD ngày càng mạnh mẽ, Cấu hình mặc định của mô-đun 2 cổng AD, 20 nút nhấn.

● Refer diagram

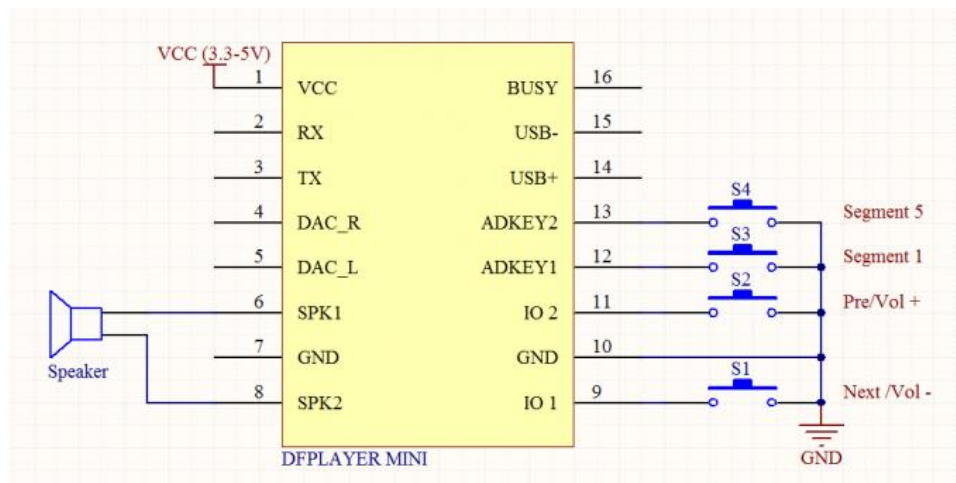


Hình 2.18: Ad key mode

### 3) I/O mode

Đây là cách đơn giản để sử dụng Module.

● Refer diagram



Hình 2.19: I/O mode

Trong chế độ này thì các nút S1 và S2 được nhấn trong thời gian ngắn ở port IO2, IO1 là play bài tiếp theo hoặc là play bài ngược lại, còn nếu nhấn lâu hơn thì sẽ chuyển chế độ tăng hoặc giảm âm thanh.

### 2.1.4.5 Lập trình cho DFplayer mini

Chúng ta đã tạo một thư viện Arduino cho DFPlayer Mini để đơn giản hóa phương pháp giúp làm việc. Kết nối phần cứng như hình trên được hiển thị và phát cùng với mã mẫu. Bạn có thể tải xuống thư viện mới nhất tại đây: **DFRobotDFPlayerMini**.

### 2.1.6 Màn hình TFT 3.2 và TFT shield LCD module



*Hình 2.20: Màn hình TFT 3.2*

#### 2.1.6.1 Màn hình TFT 3.2 có chức năng là gì?

##### Màn hình TFT – LCD

Màn hình **TFT** (Thin Film Transistor – bóng bán dẫn dạng phim mỏng) [5] bắt đầu được đưa vào smartphone vào năm 2005, có khả năng tái tạo màu tốt hơn và độ phân giải hình ảnh cao hơn so với các màn hình LCD thế hệ trước đó. Do chi phí sản xuất màn hình TFT đã giảm đáng kể từ sau 2005, công nghệ màn hình này đã xuất hiện phổ biến trên điện thoại cơ bản và smartphone giá thấp.



Các bạn đã từng trải nghiệm nhiều phiên bản LCD như LCD 16x02, 20x04, 5110 nokia, Oled, Homephone,... nhưng đối với LCD TFT thì nó sẽ lung linh hơn nhiều, thư viện UTFT thì lại cũng dễ sử dụng như một số LCD cơ bản- Với LCD TFT chúng ta sẽ DIY với những cảm biến như gia tốc, nhiệt độ, ánh sáng, lập một menu button điều khiển xe không cần android ... thậm chí nó có thể mang đến cảm giác hoàn hảo tuyệt vời khi chế tạo cùng với chế độ điều khiển servo, robot, led, làm menu smartphone, vận hành máy in3D với loại TFT cao cấp hơn...

#### **2.1.6.2. Thông số kỹ thuật**

Ø Điện áp hoạt động: 5V

Ø Độ phân giải 240x320 với 262K màu

Ø Chuẩn giao tiếp 2 chế độ 8 bit và 16 bit.

Ø Hỗ trợ cảm ứng điện trở, khe cắm thẻ nhớ SD

Sơ đồ

Số thứ tự	Tên chân	Chức năng
1	GND	Nối mass nguồn 5V
2	VCC	Nối nguồn 5V
3,18,19,20,21,22,29	NC	Not connect
5	WR	1 xung cạnh lên trên chân WR, sẽ ghi data lên LCD
6	RD	1 xung cạnh lên trên chân RD, đọc trạng thái LCD

<b>7-14</b>	<b>DB8-DB15</b>	<b>8 bit cao data</b>
<b>15</b>	<b>CS</b>	<b>F-CS: Sử dụng bộ nhớ flash ngoài</b>
<b>16</b>	<b>F-CS</b>	<b>F-CS: Sử dụng bộ nhớ flash ngoài</b>
<b>17</b>	<b>REST</b>	<b>Reset tích cực âm</b>
<b>19</b>	<b>LED_A</b>	<b>Chân Anot đèn nền</b>
<b>23</b>	<b>SD_CS</b>	<b>Sử dụng giao tiếp SPI với thẻ nhớ SD (chân chọn Slave)</b>
<b>24</b>	<b>SD_DIN</b>	<b>Sử dụng giao tiếp SPI với thẻ nhớ SD (gửi lệnh command qua chân này)</b>
<b>25</b>	<b>SD_CLK</b>	<b>Sử dụng giao tiếp SPI với thẻ nhớ SD (chân tạo xung clock)</b>
<b>26</b>	<b>SD_DO</b>	<b>Sử dụng giao tiếp SPI với thẻ nhớ SD (chân đọc data từ thẻ nhớ SD)</b>
<b>27</b>	<b>T-IRQ</b>	<b>Khi có cảm ứng, chân này được kéo xuống mức 0.</b>
<b>28</b>	<b>T-DO</b>	<b>Sử dụng chức năng cảm ứng, giao tiếp SPI (đọc data-nối MISO)</b>
<b>30</b>	<b>T-DIN</b>	<b>Sử dụng chức năng cảm ứng, giao tiếp SPI (gửi data-nối MOSI)</b>
<b>31</b>	<b>T-CS</b>	<b>Sử dụng chức năng cảm ứng, giao tiếp SPI (chọn Slave)</b>
<b>32</b>	<b>T-CLK</b>	<b>Sử dụng chức năng cảm ứng, giao tiếp SPI (chân tạo xung clock)</b>
<b>33-40</b>	<b>DB7-DB0</b>	<b>8 bit thấp data(Nối GND trong mode 8 bits)</b>

**Hình 2.21: Bảng sự thật TFT 3.2**

### **2.1.6.3 Phương thức giao tiếp.**

Giao tiếp TFT với vi điều khiển: Sử dụng vào ra IO, giao tiếp SPI

Chức năng screen (hiển thị):

+Giao tiếp 8 bit: Sử dụng 8 bit cao, 8 bit thấp nối GND.

+Giao tiếp 16 bit: Sử dụng 8 bit cao và 8 bit thấp.

Trong chế độ 8 bit: Gửi 8 bit cao trước, sau đó gửi 8 bit thấp.

Ví dụ : VH, VL theo thứ tự là 8 bits cao và 8 bits thấp của data.

```
xuat_data_8 bit(VH);  
LCD_WR_OFF;  
LCD_WR_ON;  
xuat_data_8 bit(VL);  
LCD_WR_OFF;  
LCD_WR_ON;
```

- Nhận xét:

- + Ưu điểm: Tiết kiệm được chân vi điều khiển (chỉ sử dụng 8 chân để truyền data cho màn hình TFT)
- + Nhược điểm: Tốc độ chậm hơn so với 16 bit.

Trong chế độ 16 bit: Sử dụng cả 16 chân DB0-DB15 để truyền data

Sample: VH, VL theo thứ tự là 8 bits cao và 8 bits thấp của data.

```
PORT_H_Data=VH;  
LCD_WR_OFF;  
LCD_WR_ON;  
PORT_L_Data=VL;  
LCD_WR_OFF;  
LCD_WR_ON;
```

- Nhận xét:

- + Ưu điểm: Tốc độ nhanh vì chế độ là song song, trực tiếp.
  - + Nhược điểm: Không tiết kiệm chân vi điều khiển sử dụng
- Quá trình khởi tạo màn hình TFT (theo thứ tự hàm init)

\* Đưa chân CS =1, tạo 1 xung cạnh xuống và 1 xung cạnh lên trên chân RESET để đánh thức IC màn hình TFT

\* Tiếp tục đưa chân RD và chân WR =1 để ghi lệnh cho TFT.

### 2.1.7. Arduino mega 2560

Mega 2560 [7] là board mạch vi điều khiển dựa trên ATmega2560. Nó có 54 chân đầu vào / đầu ra kỹ thuật số (trong đó có 14 đầu ra PWM), 16 đầu vào analog, 4 UART (cổng nối tiếp phần cứng), bộ dao động tinh thể 16 MHz, kết nối USB, giắc cắm nguồn, tiêu đề ICSP và nút đặt lại. Nó chứa mọi thứ cần thiết để hỗ trợ vi điều khiển; chỉ cần kết nối nó với một máy tính bằng cáp USB hoặc cấp điện cho nó bằng bộ chuyển đổi AC-to-DC hoặc pin để bắt đầu. Mega tương thích với hầu hết các khiên được thiết kế cho Arduino Duemilanove hoặc Diecimila.

## Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

**Hình 2.22: Arduino mega 2560**

Mega 2560 là bản cập nhật cho Arduino Mega. Tóm lược vi điều khiển ATmega2560, điện áp hoạt động 5V, điện áp đầu vào (được khuyến nghị) 7-12V, điện áp đầu vào (giới hạn) 6-20V, chân I / O kỹ thuật số 54 (trong đó 14 cung cấp đầu ra PWM). Pins đầu vào tương tự 16, DC hiện tại cho mỗi I / O Pin 40 mA DC hiện tại cho 3.3V Pin 50 mA, bộ nhớ flash 256 KB trong đó 8 KB được trình tải khởi động sử dụng SRAM 8 KB, EEPROM 4 KB, xung clock : 16 MHz.

CỔNG SERIAL	CHÂN RX	CHÂN TX
Cổng 0	0	1
Cổng 1	19	18
Cổng 2	17	16
Cổng 3	15	14



### 2.1.8. Loa 3 w



Tính năng chính:

Loa màng rung 8 ohm 0.5W có chất lượng hoàn thiện tốt, được sử dụng rộng rãi cho các dự án âm thanh của bạn. Lắp đặt dễ dàng, thuận tiện.

Ứng dụng: loa 8 ohm 0.5W thích hợp cho việc chế tạo loa hộp mini, máy nghe thông tin, chuông cửa, máy tính, loa đa phương tiện, các dự án về cảnh báo âm thanh, âm ly đơn giản, các dự án chế tạo robot...

Thông số kỹ thuật:

Trở kháng: 8ohm.

Công suất: 3W.

Hình dạng: dạng tròn.

Đường kính loa: 50mm.

Chất liệu: nhôm, nhựa, nam châm.

Trọng lượng: 42g.

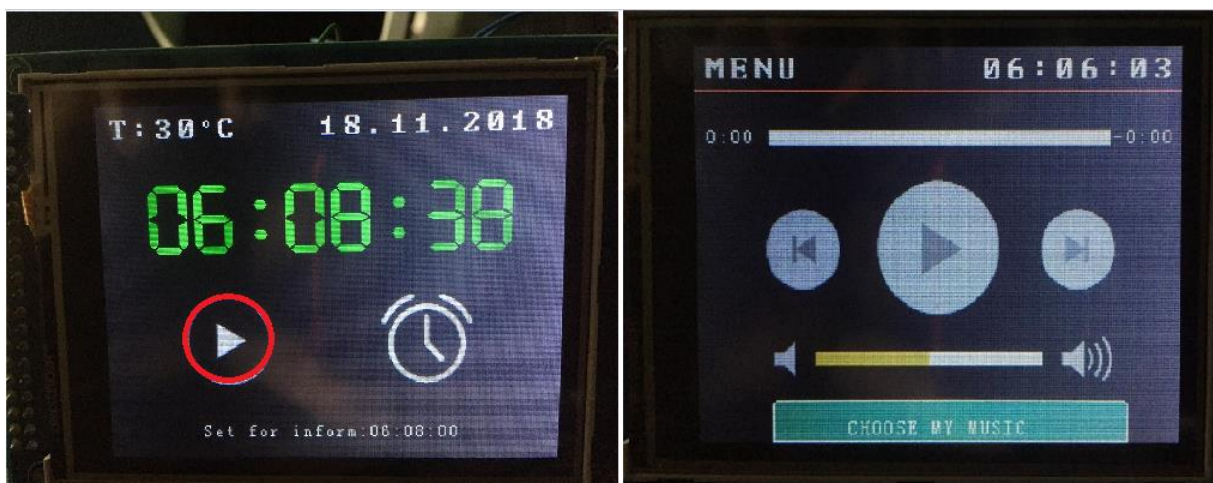
## 2.2. Giao diện người dùng trên màn hình 3.2 TFT

### 2.2.1 Các thanh công cụ điều khiển

Ở màn hình cảm ứng giúp ta thao tác dễ dàng hơn là dùng nút bấm như cách thông thường, ở đây em có các hình ảnh cho thêm sinh động để tạo ra các giao diện người dùng. Gồm có các hình AlarmButton, ButtonPlay, ButtonPause,.. điều khiển tắt mở thông tin, chọn folder để phát thông tin theo yêu cầu.



### 2.2.2 Chọn bài hát và phát thông tin theo yêu cầu

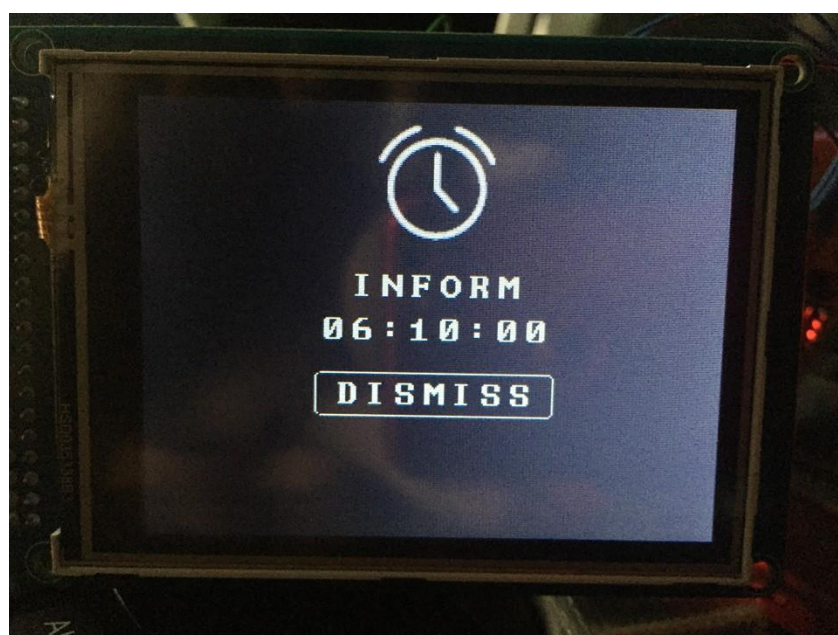


Vào ButtonPlay ta sẽ vào track bar đó là một khung chơi thông tin như tăng, giảm âm thanh, nhấn bài tiếp theo, lùi bài hát và ta có thể chọn bài hát bằng folder như thông tin hàn quốc, quảng cáo, để có thể chọn bài hát để phát.





Vào AlarmButton ta thiết lập giờ sẵn để so sánh với thời gian thực nếu trùng thì sẽ phát thông tin và hiện thông báo đã đến giờ phát.



### 2.3. Giới thiệu tổng quát về Blynk

#### 2.3.1 App Blynk là gì?

Blynk [6] là một ứng dụng chạy trên nền tảng iOS và Android để điều khiển và giám sát thiết bị thông qua internet. Blynk không bị ràng buộc với những phần cứng cụ thể nào cả, thay vào đó, nó hỗ trợ phần cứng cho bạn lựa chọn như Arduino, Raspberry Pi, ESP8266 và nhiều module phần cứng phổ biến khác.



*Hình 2.23: App Blynk*

#### 2.3.2 Tại sao phải dùng App Blynk

- Dễ sử dụng: việc cài đặt ứng dụng và đăng ký tài khoản trên điện thoại rất đơn giản cho cả IOS và Android
- Chức năng phong phú: Blynk hỗ trợ rất nhiều chức năng với giao diện đẹp và thân thiện, bạn chỉ việc kéo thả đối tượng và sử dụng nó.
- Không phải lập trình ứng dụng: nếu bạn không có kiến thức về lập trình app cho Android cũng như IOS thì Blynk là một ứng dụng tuyệt vời để giúp bạn khám phá thế giới IOTs.
- Điều khiển, giám sát thiết bị ở bất kì đâu thông qua internet với khả năng đồng bộ hóa trạng thái và thiết bị.
- **Dễ sử dụng:** Quá đơn giản, chỉ việc vào store, cài đặt, sau đó đăng ký tài khoản và mất không quá 5 phút để làm quen.
- **Đẹp và đầy đủ:** Giao diện của Blynk quá tuyệt vời, sử dụng bằng cách kéo thả, bạn cần nút bấm, kéo thả nút bấm, bạn cần đồ thị, kéo thả đồ thị, bạn cần LCD, kéo thả LCD, tóm lại là bạn cần gì thì kéo thả cái đó.
- **Thử nghiệm nhanh chóng, có thể điều khiển giám sát ở bất kỳ nơi nào có internet.**

### 2.4. Hướng dẫn sử dụng App Blynk

Blynk thực ra là một cái app trên điện thoại, cho phép người dùng có thể tạo ra giao diện và điều khiển thiết bị theo ý thích của cá nhân:

Tất nhiên ngoài những điểm lợi từ blynk thì còn có những cái hạn chế như phải mua energy để tạo được nhiều giao diện và chia sẻ giao diện cho người khác. Những cái này cũng không phải là vấn đề lớn lắm.

#### 2.4.1 Tải ứng dụng Blynk trên ANDROID hoặc IOS

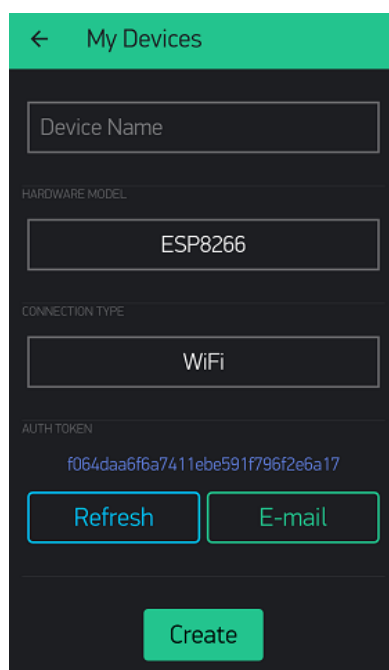
**BLYNK CHO ANDROID**

**BLYNK CHO IPHONE**

#### 2.4.2 Lấy mã AUTH TOKEN

Để kết nối với ứng dụng Blynk và phần cứng của bạn, bạn cần có một mã Token Xác thực.

