# **Lời Cảm Ơn**

# **Mục Lục**

# **Lời Nói Đầu**

# **Chương 1: Giới thiệu chung**

* 1. **Lý do chọn đề tài**

Hàng tháng, chúng ta phải luôn trả các hóa đơn điện – nước mà hầu như số tiền phải đóng lại ngày một tăng cao. Lý do ở đây là ta không thể hoàn toàn kiểm soát được mức điện – nước đã được sử dụng, bởi hầu như ta không có bất kỳ con số thống kê cụ thể nào cả, ngoài việc tự ước lượng. Hiện nay, trên thị trường cũng có bán các thiết bị để giám sát điện năng tiêu thụ với độ chính xác khá cao, nhưng giá thành thì lại không hề rẻ, cũng như hạn chế về mặt giám sát từ xa. Nhận thấy được điều này, nhóm chúng em muốn tạo ra một ứng dụng giúp cho các hộ gia đình có thể dễ dàng thống kê - giám sát được lượng điện - nước mà họ sử dụng hàng ngày; để từ đó họ có thể kiểm soát và đề ra phương án sử dụng một cách hiệu quả và tiết kiệm hơn. Đó là lý do em quyết định lựa chọn và thực hiện đề tài “**Xây Dựng Hệ Thống Giám Sát Tiêu Thụ Điện Năng Và Nước Sinh Hoạt Bằng Công Nghệ IOT**”.

**1.2. MỤC TIÊU**

Thiết kế một hệ thống tiến hành đo lượng điện – nước tiêu thụ, và đều đặn cập nhật các thông số đó lên một App để thuận tiện cho công việc giám sát. Hệ thống ứng dụng công nghệ IoT, giúp cho người dùng ở bất kỳ đâu cũng có thể dễ dàng theo dõi được. Đồng thời, ứng dụng cũng xây dựng một hệ thống các user, giúp cho quản trị viên dễ dàng hơn trong việc kiểm soát thông tin người dùng.

**1.3 NỘI DUNG NGHIÊN CỨU**

• NỘI DUNG 1: Tìm hiểu và lựa chọn các giải pháp thiết kế.

• NỘI DUNG 2: Thu thập tài liệu về các cảm biến, module wifi, bộ vi xử lý, cũng như 1 App khả dụng.

• NỘI DUNG 3: Thiết kế, lập trình cho hệ thống điều khiển, chạy thử nghiệm.

• NỘI DUNG 4: Thiết kế mô hình, chỉnh sửa và cải tiến từ những phương án đã chọn • NỘI DUNG 5: Đánh giá kết quả thực hiện

**1.4. GIỚI HẠN**

• Hệ thống chỉ dừng lại ở công việc giám sát các thông số

• Mô hình không quá to để có thể dễ dàng sử dụng ở nhà

• Sử dụng nguồn điện lấy trực tiếp từ lưới điện gia đình.

• Sai số hệ thống chấp nhận được 1.5.

# **Chương 2: Phân tích bài toán**

## **2.1. Lựa chọn công nghệ kết nối**

Trong hệ thống IOT, có rất nhiều loại công nghệ kết nối được sử dụng, có thể kể tới như: Zigbee, Lora, Bluetooth, … và đặc biệt phải kể tới đó là công nghệ kết nối wifi.

Wifi là viết tắt của **Wireless Fidelity**. Nhiệm vụ của wifi này là sử dụng sóng vô tuyến để truyền tín hiệu đến các thiết bị kết nối mà không kết đến đây. Loại sóng vô tuyến này giống như sóng điện thoại, radio,.. nhưng đường truyền sóng ngắn hơn. Đặc biệt là tất cả các thiết bị thông minh như máy tính bảng, smartphone, laptop đều có khả năng kết nối wifi, nên rất thuận tiện trong quá trình sử dụng.

Wifi sử dụng sóng radio (sóng vô tuyến) để truyền thông tin qua hệ thống mạng. Máy tính của bạn bao gồm một card mạng không dây sẽ truyền dữ liệu gửi vào tín hiệu radio. Tương tự tín hiệu này sẽ được truyền đi thông qua một ăng-ten, một bộ giải mã gọi là router. Sau khi giải mã xong, dữ liệu sẽ được gửi đến Internet thông qua một kết nối Ethernet có dây. Khi mạng không dây hoạt động như đường 2 chiều, các dữ liệu nhận được từ internet cũng sẽ đi qua router và được mã hoá thành tín hiệu radio để card mạng không dây trên máy tính nhận.



*Hình 1. 2. Hoạt động của WiFi*

Wifi được sử dụng rất phổ biến trong các hệ thống IOT bởi nó cho phép thiết bị có khả năng kết nối không dây tới internet một cách thuận tiện bằng việc tận dụng sẵn cơ sở hạ tầng mạng sẵn có và phổ biến từ trước, thay vì phải xây dựng, lắp đặt mới như các chuẩn kết nối khác.

## **2.2. Lựa chọn linh kiện sử dụng**

Với yêu cầu của đề tài đặt ra, cần xác định các tính năng của bài toán bao gồm:

* Đo lượng điện tiêu thụ
* Đo lượng nước tiêu thụ
* Gửi số liệu đo đạt tới người dùng, thiết bị theo dõi.

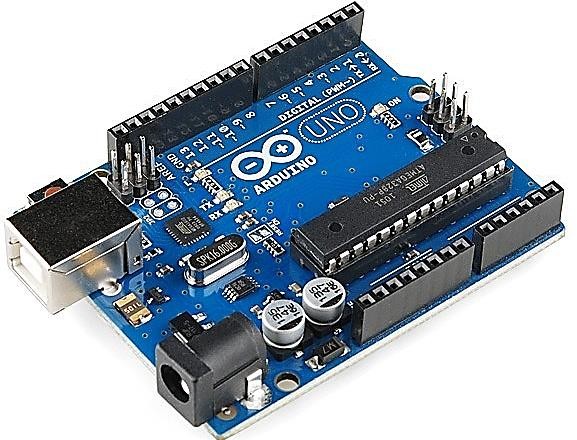
Từ những tiêu chí trên, cần lựa chọn vi điều khiển, cảm biến đo điện năng tiêu thụ và cảm biến đo lưu lượng nước cùng với chip wifi để giúp thiết bị kết nối tới internet. Dưới đây là những linh kiện được sử dụng trong hệ thống:

* Arduino Uno R3
* Esp8266
* Cảm biến Pzem004t-v3
* Cảm biến lưu lượng nước YF-S201 DN15

### **2.2.1 Arduino Uno R3**

a. Khối điều khiển trung tâm Arduino UNO R3

Arduino một nền tảng [mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phần cứng và phần mềm. Phần cứng Arduino (các board mạch vi xử lý) được sinh ra tại thị trấn Ivrea ở Ý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau.



*Hình 1: Arduino UNO R3*

b*.* Thông số kỹ thuật

|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 họ 8bit |
| Điện áp hoạt động | 5V DC (chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | khoảng 30mA |
| Điện áp vào khuyên dùng | 7-12V DC |
| Điện áp vào giới hạn | 6-20V DC |
| Số chân Digital I/O | 14 (6 chân hardware PWM) |
| Số chân Analog | 6 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa (3.3V) | 50 mA |
| Bộ nhớ flash | 32 KB (ATmega328) với 0.5KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1KB (ATmega328) |

**c. Cấp nguồn và các chân nguồn**

**Cấp nguồn**

Arduino UNO có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyên dùng là 7-12V DC và giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lí nhất nếu bạn không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, bạn sẽ làm hỏng Arduino UNO.