1. Tạo tài khoản trên ứng dụng Blynk của máy chủ Việt Nam. Phần Custom điền (IP: cloud.blynk.vn và Port: 8443)
2. Tạo một dự án mới. Sau đó, chọn bảng kết nối mà bạn sẽ sử dụng.
3. Sau khi tạo dự án mới thành công, bạn cần sao chép hoặc gửi mã xác thực Auth Token qua email tài khoản.
4. Kiểm tra email trong hộp thư đến và tìm mã xác thực Auth Token.



*Hình 2.24 Auth token*

### 2.4.3 Cài đặt thư viện Blynk

- Thư viện Blynk nên được cài đặt bằng tay. Làm theo hướng dẫn:

TẢI THƯ VIỆN BLYNK VIỆT NAM

1. Tải thư viện **Blynk.zip** phía trên.
2. Giải nén thư viện **Blynk.zip**. Bạn sẽ thấy thư mục chứa thư viện Blynk.
3. Sao chép thư mục **Blynk** vào **thư viện libraries trên Arduino IDE** (thư mục libraries trên **Arduino IDE**. Nếu không thấy xin vào trên phần mềm **Arduino IDE: Chọn File -> Preferences** (lúc này sẽ thấy đường dẫn chứa thư viện libraries )

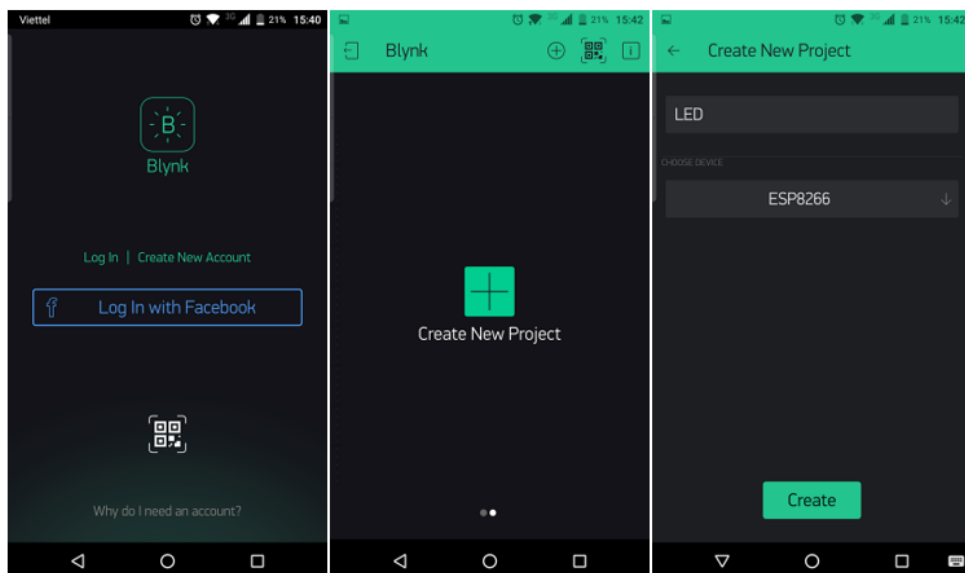
Ví Dụ: **Documents\Arduino\libraries\Blynk...**

Lưu ý rằng thư viện Blynk sẽ hoạt động nếu bạn chọn đúng đường dẫn thư viện.

### 2.4.4 Tạo code ví dụ

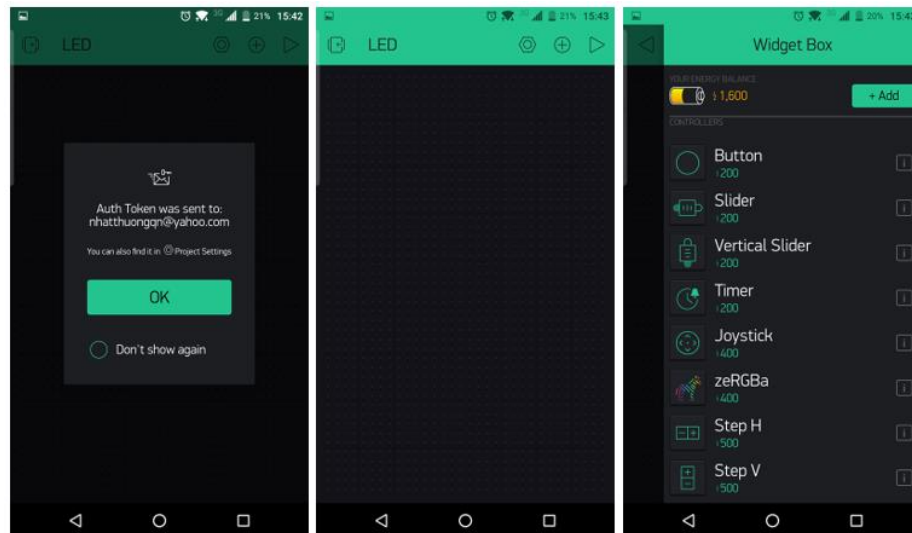
Để kết nối với ứng dụng Blynk và bộ điều khiển bạn cần có một mã Token xác thực. Trước hết bạn phải có một tài khoản gmail, yahoo ,Facebook.... hoặc chúng ta tạo một tài khoản mới. Sau khi có tài khoản, khởi động chương trình Blynk được cài đặt trên điện thoại lúc này giao diện blynk và thực hiện các bước như hình sau:

1. Tạo một project, đây được hiểu giống như là một ứng dụng.
2. Điền tên Project và chọn Board phần cứng (Các bạn có thể chọn NodeMCU hoặc ESP8266)



Hình 2.25 Giao diện của App Blynk

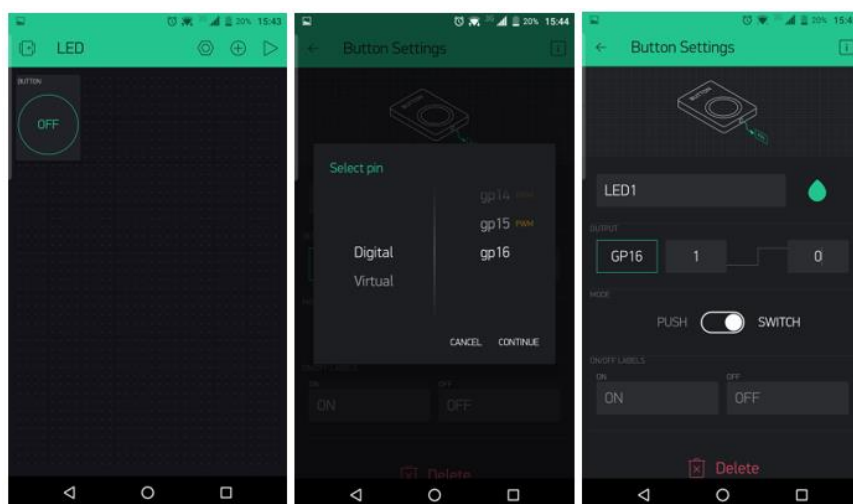
Mỗi project, Blynk sẽ gửi cho bạn 1 mã Auth Token để nhập vào trong code của Board mạch điều khiển NodeMCU, Các bạn có thể lựa chọn nhiều chức năng như nút bấm, hẹn giờ, LCD... để đưa vào project của mình. Mỗi đối tượng các bạn chọn sẽ tốn energy (1 đơn vị giới hạn khi bạn dùng server miễn phí).



**Hình 2.26. Thanh công cụ của Blynk**

Sau khi chọn xong đối tượng, các bạn nhấp vào để cấu hình chân, các mức logic...

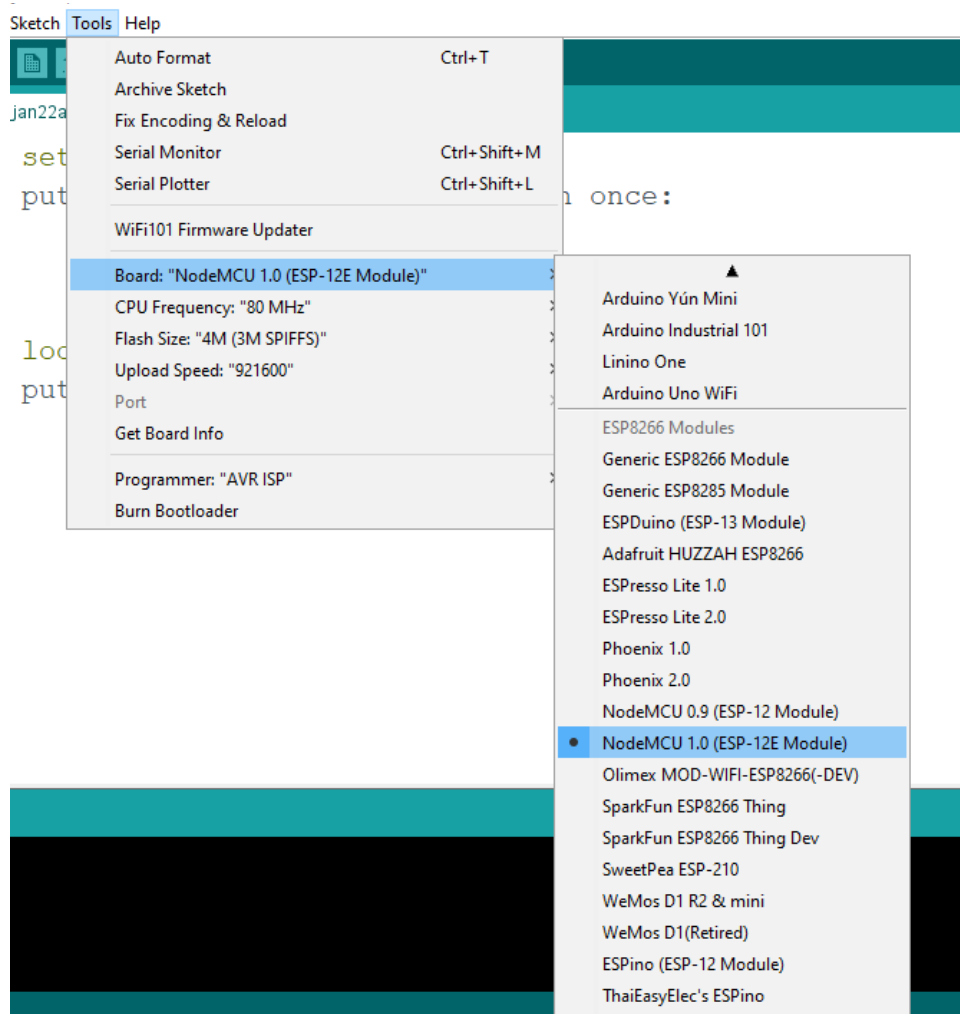
Các cấu hình này sẽ tác động đến board phần cứng. Ví dụ nhất nút thì chân GPIO16 sẽ chuyển từ trạng thái logic 1 sang logic 0:



**Hình 2.27 Thiết lập công cụ trên Blynk**

Sau khi cài đặt xong trên điện thoại thì các bạn lập trình cho board phần cứng. Thư viện Blynk trên Arduino IDE hỗ trợ rất nhiều ví dụ cho Blynk để các bạn có thể dùng thử, hiểu cách thức hoạt động...