Các chân nguồn

**GND (Ground):** cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.

**5V:** cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.

**3.3V**: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.

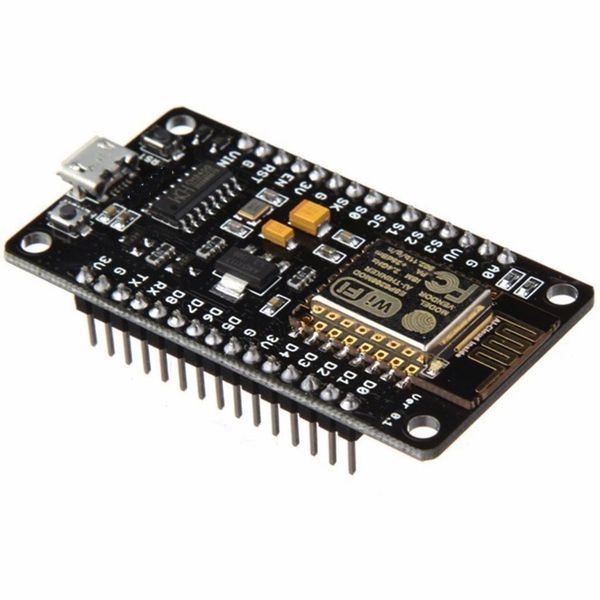
**Vin (Voltage Input)**: để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.

**IOREF**: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.

**RESET**: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ

### **2.2.2 Kit RF Thu Phát Wifi ESP8266 Node MCU**

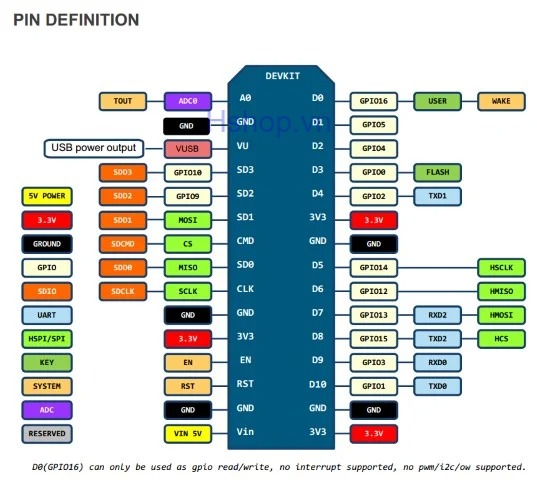
Kit RF thu phát Wifi ESP8266 NodeMCU Lua CP2102 là kit phát triển dựa trên nền chip Wifi SoC ESP8266 với thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 trở nên rất đơn giản.  
  
Kit RF thu phát Wifi ESP8266 NodeMCU Lua CP2102 được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT.



**Hình 1.1** Kit RF Thu Phát Wifi ESP8266 NodeMCU Lua CP2102

**Các thông số kỹ thuật của module ACS712 -20A:**

* IC chính: ESP8266
* Phiên bản firmware: NodeMCU Lua
* Chip nạp và giao tiếp UART: CP2102.
* GPIO tương thích hoàn toàn với firmware Node MCU.
* Cấp nguồn: 5VDC MicroUSB hoặc Vin.
* GIPO giao tiếp mức 3.3VDC
* Tích hợp Led báo trạng thái, nút Reset, Flash.
* Tương thích hoàn toàn với trình biên dịch Arduino.
* Kích thước: 25 x 50 mm



Hình 4 Sơ đồ chân ESP8266 Node MCU

### **2.2.3 Cảm biến lưu lượng nước YF-S201 DN15**

Ta có thể sử dụng đồng hồ đo nước thông dụng để biết được lượng nước sinh hoạt đã sử dụng. Tuy nhiên, để có thể giám sát ở bất cứ đâu thông qua internet ứng dụng công nghệ IoT, ta cần phải sử dụng một cảm biến chuyên dụng, để đọc và gửi dữ liệu từ cảm biến về vi bộ xử lý. Trong đề này sử dụng cảm biến lưu lượng S201 để đo.

Cảm biến S201 bên trong có chứa một cánh quạt để đếm lượng chất lỏng chảy qua nó và có một cảm biến từ Hall xuất ra các xung khi có sự thay đổi trạng thái đầu ra. Cảm biến Hall được hàn kín trong ống để được an toàn và khô ráo.



**Hình 2.3** Hình ảnh của cảm biến lưu lượng S201

**Cảm biến lưu lượng có 3 dây:**

* Dây đỏ: cấp nguồn 5V – 24VDC
* Dây đen: GND
* Dây vàng: ngõ ra của cảm biến Hall

**Thông số kỹ thuật của cảm biến S201:**

* Điện áp làm việc: 5V – 24VDC
* Loại ngõ ra: 5V TTL
* Dòng điện cao nhất: 15mA (5V)
* Mức độ dòng chảy: từ 1 đến 30 L/phút
* Vận hành ở nhiệt độ: -25ºC – 80ºC
* Nhiệt độ dòng chảy: <120ºC
* Áp lực nước tối đa: 2Mpa
* Vận hành ở độ ẩm: 35% - 80% RH
* Sai số: 10%
* Số xung trên lít: 450

Với các tín hiệu xung ra là một dải xung vuông đơn giản, ta có thể dễ dàng đọc và tính được lưu lượng nước bằng việc đếm xung từ ngõ ra của cảm biến theo công thức: *Tần số xung (Hz) / 7,5 = tốc độ dòng chảy (L / phút)*

**Một số lưu ý khi sử dụng:**

* Nên đặt cảm biến ở trên cùng dòng chảy
* Không cho dòng chảy có chất hóa học, ăn mòn
* Không chịu va đập khi sử dụng
* Đặt cảm biến thẳng đứng không lệch quá 5 độ
* Nhiệt độ nước chảy qua dưới 120 độ C

### **2.2.4 Cảm biến Pzem004t-v3**

Mạch đo điện AC đa năng 100A giao tiếp UART PZEM-004T được sử dụng để đo và theo dõi gần như hoàn toàn các thông số về điện năng AC của mạch điện như điện áp hoạt động, dòng tiêu thụ, công suất và năng lượng tiêu thụ, mạch sử dụng giao tiếp UART dễ dàng kết nối truyền dữ liệu tới Vi điều kiển hoặc máy tính, thích hợp cho các ứng dụng theo dõi năng lượng, IoT…

Mạch đo điện AC đa năng 100A giao tiếp UART PZEM-004T nhỏ gọn, dễ lắp đặt, sử dụng cách đo dòng cách ly an toàn và khả năng đo dòng lên đến 100A, mạch có chất lượng gia công và linh kiện tốt, độ bền cao.

**Thông số kỹ thuật:**

* Điện áp đo và hoạt động: 80 ~ 260VAC / 50 - 60Hz, sai số 0.01
* Dòng điện đo và hoạt động: 0 ~ 100A, sai số 0.01
* Công suất đo và hoạt động: 0 ~ 26000W
* Năng lượng đo và hoạt động: 0~9999kWh.
* Giao tiếp UART mức logic TTL 5VDC baudrate mặc định 9600, 8, 1.
* Có opto cách ly an toàn giữa mạch đo và mạch nhận tín hiệu UART.
* Lưu giữ thông số năng lượng tiêu thụ trong bộ nhớ.
* Kích thước: 30 x 75 mm.



Hình 5: Mạch đo điện AC đa năng.

## **2.3** **Giao thức MQTT**

***1.4.1. Định nghĩa***

MQTT (Message Queuing Telemetry Transport) là giao thức truyền thông điệp (message) theo mô hình publish/subscribe (cung cấp / thuê bao), được sử dụng cho các thiết bị IoT với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Nó dựa trên một Broker (tạm dịch là “Máy chủ môi giới”) “nhẹ” (khá ít xử lý) và được thiết kế có tính mở (tức là không đặc trưng cho ứng dụng cụ thể nào), đơn giản và dễ cài đặt.

### ***1.4.2. Tính năng, đặc điểm nổi bật***

Dạng truyền thông điệp theo mô hình Pub/Sub cung cấp việc truyền tin phân tán một chiều, tách biệt với phần ứng dụng.

Việc truyền thông điệp là ngay lập tức, không quan tâm đến nội dung được truyền.

Sử dụng TCP/IP là giao thức nền.

Tồn tại ba mức độ tin cậy cho việc truyền dữ liệu (QoS: Quality of service)

* QoS 0: Broker/client sẽ gửi dữ liệu đúng một lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP.
* QoS 1: Broker/client sẽ gửi dữ liệu với ít nhất một lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
* QoS 2: Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng một lần, quá trình này phải trải qua 4 bước bắt tay.

Phần bao bọc dữ liệu truyền nhỏ và được giảm đến mức tối thiểu để giảm tải cho đường truyền.

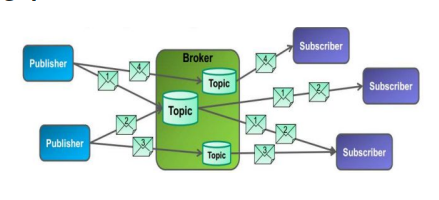
***1.4.3. Ưu điểm của MQTT***

Với những tính năng, đặc điểm nổi bật trên, MQTT mang lại nhiều lợi ích nhất là trong hệ thống SCADA (Supervisory Control And Data Acquisition) khi truy cập dữ liệu IoT.

* Truyền thông tin hiệu quả hơn.
* Tăng khả năng mở rộng.
* Giảm đáng kể tiêu thụ băng thông mạng.
* Rất phù hợp cho điều khiển và do thám.

………..

***1.4.4. Cơ chế hoạt động của MQTT theo mô hình Pub/Sub***

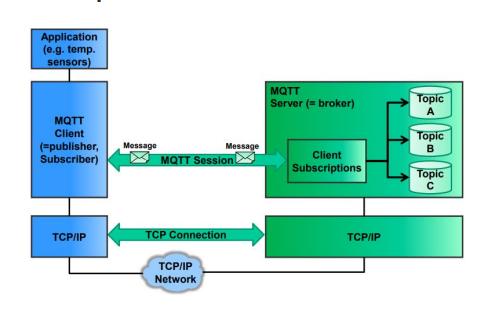


*Hình 1. 3 Cơ chế tổng quan*

MQTT hoạt động theo cơ chế client/server, nơi mà mỗi cảm biến là một khách hàng (client) và kết nối đến một máy chủ, có thể hiểu như một Máy chủ môi giới (broker), thông qua giao thức TCP (Transmission Control Protocol). Broker chịu trách nhiệm điều phối tất cả các thông điệp giữa phía gửi đến đúng phía nhận.

MQTT là giao thức định hướng bản tin. Mỗi bản tin là một đoạn rời rạc của tín hiệu và broker không thể nhìn thấy. Mỗi bản tin được publish một địa chỉ, có thể hiểu như một kênh (Topic). Client đăng kí vào một vài kênh để nhận/gửi dữ liệu, gọi là subscribe. Client có thể subscribe vào nhiều kênh. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu vào kênh đã đăng ký. Khi một client gửi một bản tin đến một kênh nào đó gọi là publish.

***1.4.5. Các thành phần chính của MQTT***



*Hình 1. 4 Kiến trúc thành phần*

Thành phần chính của MQTT là Client (Publisher/Subscriber), Server (Broker), Sessions (tạm dịch là Phiên làm việc), Subscriptions và Topics.

MQTT Client (Publisher/Subscriber): Clients sẽ subscribe một hoặc nhiều topics để gửi và nhận thông điệp từ những topic tương ứng.

MQTT Server (Broker): Broker nhận những thông tin subscribe (Subscriptions) từ client, nhận thông điệp, chuyển những thông điệp đến các Subscriber tương ứng dựa trên Subscriptions từ client.

Topic: Có thể coi Topic là một hàng đợi các thông điệp, và có sẵn khuôn mẫu dành cho Subscriber hoặc Publisher. Một cách logic thì các topic cho phép Client trao đổi thông tin với những ngữ nghĩa đã được định nghĩa sẵn. Ví dụ: Dữ liệu cảm biến nhiệt độ của một tòa nhà.

Session: Một session được định nghĩa là kết nối từ client đến server. Tất cả các giao tiếp giữa client và server đều là 1 phần của session.

Subscription: Không giống như session, subscription về mặt logic là kết nối từ client đến topic. Khi đã subscribe một topic, Client có thể nhận/gửi thông điệp (message) với topic đó.

## **2.4 React Native**

**React Native** là một framework được tạo bởi Facebook, cho phép các lập trình viên sử dụng JavaScript để làm mobile apps trên cả Android và iOS với có trải nghiệm và hiệu năng như native. React Native vượt trội ở chỗ chỉ cần viết một lần là có thể build ứng dụng cho cả IOS lẫn Android. Việc này giúp chúng ta có thể tiết kiệm được thời gian, công sức, tiền bạc. Giúp tốc độ ra sản phẩm cũng như cập nhật ứng dụng nhanh chóng mặt. Có thể nói React Native là một cross-platform để xây dựng một ứng dụng di động hiệu quả.

### **2.4.1 Những ưu điểm của React Native**

React Native cùng với [Flutter](https://topdev.vn/blog/flutter-la-gi/) đang là xu hướng lập trình di động hiện nay bởi tính đa nền tảng cũng như tiết kiệm thời gian triển khai dự án. Sau đây là những lợi ích mà nó đem lại:

#### **Thời gian học ngắn hơn**

Một lý do lập trình mobile app rất khó và tốn thời gian là vì thực tế bạn cần tìm hiểu 2 hệ sinh thái hoàn toàn khác biệt. Nếu bạn muốn lập trình app iOS, bạn phải học Swift hoặc Objective-C và Cocoa Pods.

Nếu muốn lập trình app Android, bạn cần học Java hoặc Kotlin và Android SDK. Tôi từng viết code với 3 ngôn ngữ là Swift, Objective C, Java và không thực sự hứng thú với việc tranh luận ngôn ngữ nào tốt hơn.

Nếu bạn chọn lập trình trên React Native, phần lớn thời gian bạn sẽ chỉ cần học 1 bộ công cụ. Có rất nhiều thứ để bạn làm quen như: JavaScript, Node, React Native… nhưng chỉ có 1 công cụ duy nhất để học.

#### **Khả năng tái sử dụng code**

Khả năng sử dụng lại code đóng vai trò quan trọng trong lập trình phần mềm, nên mỗi khi bạn có thể sử dụng lại code thì React Native là công cụ tốt.