Hình 2.28. Nạp vào ESP8266

### 2.4.5 Dán mã xác thực

- Tên code ví dụ bạn tìm dòng code dưới đây:

```
char auth[] = "YourAuthToken";
```

- Sửa mã xác thực YourAuthToken (Mã xác thực nằm trong email của bạn sau khi bạn tạo dự án trong ứng dụng Blynk

Ví dụ: `char auth[] = "53e4da8793764b6197fc44a673ce4e21";`

- Upload code cho phần cứng của bạn
- Mở seri monitor trên Arduino IDE. Bạn sẽ thấy nó như thế này

```
____ _
/_)//_ _ _ _ _ // _
/_ _ /// _ \ ' _/
/_ _/_/_ ,/_/_/_ \
    / _/ v0.4.4
```

[1240] Connecting to YourWiFi

[1240] Connected to YourWiFi

[1240] My IP: 192.168.10.172

[1240] Blynk v0.4.4

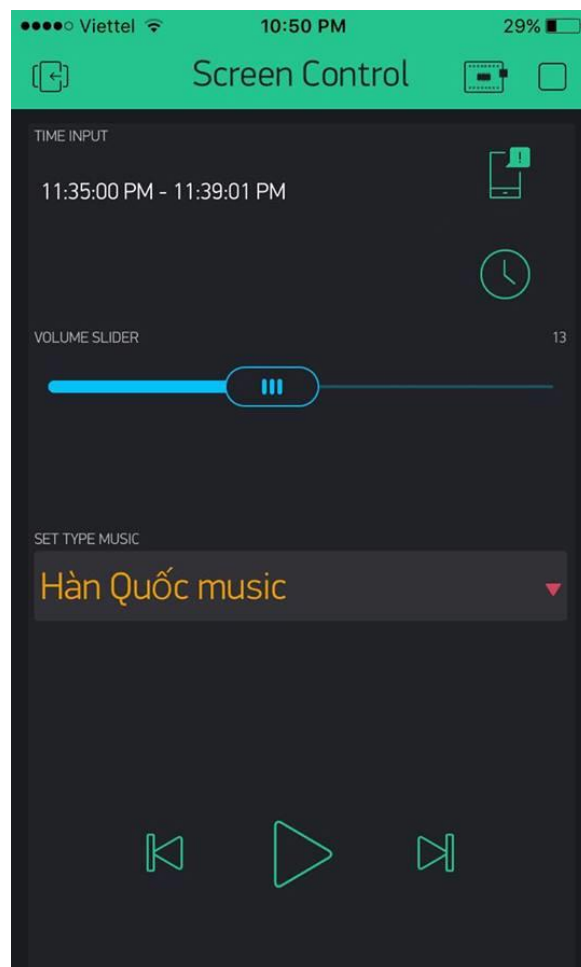
[5001] Connecting to cloud.blynk.vn:8442

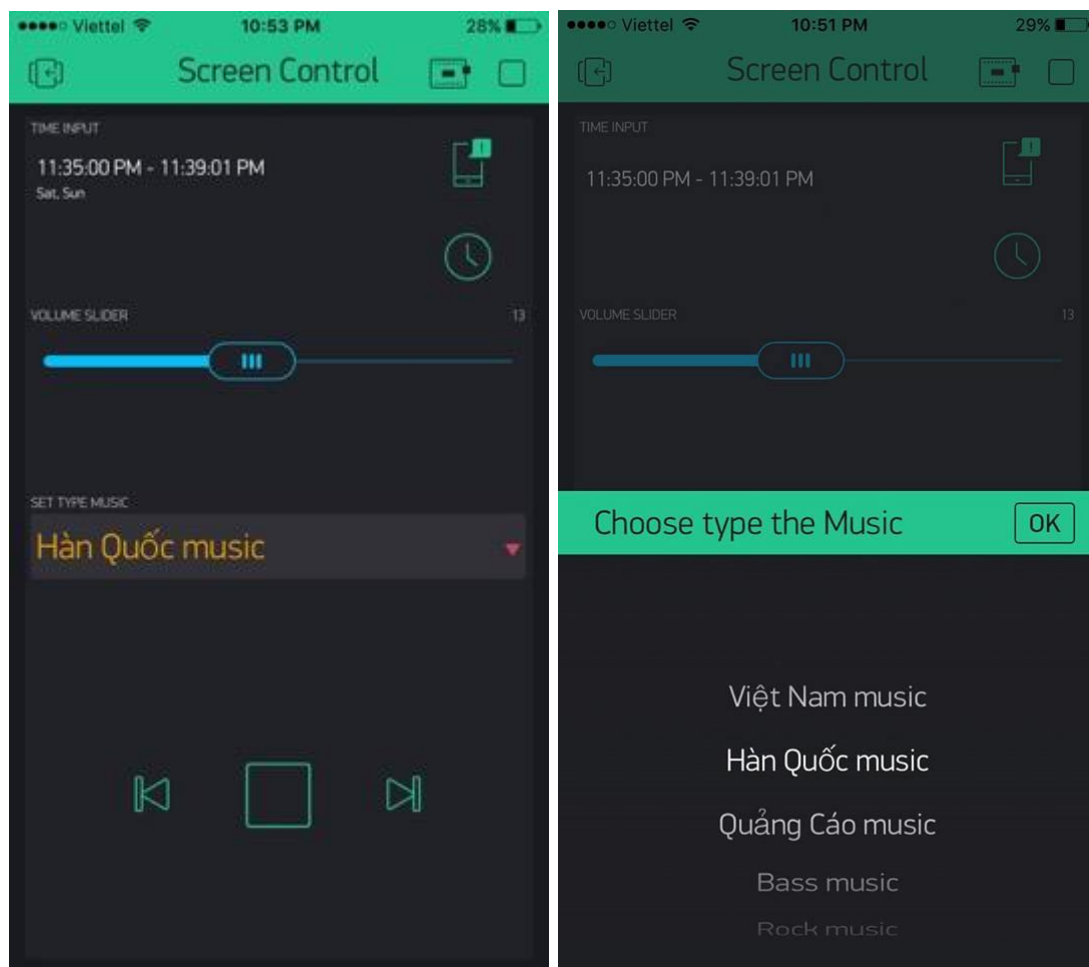
[5329] Ready (ping: 1ms)

**Hoàn tất! nó đã kết nối!**

## 2.5. Xây dựng app điều khiển bằng qua Blynk

### 2.5.1 Gồm những chức năng



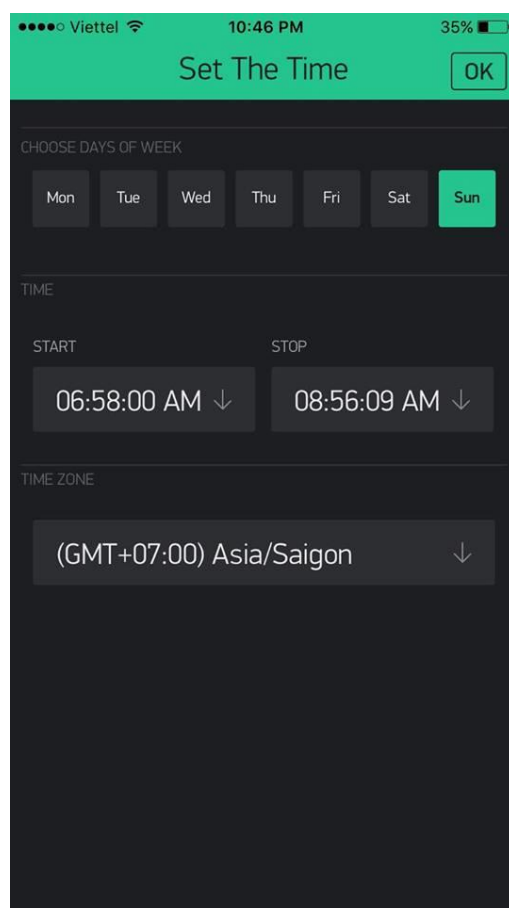


Gồm những thanh công cụ điều khiển thông tin như tăng giảm âm lượng được sử dụng bởi thanh trượt trong app, ở đây app sẽ ghi dữ liệu khi thanh trượt dừng lại, gồm các nút nhấn cho bài tiếp theo hoặc lùi trở lại, pause hoặc stop không những thế nó có thể có 2 chế độ chính là tự động phát tất cả các bài hát và chế độ bình thường.