Tuy nhiên, sẽ luôn có vài UI code thông dụng có thể được chia sẻ chung với nhau cùng tất cả logic. Tính năng “có thể chia sẻ code” có rất nhiều lợi điểm như: tận dụng nguồn nhân lực tốt hơn, duy trì ít code hơn, ít bugs hơn, các tính năng trong cả 2 platforms cũng tương tự nhau…

#### **Cộng đồng lớn**

React Native đang trở lên rất phổ biến, nhiều developer đang đóng góp để làm React Native tốt hơn. Đặc biệt là nó được tạo ra và hỗ trợ bởi tập đoàn Facebook. React Native Github repro là một nguồn mở và có hàng nghìn cộng tác viên hoạt động rất năng nổ. Cộng đồng rất lớn và đang phát triển mạnh mẽ. Nhiều vấn đề đã và đang được giải quyết và bạn sẽ không cần phải tốn thời gian để nghiên cứu lại trong suốt quá trình học và làm việc với React Native.

### **2.4.2 Những nhược điểm của React Native**

React Native là một giải pháp tuyệt vời cho phát triển ứng dụng trên điện thoại di động, tuy nhiên đến thời điểm hiện tại, vẫn còn tồn tại một số khuyết điểm:

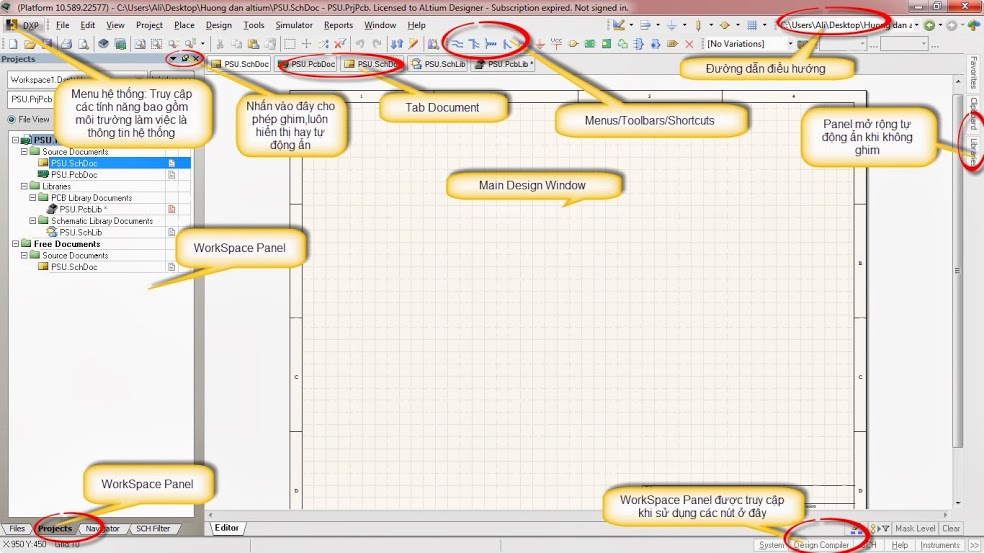
1. Vẫn còn thiếu các component quan trọng nhưng dần dần cũng đang có thêm nhiều cập nhật mới kể từ bài viết này mà tôi chưa được biết.
2. Không build được ứng dụng iOS trên Window và Linux: do yêu cầu từ Apple, mọi ứng dụng iOS cần được sử dụng nhiều native libs, cert…từ Xcode.
3. React Native không thể build được ứng dụng “quá phức tạp” nếu bạn không biết Swift/Objecive-C, Java – tính phức tạp ở đây là ứng dụng của bạn cần phải chỉnh sửa các component. Để viết được 1 ứng dụng native bằng javascript thì “luôn luôn” có sẵn các component đã được viết từ Swift/Objective-C (iOS) và Java (Android) để bạn sử dụng. Trường hợp bạn muốn chỉnh sửa 1 component nào đó: thay đổi thành phần hoặc thêm API thì bạn phải tự viết bằng chính ngôn ngữ tương ứng của iOS hoặc Android. Nhờ cộng đồng lớn nên cũng có nhiều lập trình viên khác đã viết nhiều component cần thiết cho hầu hết ứng dụng (đây cũng là lý do vì sao Facebook biến React Native thành mã nguồn mở).
4. Không dùng để viết game có tính đồ họa và cách chơi phức tạp.
5. Dùng ES2015/ES6 nên đây là cấu trúc mới cho Javascript từ 2015, vì khá là mới nên những cấu trúc của nó có thể bạn chưa quen, dẫn tới việc khó khăn trong việc tiếp cận.

# **Chương 3: Thiết kế hệ thống**

## **3.1 Công cụ sử dụng**

### 3.**1.1 Altium**

Altium Designer là một trong những phần mềm tự động hóa thiết kế PCB và điện tử cho các bo mạch in tốt và phổ biến nhất hiện nay. Nó được phát triển bởi công ty phần mềm Altium Limited có trụ sở tại Úc.



**Hình 3.6 Giao diện phần mềm Altium**

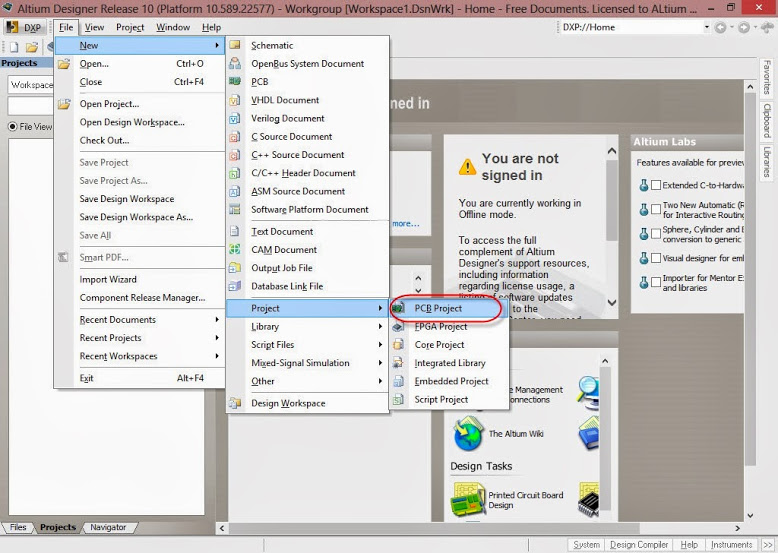
* Altium cung cấp một môi trường làm việc với số lượng phím tắt tương đối lớn và hoàn chỉnh. Để ý các phím tắt là cái kí tự có gạch chân trong các Menu Bar. Tận dụng điều đó giúp thiết kế nhanh và chuyên nghiệp hơn.
* Chú ý một số khái niệm đã nêu trong bài tổng quan về mạch in và các tiêu chuẩn khi thiết kế mạch in.
* Để làm quen dần với phím tắt, các phím tắt được trình bày sau mỗi thao tác, xuất hiện trong dấu ngoặc đơn. Tập sử dụng các phím tắt và ghi nhớ, các mạch sau này thực hiện dễ dàng và thao tác nhanh hơn rất nhiều.
* Mỗi linh kiện sẽ có hình dáng và kích thước riêng, tùy loại linh kiện mà cấu tạo Footprint sẽ khác nhau, ví dụ linh kiện hàn xuyên mạch vẽ trên Multiplayer để hiện đầy đủ các lớp, kích thước lỗ khoan. Đối với linh kiện dán thì xuất hiện lớp Top Paster, được thực hiện trên một mặt của mạch.
* Các layer:
  + Top layer: Lớp đường mạch ở mặt trên.
  + Bottom layer: Lớp đường mạch ở mặt dưới.
  + Top overlay: Hiển thị thông tin linh kiện ở mặt trên.
  + Bottom overlay: Hiển thị thông tin linh kiện ở mặt dưới.
  + Top paste: được tạo ra bởi các chân của linh kiện dán ở mặt trên.
  + Bottom paste: được tạo ra bởi các chân của linh kiện dán ở mặt dưới.
  + Top solder: Lớp Solder Mask ở mặt trên, xem thêm trong bài tổng quan về mạch in.
  + Bottom solder: Lớp Solder Mask ở mặt dưới, xem thêm trong bài tổng quan về mạch in.
  + Keep out layer: đường giới hạn board, board được cắt theo nhưng đường vẽ trên lớp này.