### 2.5.2 Hẹn giờ phát thông tin

Sau đây là chế độ chính của mạch là phát thông tin theo thời gian định sẵn, chúng ta sẽ có thời gian và ngày để bắt đầu phát ngày kết thúc cũng vậy, ở đây ta thiết lập ở múi giờ (GMT+07:00) Sài gòn nếu muốn có thể chọn múi giờ khác.





### 2.6.1 Khái niệm cơ bản về giao thức Modbus

Sau các khái niệm cơ bản về các chuẩn giao tiếp vật lý cơ bản mà chúng ta hay thấy như RS232, RS485 (còn gọi là serial), hôm nay chúng ta cùng trao đổi về giao thức Modbus, một giao thức được sử dụng rộng rãi trong công nghiệp. Nói một cách dễ hiểu, chúng ta có thể tưởng tượng RS232, RS485 là phần xác, còn giao thức như Modbus là phần hồn.

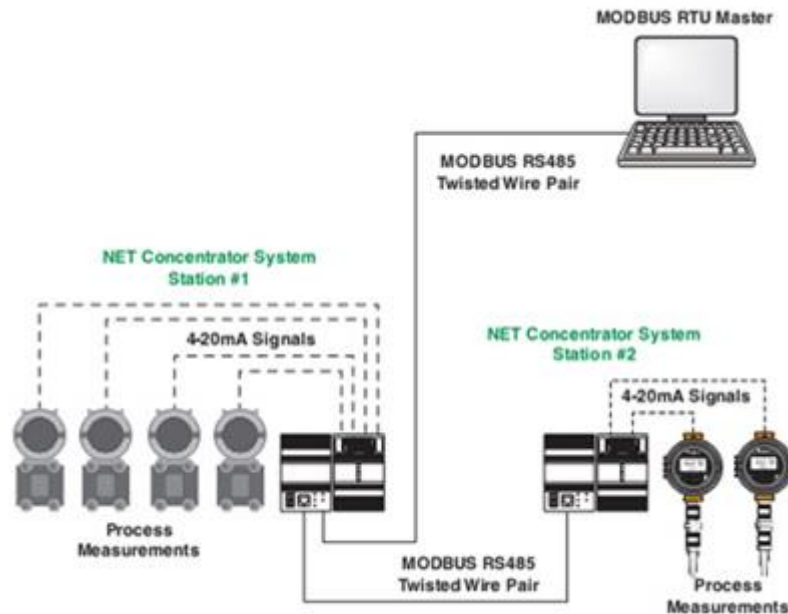
#### *Vậy, giao thức Modbus là gì?*

MODBUS [8] do Modicon (hiện nay thuộc Schneider Electric) phát triển năm 1979, là một phương tiện truyền thông với nhiều thiết bị thông qua một cặp dây xoắn đơn. Ban đầu, nó hoạt động trên RS232, nhưng sau đó nó sử dụng cho cả RS485 để đạt tốc độ cao hơn, khoảng cách dài hơn, và mạng đa điểm (multi-drop). MODBUS đã nhanh chóng trở thành tiêu chuẩn thông dụng trong ngành tự động hóa, và Modicon đã cho ra mắt công chúng như một protocol miễn phí.

MODBUS là một hệ thống “chủ - tớ”, “chủ” được kết nối với một hay nhiều “tớ”. “Chủ” thường là một PLC, PC, DCS, hay RTU. “Tớ” MODBUS RTU thường là các thiết bị hiện trường, tất cả được kết nối với mạng trong cấu hình multi-drop. Khi một chủ MODBUS RTU muốn có thông tin từ thiết bị, chủ sẽ gửi một thông điệp về dữ

liệu cần, tóm tắt dò lỗi tới địa chỉ thiết bị. Mọi thiết bị khác trên mạng sẽ nhận thông điệp này nhưng chỉ có thiết bị nào được chỉ định mới có phản ứng.

Các thiết bị trên mạng MODBUS không thể tạo ra kết nối; chúng chỉ có thể phản ứng. Nói cách khác, chúng “lên tiếng” chỉ khi được “nói tới”. Một số nhà sản xuất đang phát triển các thiết bị lai ghép hoạt động như các tớ MODBUS, tuy nhiên chúng cũng có “khả năng viết”, do đó làm cho chúng trở thành các thiết bị chủ ảo.



**Hình 2.29: Sơ đồ kết nối modbus**

### 2.6.1 Các chuẩn modbus nào đang được sử dụng phổ biến?

Hiện nay, [5] có 03 chuẩn modbus đang được sử dụng phổ biến trong công nghiệp - tự động hóa là: Modbus RTU, Modbus ASCII, Modbus TCP.

Vậy, 03 chuẩn này có gì giống và khác nhau?

Tất cả thông điệp được gửi dưới cùng một format. Sự khác nhau duy nhất giữa 3 loại MODBUS là cách thức thông điệp được mã hóa. Cụ thể:

- **Modbus ASCII:**

Mọi thông điệp được mã hóa bằng hexadeci-mal, sử dụng đặc tính ASCII 4 bit. Đối với mỗi một byte thông tin, cần có 2 byte truyền thông, gấp đôi so với MODBUS RTU hay MODBUS/TCP. Tuy nhiên, MODBUS ASC II chậm nhất trong số 3 loại protocol, nhưng lại thích hợp khi modem điện thoại hay kết nối sử dụng sóng radio do ASC II sử dụng các tính năng phân định thông điệp. Do tính năng phân định này, mọi rắc rối trong phương tiện truyền dẫn sẽ không làm thiết bị nhận dịch sai thông tin. Điều này quan trọng khi đề cập đến các modem

chậm, điện thoại di động, kết nối ổn hay các phương tiện truyền thông khó tính khác.

- Modbus RTU: Dữ liệu được mã hóa theo hệ nhị phân, và chỉ cần một byte truyền thông cho một byte dữ liệu. Đây là thiết bị lý tưởng đối với RS 232 hay mạng RS485 đa điểm, tốc độ từ 1200 đến 115 baud. Tốc độ phổ biến nhất là 9600 đến 19200 baud. MODBUS RTU là protocol công nghiệp được sử dụng rộng rãi nhất, do đó hầu như trong bài viết này chỉ tập trung đề cập đến cơ sở và ứng dụng của nó.
- Modbus TCP: MODBUS/TCP đơn giản là MODBUS qua Ethernet. Thay vì sử dụng thiết bị này cho việc kết nối với các thiết bị từ, do đó các địa chỉ IP được sử dụng. Với MODBUS/TCP, dữ liệu MODBUS được tóm lược đơn giản trong một gói TCP/IP. Do đó, bất cứ mạng Ethernet hỗ trợ MODBUS/ IP sẽ ngay lập tức hỗ trợ MODBUS/TCP.

*Modbus gateway là gì?*

Modbus gateway là một thiết bị cho phép chuyển đổi qua lại giữa giao thức Modbus RTU và Modbus TCP. Thông thường thiết bị sẽ có 01 cổng serial (RS232/RS485) và 01 cổng Ethernet.

Tại sao lại là 2 giao thức này? vì hai giao thức này phổ biến, đại diện cho 02 loại cổng vật lý là serial (RS232/RS485) và ethernet (cổng RJ45). Thiết bị công nghiệp dùng ở thị trường hiện nay, nhiều thiết bị chỉ hỗ trợ Modbus RTU, hoặc chỉ hỗ trợ Modbus TCP, hoặc hỗ trợ cả 2. Do đó, để kết nối các thiết bị trường này vào hệ thống modbus chung của nhà máy, xí nghiệp thì sẽ cần 01 thiết bị phiên dịch được gọi là Modbus Gateway.

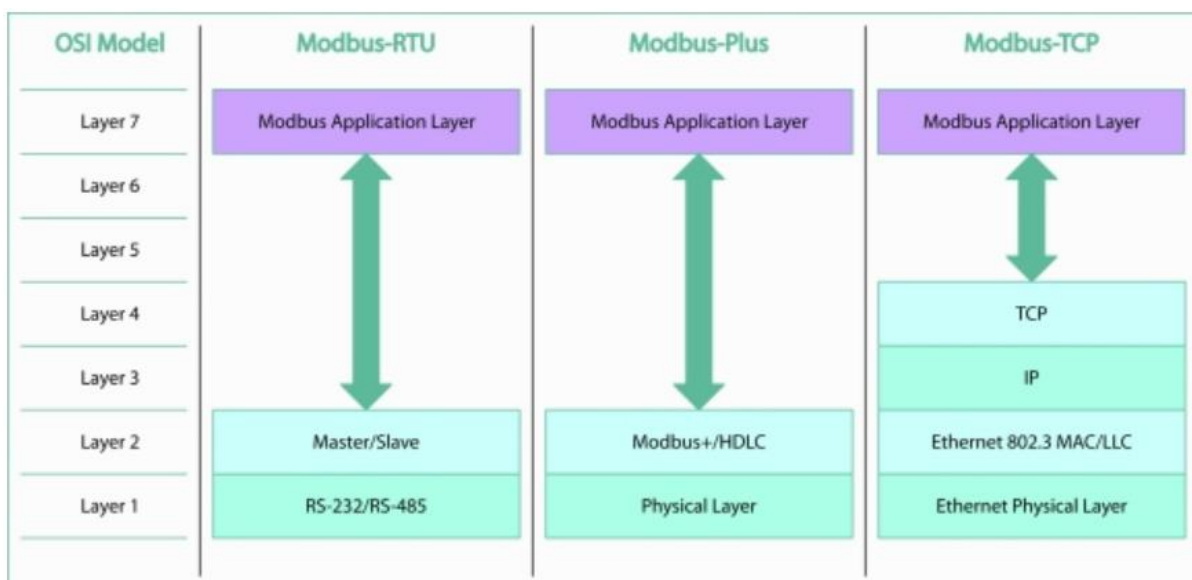
### 2.6.2 Khái niệm cơ bản về giao thức Modbus RTU

Giao thức Modbus RTU là một giao thức mở, sử dụng đường truyền vật lý RS-232 hoặc RS485 và mô hình dạng Master-Slave. Đây là một giao thức được sử dụng rộng rãi trong nhiều lĩnh vực như BMS (Building Management Systems), tự động hóa, công nghiệp, điện lực, Chắc hẳn sẽ có bạn tự hỏi, tại sao giao thức Modbus này lại thông dụng như thế, đi đến đâu, đụng vào thiết bị gì thì hầu như cũng có giao thức này? Vâng, câu trả lời cho câu hỏi trên chắc chỉ cần gói gọn trong vài từ: Ổn định - Đơn giản - Dễ dùng.