***Khởi tạo Project và tùy chỉnh giao diện làm việc***

Để thiết kế mạch in với Altium cần khởi tạo các cơ bản như sau:

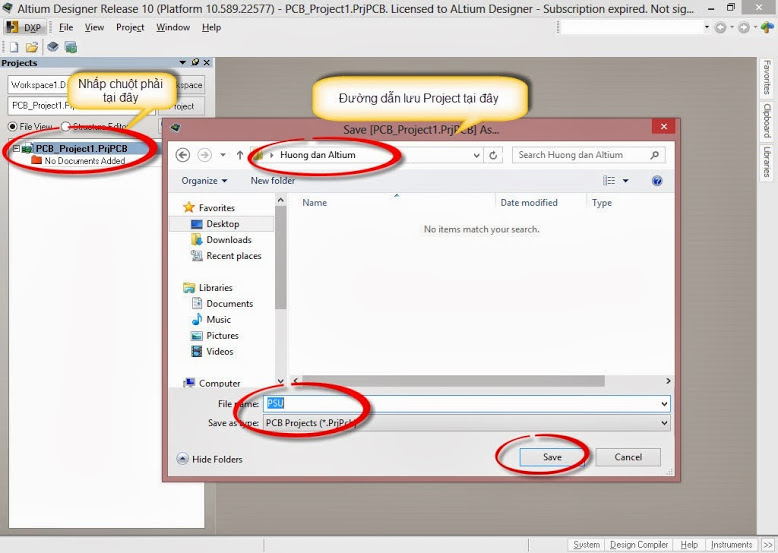
Khởi tạo một dự án PCB trong đó cơ bản bao gồm:

* Schematic: Sơ đồ mạch nguyên lý
* PCB: Sơ đồ mạch in
* Schematic Library: Thư viện dùng cho mạch nguyên lý
* PCB Library: Thư viện dùng cho mạch in.
* Tạo project mới
  + Từ giao diện chính của chương trình Chọn File->New->Project->PCB Project.

****

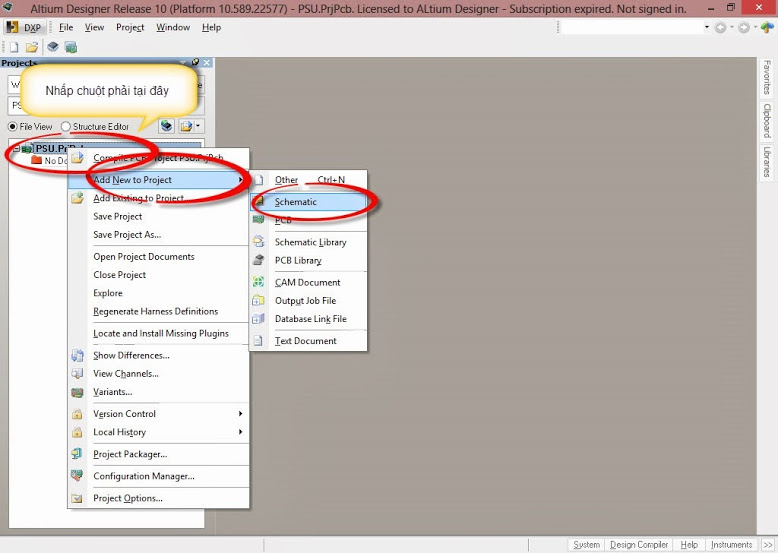
**Hình 3.7 Tạo project mới trong Altium**

* + Trên Workspace nhấn chuột phải vào PCB\_Project1 đã tạo chọn Save Project As…
  + Tại đây chọn thư mục để lưu Project, ở đây lưu với mạch tên mà bận muốn. Lưu ý việc lưu tên và đường dẫn project sẽ thuận tiện làm việc sau này.



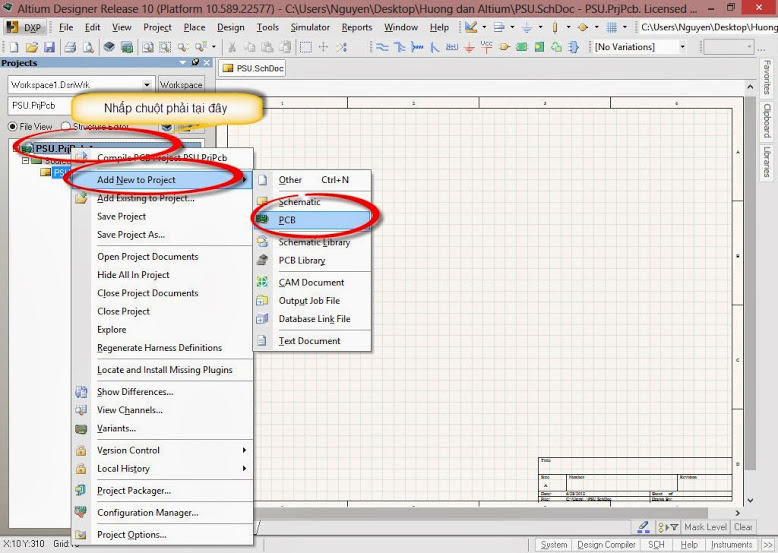
**Hình 3.8 Lưu project mới tạo**

* ***Tạo mới 1 schematic:***
  + Nhấp chuột phải trong giao diện Workspace chọn Add New to Project -> Schematic.



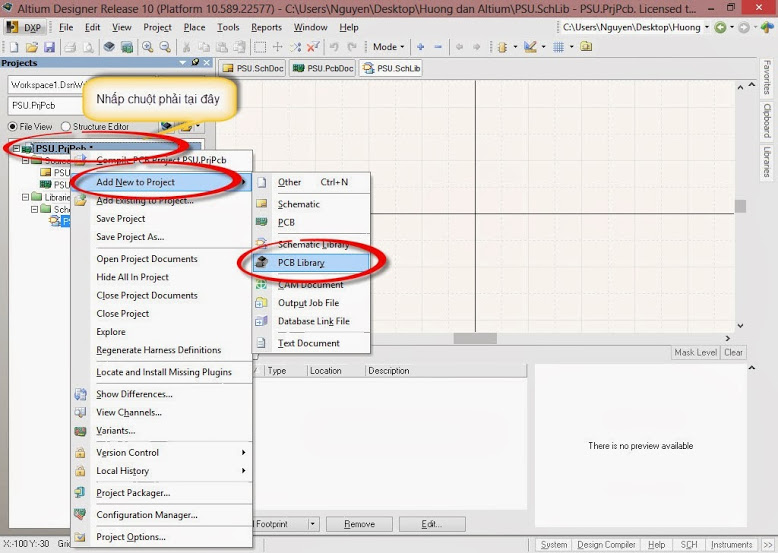
**Hình 3.9 Tạo file schematic**

* Sau đó lưu lại file Schematic tương tự như Project, lưu trong cùng một thư mục để dễ quản lý. Nhấp chuột phải vào Sheet1.SchDoc trong giao diện Work Space chọn Save As, lưu lại.
* Lưu ý rằng nếu bạn không lưu để dễ quản lý thì sau này khó làm việc.
* Tạo mới 1 PCB (sơ đồ thực hiện trên mạch in).
  + Nhấp chuột phải trong giao diện Workspace chọn Add New to Project -> PCB.