Modbus được coi là giao thức truyền thông hoạt động ở tầng "Application", cung cấp khả năng truyền thông Master/Slave giữa các thiết bị được kết nối thông qua các bus hoặc network. Trên mô hình OSI, Modbus được đặt ở lớp 7. Modbus được xác

## CHƯƠNG II: GIỚI THIỆU CÁC MODULE DS1307, DFplayer mini

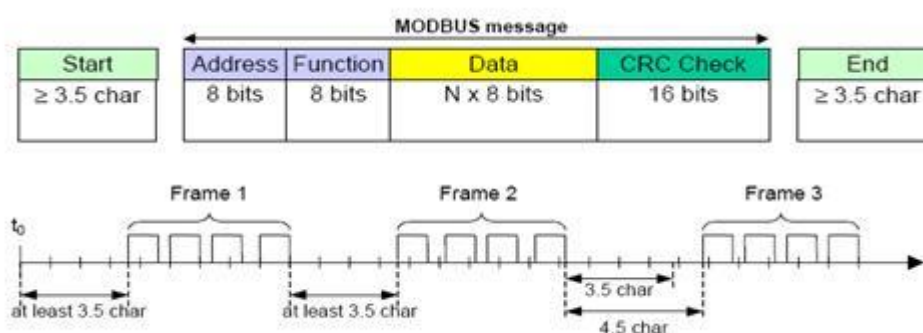
định là một giao thức hoạt động theo "hỏi/đáp" và sử dụng các "function codes" tương ứng để hỏi đáp.



Hình 2.30: Mô hình OSI

### 2.6.3 Cấu trúc bản tin Modbus RTU

Một bản tin Modbus RTU bao gồm: 1 byte địa chỉ - 1 byte mã hàm - n byte dữ liệu - 2 byte CRC như hình ở dưới:



Hình 2.31: Khung truyền dữ liệu

Chức năng và vai trò cụ thể như sau:

- **Byte địa chỉ:** xác định thiết bị mang địa chỉ được nhận dữ liệu (đối với Slave) hoặc dữ liệu nhận được từ địa chỉ nào (đối với Master). Địa chỉ này được quy định từ 0 - 254

- **Byte mã hàm:** được quy định từ Master, xác định yêu cầu dữ liệu từ thiết bị Slave. Ví dụ mã 01: đọc dữ liệu lưu trữ dạng Bit, 03: đọc dữ liệu tức thời dạng Byte, 05: ghi dữ liệu 1 bit vào Slave, 15: ghi dữ liệu nhiều bit vào Slave ...
- **Byte dữ liệu:** xác định dữ liệu trao đổi giữa Master và Slave.
  - Đọc dữ liệu:
    - Master: 2 byte địa chỉ dữ liệu - 2 byte độ dài dữ liệu
    - Slave: 2 byte địa chỉ dữ liệu - 2 byte độ dài dữ liệu - n byte dữ liệu đọc được
  - Ghi dữ liệu:
    - Master: 2 byte địa chỉ dữ liệu - 2 byte độ dài dữ liệu - n byte dữ liệu cần ghi
    - Slave: 2 byte địa chỉ dữ liệu - 2 byte độ dài dữ liệu
- **Byte CRC:** 2 byte kiểm tra lỗi của hàm truyền. cách tính giá trị của Byte CRC 16 Bit

### Sự khác nhau giữa Modbus RTU và Modbus TCP

Sự khác nhau cơ bản giữa MODBUS RTU và MODBUS TCP ( còn được gọi là modbus IP, Modbus Ethernet hay Modbus TCP/IP) là Modbus TCP chạy ở công vật lý Ethernet và Modbus RTU thì chạy ở công vật lý serial nối tiếp ( RS232 hoặc RS485).

### MODBUS RTU Memory Map

Modbus RTU Data Type	Common name	Starting address
Modbus Coils	Bits, binary values, flags	00001
Digital Inputs	Binary inputs	10001
Analog Inputs	Binary inputs	30001
Modbus Registers	Analog values, variables	40001

### Sự khác nhau giữa Modbus RTU và Modbus ASCII

Đây là hai chế độ truyền dữ liệu cơ bản trong giao thức Modbus khi sử dụng đường truyền nối tiếp là: ASCII và RTU. Mỗi một chuẩn sẽ có một cách mã hóa tin nhắn khác nhau, cho dù đều alf chuẩn Modbus chung. Ví dụ, Modbus ASCII cho phép người đọc có thể đọc trực tiếp tin nhắn trong bản tin. Nhưng với Modbus RTU, thì nội dung data đã được mã hóa nhị phân và không thể đọc được trong quá trình giám sát. Một điểm đặc biệt trong giao thức Modbus là trong một đường truyền dẫn Modbus, tất cả các giao thức phải là giống nhau, có nghĩa là Modbus ASCII không thể giao tiếp với modbus RTU và ngược lại.

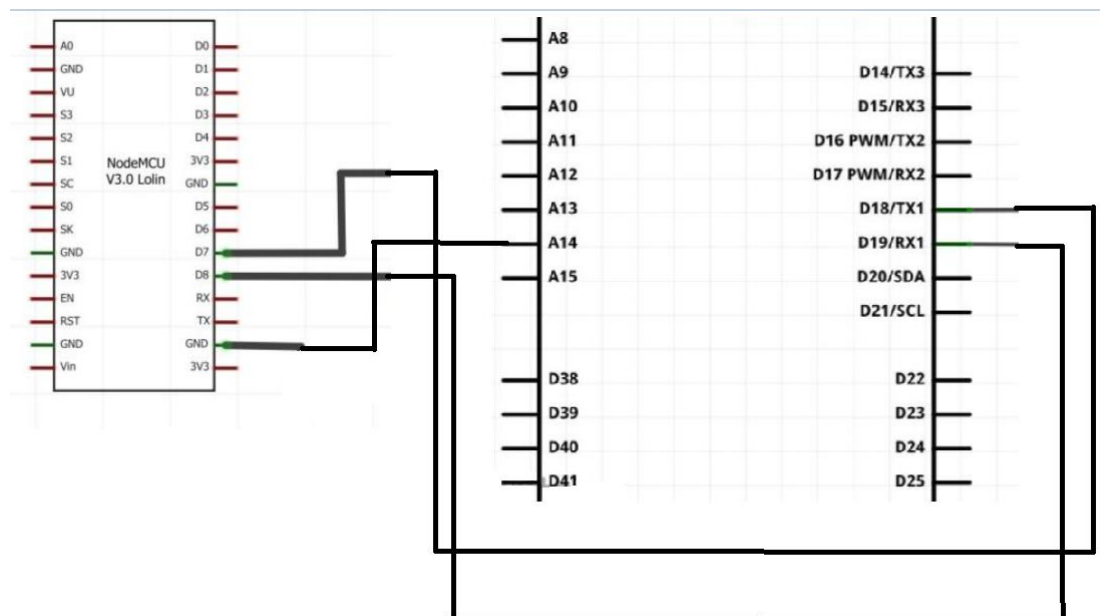
### Các tính chất của Modbus ASCII và Modbus RTU

	Modbus/ASCII		Modbus/RTU	
Characters	ASCII 0...9 and A..F		Binary 0...255	
Error check	LRC Longitudinal Redundancy Check		CRC Cyclic Redundancy Check	
Frame start	character ':'		3.5 chars silence	
Frame end	characters CR/LF		3.5 chars silence	
Gaps in message	1 sec		1.5 times char length	
Start bit	1		1	
Data bits	7		8	
Parity	even/odd	none	even/odd	none
Stop bits	1	2	1	2

Ở đây project của mình sẽ gồm:

Truyền theo kiểu modbus.

Phân nối dây





### Ở phần Node mcu slaves

```
#include <SimpleTimer.h>
#include <modbus.h>
#include <modbusDevice.h>
#include <modbusRegBank.h>
#include <modbusSlave.h>
#include <SoftwareSerial.h> // Modbus RTU pins  D7(13),D8(15)  RX,TX

modbusDevice regBank;
modbusSlave slave;
int s = 0;
int t = 0;

void setup()
{
    // You can also specify server:
    //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
    //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
    timer.setInterval(1000L, processes); //Real sensor in 1s
    //Assign the modbus device ID.
    regBank.setId(1);
    // Holding registers
    regBank.add(40005);
    regBank.add(40006);
    regBank.add(40007);
    regBank.add(40008);
    regBank.add(40009);
    regBank.add(40010);

    slave._device = &regBank;

    Serial.begin(9600);
    delay(100);
    // Initialize the serial port for coms at 9600 baud
    slave.setBaud(9600);
    delay(100);
    Serial.println("Connected ");
    Serial.println("Modbus RTU Slave Online");
}

void loop()
{
    //Goi du lieu cho mater

    SetAlarm = String(SGio)+ ":" +String(SPhut)+ ":" +String(SGiay);
    NotAlarm = String(FGio)+ ":" +String(FPhut)+ ":" +String(FGiay);
    // Set Alarm
    Serial.println("Set Alarm music: ");
```

```
Serial.print(SetAlarm);
regBank.set(40005,SetAlarm);
delay(20);

// Not Alarm
Serial.println("Not Alarm music: ");
Serial.print(NotAlarm);
regBank.set(40006,NotAlarm);
delay(20);

// Set volume music
Serial.println("Volume music: ");
Serial.print(vl);
regBank.set(40007,vl);
delay(20);

// Set control music
Serial.println("Control music: ");
Serial.print(control);
regBank.set(40008,control);
delay(20);

// Set Floder music
Serial.println("Floder music: ");
Serial.print(Floder);
regBank.set(40010,Floder);
delay(20);
slave.run();
timer.run();

// Set Song music
Serial.println("Song music: ");
Serial.print(song);
regBank.set(40010,song);
delay(20);
slave.run();
timer.run();
}
```

Mater Arduino lấy dữ liệu

```
#include <ModbusMaster232.h>
#include <SoftwareSerial.h> // Modbus RTU pins  D7(13),D8(15)  RX,TX
// Instantiate ModbusMaster object as slave ID 1
ModbusMaster232 node (1);
int Floder = 0;
int song = 0;
int control = 0;
int vl = 0;

int SetAlarm = 0;
int NotAlarm = 0;
void setup() {
  Serial.begin (9600);
  Delay (100);
  node.begin (9600); // Modbus RTU
  delay (1000);
  Serial.println ("Connected ");
  Serial.println ("Modbus RTU Master Online");
}

void loop(){

//Lay nhung gia tri tu slave goi ve
//Set Alarm
  Serial.println (" ");
  node.readHoldingRegisters (4, 1);
  Serial.print ("[4] Set Alarm music ");
  SetAlarm = node.getResponseBuffer (0);
  Serial.println (SetAlarm);
  node.clearResponseBuffer ();

//Not Alarm
  Serial.println (" ");
  node.readHoldingRegisters (5, 1);
  Serial.print ("[5] Not Alarm music ");
  NotAlarm = node.getResponseBuffer (0);
  Serial.println (NotAlarm);
  node.clearResponseBuffer ();

//Volume
  Serial.println (" ");
  node.readHoldingRegisters (6, 1);
  Serial.print ("[6] Volume music ");
  vl = node.getResponseBuffer(0);
  Serial.println (vl);
```

```
node.clearResponseBuffer ();

//Control
Serial.println (" ");
node.readHoldingRegisters (7, 1);
Serial.print ("[7] Control music ");
control = node.getResponseBuffer(0);
Serial.println (control);
node.clearResponseBuffer ();

//floder
Serial.println (" ");
node.readHoldingRegisters (8, 1);
Serial.print ("[8] floder music ");
Floder = node.getResponseBuffer (0);
Serial.println (Floder);
node.clearResponseBuffer ();

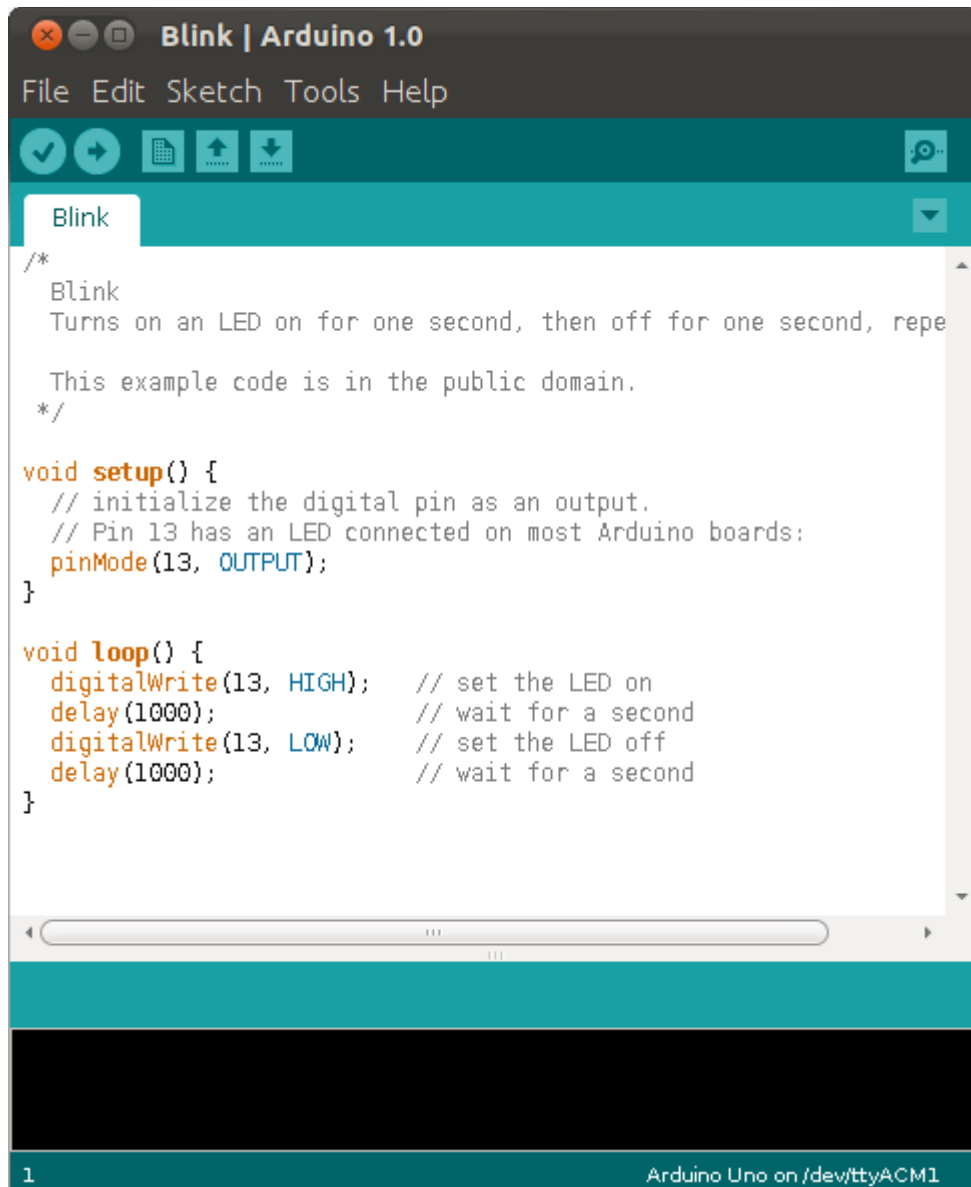
//Song
Serial.println (" ");
node.readHoldingRegisters (9, 1);
Serial.print ("[9] Song ");
song = node.getResponseBuffer(0);
Serial.println (song);
node.clearResponseBuffer ();
delay (100);

}
```

### CHƯƠNG III: THIẾT KẾ MÔ HÌNH PHẦN CỨNG VÀ PHẦN MỀM LẬP TRÌNH ỨNG DỤNG

#### 3.1 Thiết kế mô hình phần cứng và phần mềm lập trình cho hệ thống.

##### 3.1.1. Tìm hiểu phần mềm lập trình Arduino IDE:



**Hình 3.1: Phần mềm lập trình Arduino IDE**

Arduino IDE là một ứng dụng cross-platform (nền tảng) được viết bằng Java, và từ IDE này sẽ được sử dụng cho Ngôn ngữ lập trình xử lý (Processing programming language) và project Wiring. Nó được thiết kế để dành cho những người mới tập làm quen với lĩnh vực phát triển phần mềm. Nó bao gồm một chương trình code editor với

### CHƯƠNG III: THIẾT KẾ MÔ HÌNH PHẦN CỨNG VÀ PHẦN MỀM LẬP TRÌNH ỨNG DỤNG

---

các chức năng như đánh dấu cú pháp, tự động brace matching, và tự động canh lề, cũng như compile (biên dịch) và upload chương trình lên board chỉ với 1 cú nhấp chuột. Một chương trình hoặc code viết cho Arduino được gọi là một *sketch*.

Các chương trình Arduino được viết bằng C hoặc C++. Arduino IDE đi kèm với một thư viện phần mềm được gọi là "Wiring", từ project Wiring gốc, có thể giúp các thao tác input/output được dễ dàng hơn. Người dùng chỉ cần định nghĩa 2 hàm để tạo ra một chương trình vòng thực thi (cyclic executive) có thể chạy được:

- `setup()`: hàm này chạy mỗi khi khởi động một chương trình, dùng để thiết lập các cài đặt
- `loop()`: hàm này được gọi lặp lại cho đến khi tắt nguồn board mạch