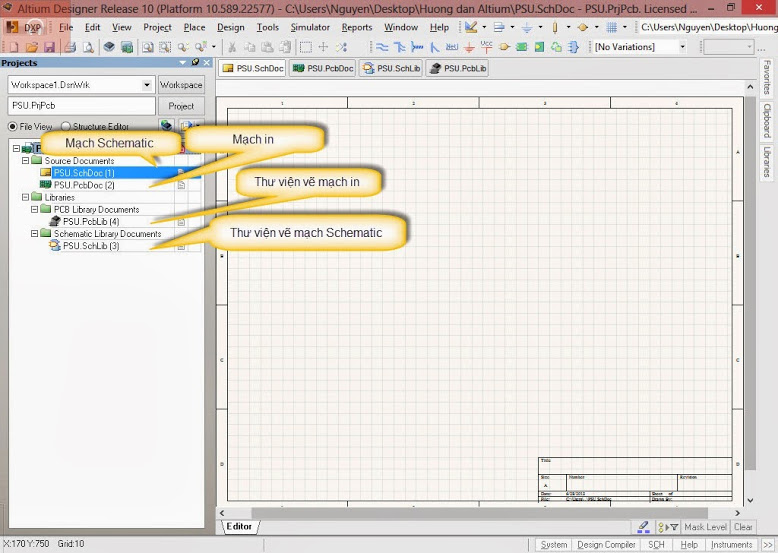
**Hình 3.10 Tạo file PCB**

* + Sau đó lưu lại file Schematic tương tự như Project, lưu trong cùng một thư mục để dễ quản lý. Nhấp chuột phải vào PCB1.PcbDoc trong giao diện Workspace chọn Save As, sau đó lưu lại.
  + Lưu ý là nếu chưa lưu các bạn sẽ không Update được từ sơ đồ mạch nguyên lý sang mạch in.
* Tạo thư viện vẽ Footprint để thực hiện trên mạch in.
  + Trong giao diện Workspace nhấp chuột phải chọn Add New to Project -> PCB Library.



**Hình 3.11 Tạo thư viện**

* Sau đó các lưu lại thư viện
* Và đây là giao diện cuối cùng sau khi đã tạo xong một Project.



**Hình 3.12 Project hoàn chỉnh**

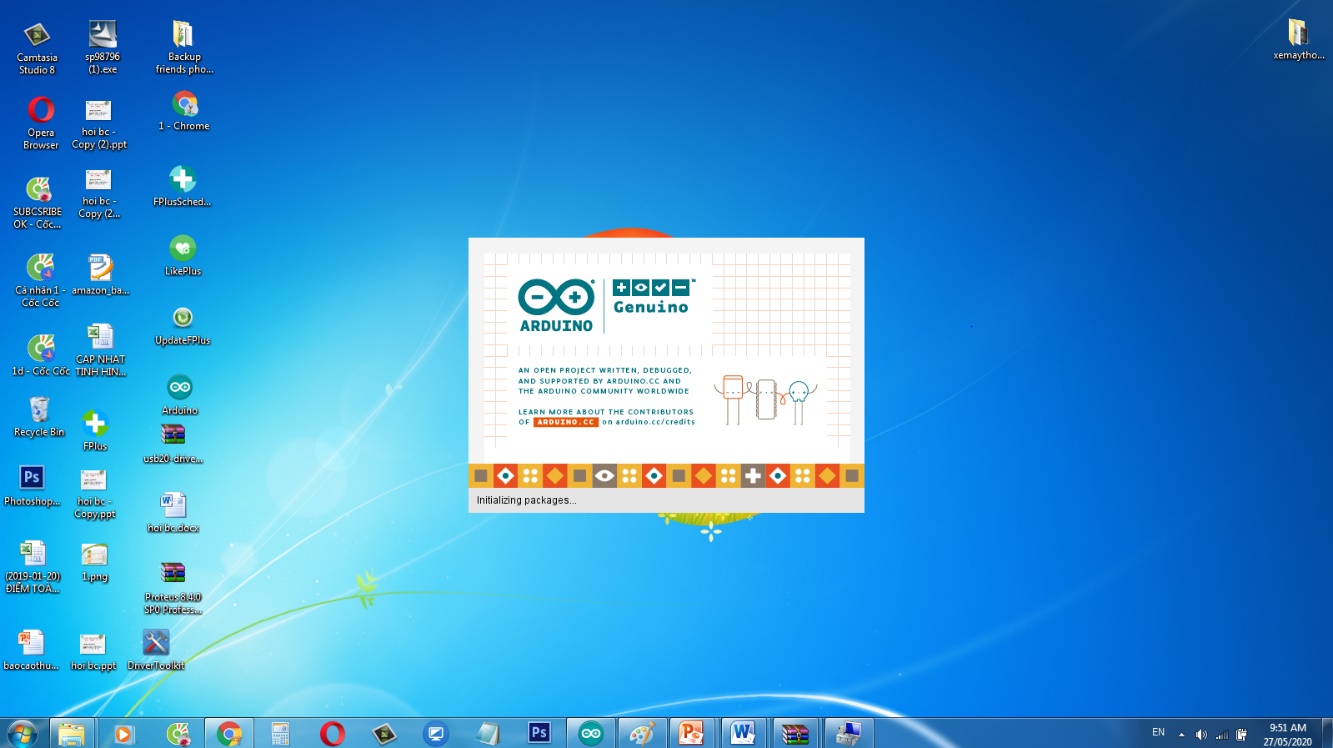
### **3.1.2 Arduino IDE**

Chúng ta có thể hiểu [Arduino IDE](https://chotroihn.vn/arduino) là một trình soạn thảo để viết code và nạp vào linh kiện arduino.

Bản thân Arduino là một một nền tảng mã nguồn mở bao gồm phần cứng và phần mềm. Phần cứng bao gồm các board mạch được thiết kế sẵn với các cảm biến, linh kiện (hiện nay đã có đến hơn 300000 bo mạch khác nhau được thương mại). Còn phần mềm của arduino có thể sử dụng các cảm biến, linh kiện ấy của arduino một các linh hoạt tùy thuộc vào mục đích sử dụng.

Ta có thể thấy, [Arduino IDE](https://chotroihn.vn/arduino) có vai trò quan trọng để nạp các chương trình code vào trong linh kiện arduino. Hiểu đơn giản thì phần mềm này như phần dây dẫn điện để đưa điện năng đến với động cơ quạt từ đó quạt mới hoạt động được.