Một chương trình điển hình cho một bộ vi điều khiển đơn giản chỉ là làm cho một bóng đèn Led sáng/tắt. Trong môi trường Arduino, ta sẽ phải viết một chương trình giống như sau:

```
#define LED_PIN 13

void setup () {
    pinMode (LED_PIN, OUTPUT); // Đặt chân 13 làm đầu ra
    digital
}

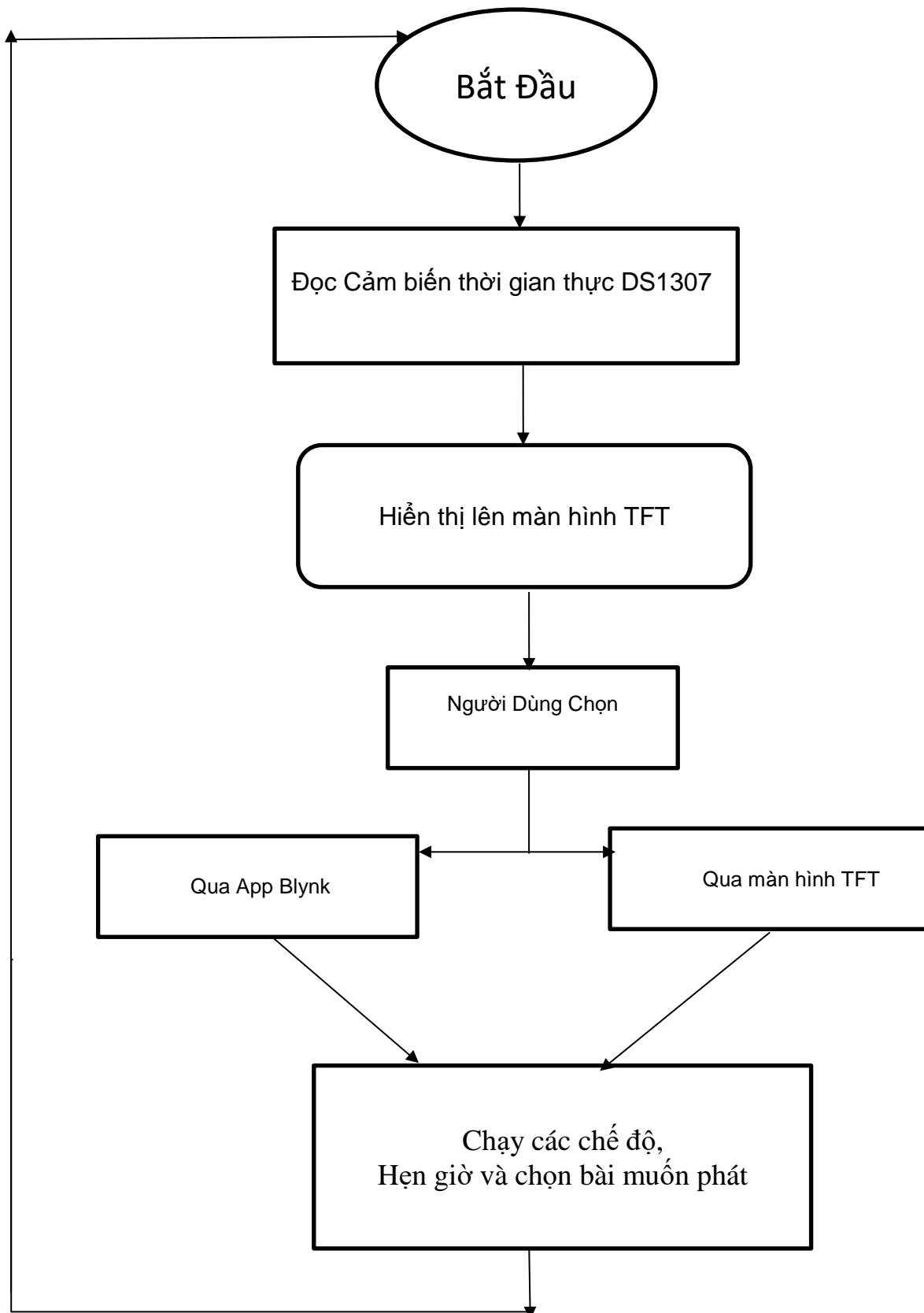
void loop () {
    digitalWrite (LED_PIN, HIGH); // Bật LED on
    delay (1000); // chờ trong 1 giây (1000 mili giây)
    digitalWrite (LED_PIN, LOW); // Tắt LED off
    delay (1000); // chờ trong 1s
}
```

Arduino IDE này sử dụng GNU toolchain và AVR Libc để biên dịch chương trình, và sử dụng avrdude để upload chương trình lên board.

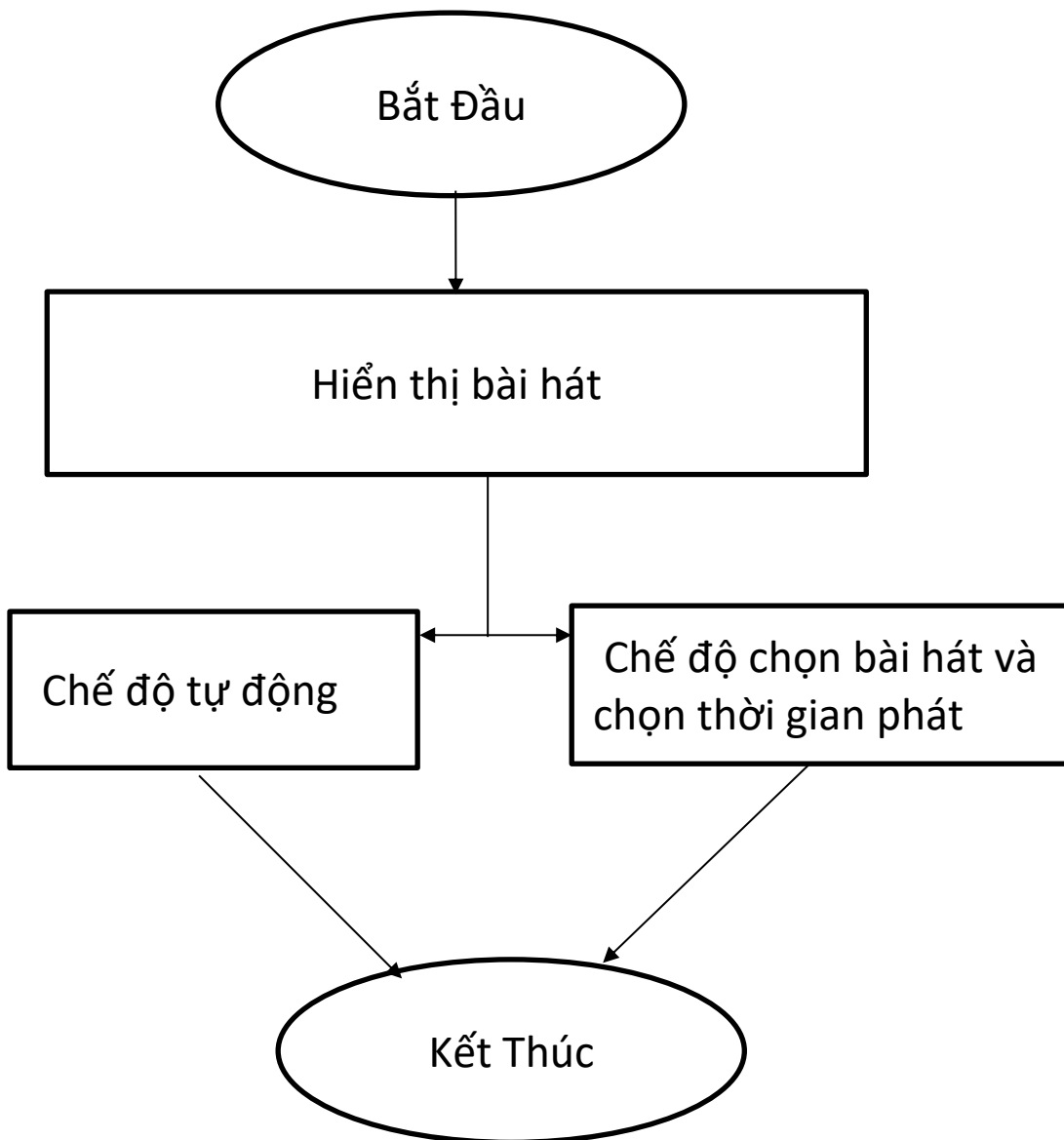




#### 3.1.3 Sơ đồ giải thuật:



App Blynk



### CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sau nhiều tháng thực hiện nỗ lực và cố gắng cùng với sự hướng dẫn tận tình của cô ThS. Nguyễn Lan Anh, tập nghiên cứu đã hoàn thành đúng thời hạn theo quy định, theo yêu cầu đặt ra nghiên cứu ứng dụng mạch hẹn giờ phát thông tin cần phát:

Tìm hiểu nguyên lý mạch giải mã phát nhạc và các mô hình điều khiển bằng wifi.

Tìm hiểu các thiết bị linh kiện sử dụng trong hệ thống như: Module wifi ESP8266, vi điều khiển, các module DFplayer, DS1307...

Tìm hiểu các phần mềm thiết kế và thực hiện hệ thống: Phần mềm tạo APP, phần mềm lập trình vi điều khiển...

Thực hiện thiết kế mạch hẹn giờ phát thông tin.

#### **Hướng phát triển của đề tài:**

- Thiết kế mạch nhỏ gọn hơn có thể làm đồng hồ đeo tay.
- Hạn chế sử dụng Module mà sử dụng chip để tiết kiệm chi phí.
- Kết hợp sử dụng module điều khiển bằng giọng nói.
- Phát triển thêm màn hình OLED rực rỡ không hạn chế góc nhìn, radio FM tích hợp, chức năng ghi âm cùng với ngõ vào âm thanh phụ, duyệt ảnh, xem phim.
- Như máy MP3 này có các tính năng như: Radio FM, ghi âm giọng nói, đồng hồ báo thức, hỗ trợ nhiều định dạng âm thanh khác nhau, đọc văn bản. Ngoài ra, còn nên được trang bị hiệu ứng âm thanh SRS Wow, thời gian sử dụng pin phải lâu.

Một lần nữa em chân thành cảm ơn ThS. Nguyễn Lan Anh đã tận tình giúp đỡ, chỉ bảo và đưa hướng đi đúng đắn cho em trong thời gian qua, giúp em có thể hoàn thành tốt những nhiệm vụ đã đặt ra.

Trong quá trình thực hiện nghiên cứu không tránh khỏi những sai sót, em rất mong nhận được sự chia sẻ, chỉ bảo của thầy cô và bạn bè. Em xin chân thành cảm ơn.

### TÀI LIỆU THAM KHẢO

[1] <https://www.espressif.com/en/products/hardware/esp8266ex/overview>

[2] <https://hocarm.org/>

[3] [\*Ds1307 Datasheet - AllDataSheet.com\*](#)

[4] <http://www.picaxe.com/docs/spe033.pdf>

[5] <https://www.cisco.com>

Các trang web tham khảo:

[6] <https://www.dfrobot.com/>

[7] <https://www.codeproject.com>

[8] <https://www.blynk.cc/>

[9] <https://howtomechatronics.com/>

[10] <https://www.instructables.com/howto/arduino/>

[11] <https://github.com/>