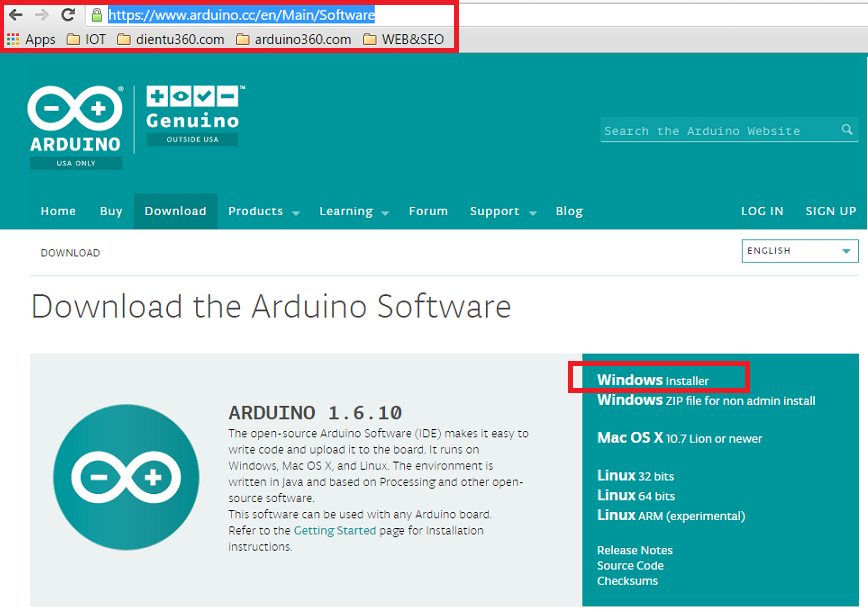
[Arduino IDE](https://chotroihn.vn/arduino) được viết bằng ngôn ngữ lập trình Java là ứng dụng đa nền tảng (cross-platform). Ngôn ngữ code cho các chương trình của arduino là bằng C hoặc C++.



Hình 2.1 :Giao diện phần mềm Arduino IDE

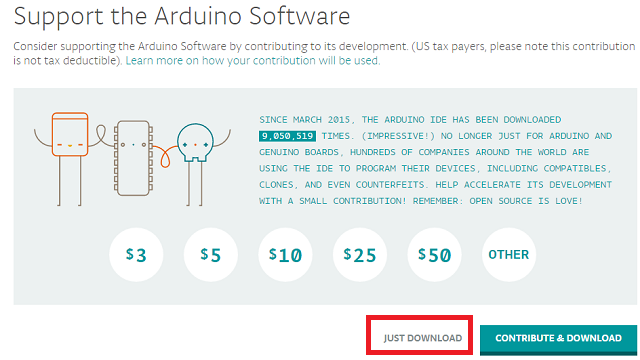
**Cài đặt phần mềm Arduino IDE**

Bước 1: Download trực tiếp phần mềm Arduino IDE hoặc Truy cập vào trang http://arduino.cc/en/Main/Software và tải về chương trình Arduino IDE mới nhất, phù hợp với hệ điều hành của máy mình bao gồm Windown, Mac OS hay Linux. Với Windown có hai bản Cài đặt (.exe) và bản Zip (chỉ cần giải nén và chạy). Tuy nhiên các bạn nên tải bản cài đặt (.exe) để trình cài đặt tự động cài đặt cả driver cho các board mạch arduino.



Hình 2.2. Địa chỉ tải phần mềm

Bước 2: Chọn JUST DOWNLOAD (free download)



Hình 2.3. Cách tải phần mềm

Bước 3 : Cài đặt driver để nhận board Arduino

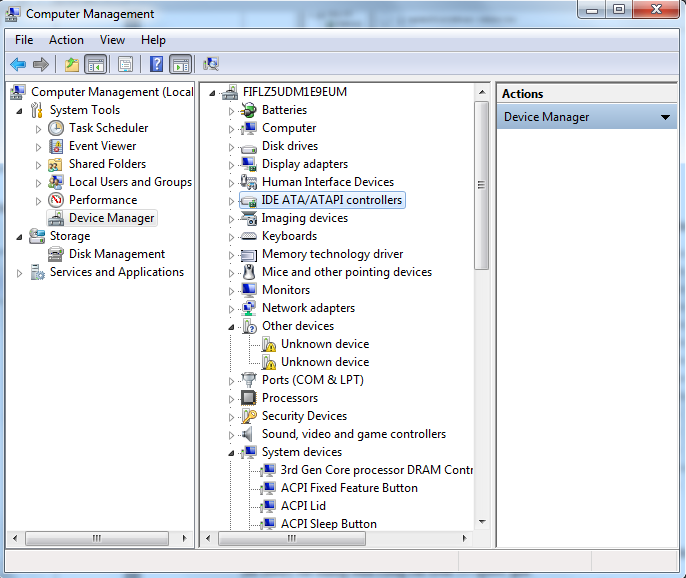
Mặc định khi cài đặt Arduino IDE phần mềm sẽ cài luôn các driver để máy tính có thể nhận được các board mạch Arduino. Tuy nhiên một số trường hợp máy tính không nhận được board mạch Arduino, khi đó các bạn làm như sau:

Cắm lại cáp USB nối giữa máy tính và Board Arduino, khi đó hiện ra bảng thông báo:



Hình 2.4. Thông báo khi cài phần mềm

Bây giờ bạn click vào Start Menu chọn Control Panel kế đến chúng ta chọn System and Security, click System và sau đó chọn Device Manager.



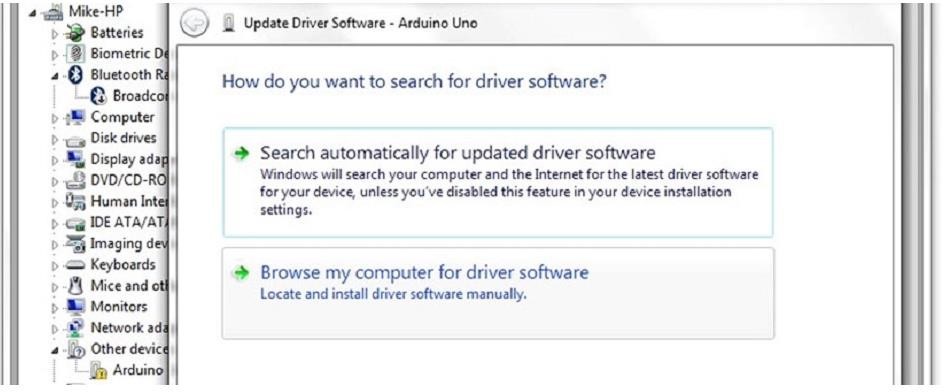
Hình 2.5. Kiểm tra driver đã được kết nối

 Chúng ta sẽ thấy cảnh báo màu vàng thiếu driver trên Arduino. Click chuột phải trên Arduino Uno icon sau đó chọn “Update Driver Software”.



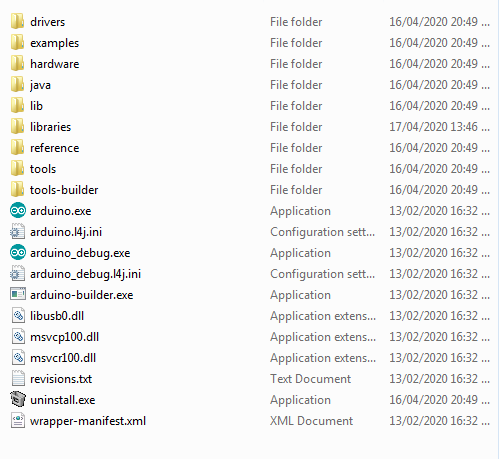
Hình 2.6. Update Driver cho phần mềm

Chọn “Browse my computer for driver software”.

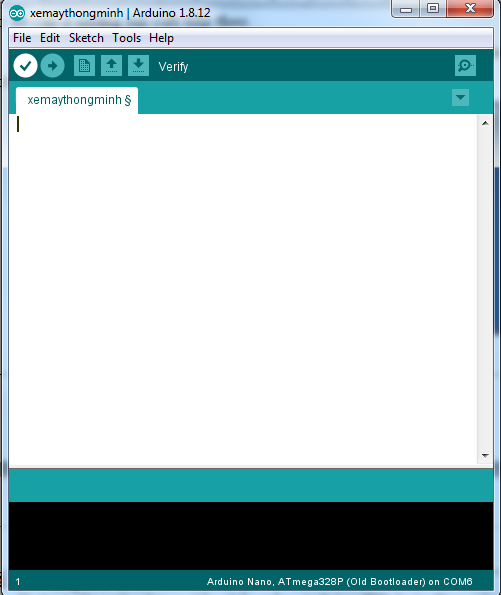


Hình 2.7. Chọn cách Update Driver

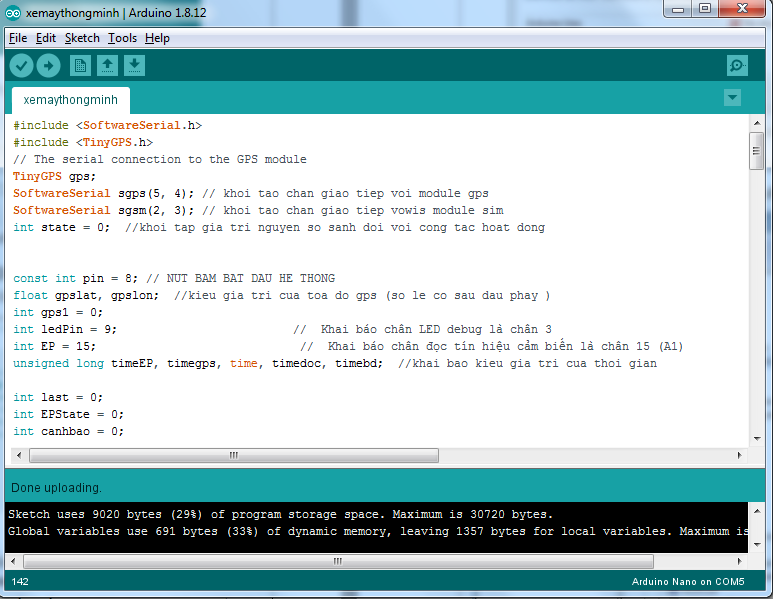
Chọn đường dẫn tới folder “driver” nơi mà phần mềm Arduino được lưu trữ.



Hình 2.8. Thư mục driver khi hoàn thành cài đặt



Hình 2.9. Phần mềm Arduino sau khi cài đặt thành công

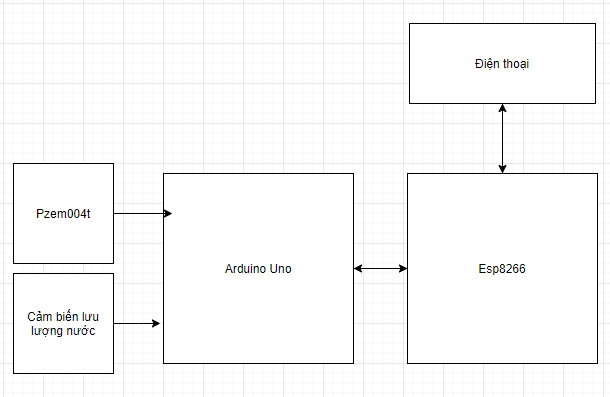


Hình 2.10. Kết quả code trên phần mềm Arduino

### **3.1.2 Visual Code**

### **3.1.3 Android Studio**

## **3.2 Sơ đồ hệ thống**



**Hình: Sơ đồ hệ thống**

## **3.4 Hình ảnh mạch thực tế**



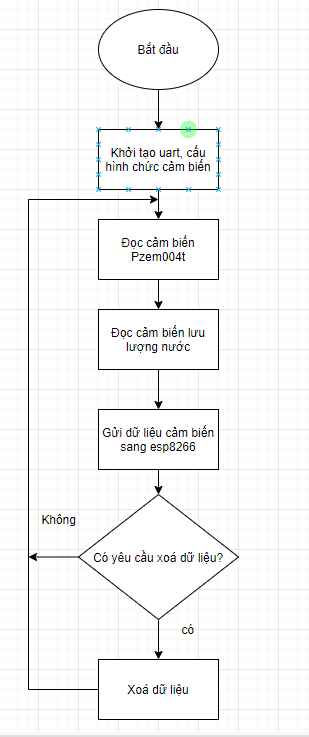
## **3.5 Nguyên lý hoạt động**

Khi được cấp nguồn, hệ thống sẽ tự động kết nối vào wifi đã được cài đặt source code, đồng thời sẽ khởi tạo các thông số phần cứng cơ bản để hoạt động.

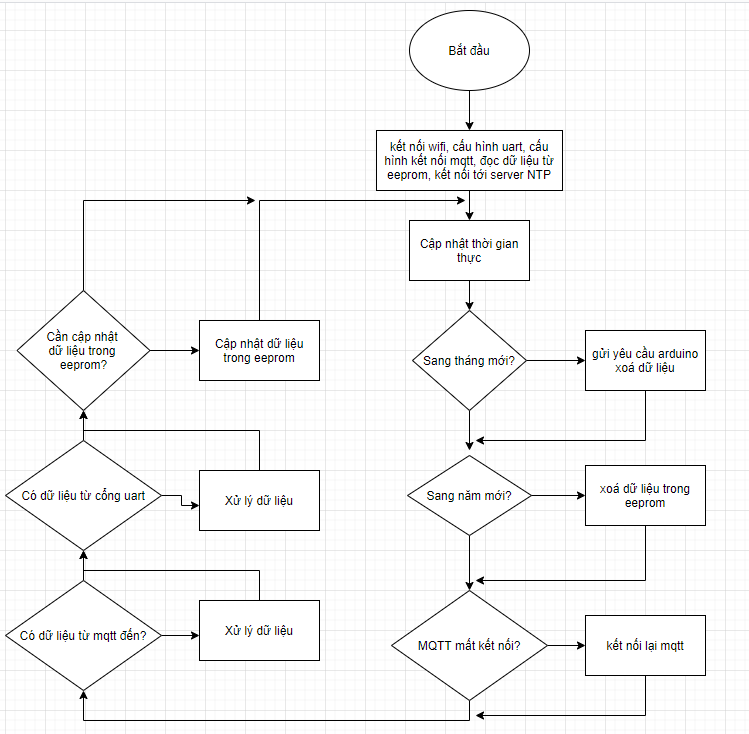
Sau khi khởi tạo xong các thành phần cơ bản, hệ thống sẽ đọc cảm biến và trả về thông tin dữ liệu của cảm biến về lượng điện tiêu thụ, lượng nước tiêu thụ mỗi khi có yêu cầu từ người dùng gửi đến thông qua giao thức mqtt.

Trên ứng dụng theo dõi sẽ có

## **3.6 Lưu đồ thuật toán**



Hình: Lưu đồ thuật toán của Arduino



Hình: Lưu đồ thuật toán của Esp8266

# **Kết Luận**

# **Nhận Xét**

# **Tài liệu tham khảo**

1. https://topdev.vn/blog/react-native/

# Phụ Lục