|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HỌC VIỆN KỸ THUẬT MẬT MÃ  **KHOA ĐIỆN TỬ-VIỄN THÔNG**    **ĐỒ ÁN 2**  **THIẾT KẾ THÙNG RÁC THÔNG MINH**   |  |  |  | | --- | --- | --- | | Nhóm sinh viên | **1. Cao Quốc Anh** | DT020101 | |  | **2. Đỗ Thị Mơ** | DT020131 | |  | **3. Nguyễn Thị Quyên**  **4. Nguyễn Quang Sơn**  **5. Mai Đức Trung** | DT020139  DT020141  DT020146 | | Giảng viên hướng dẫn | **Tô Thị Tuyết Nhung** |  |     **Hà Nội, 2021** |

**MỤC LỤC**

[**LỜI MỞ ĐẦU**](#_Toc93154912)

[**LỜI CẢM ƠN**](#_Toc93154913)

[**CHƯƠNG 1: TÌM HIỂU VỀ LÝ THUYẾT** 1](#_Toc93154914)

[**1.1.** **Cơ sở lý thuyết** 1](#_Toc93154915)

[**1.2.** **Tổng quan về cảm biến** 1](#_Toc93154916)

[**1.3.** **Cảm biến khoảng cách** 2](#_Toc93154917)

[**CHƯƠNG 2: THIẾT KẾ HỆ THỐNG** 12](#_Toc93154918)

[**2.1. Sơ đồ khối và chức năng các khối** 12](#_Toc93154919)

[2.1.1. Sơ đồ khối 12](#_Toc93154920)

[2.1.2. Chức năng các khối 12](#_Toc93154921)

[2.1.3 Nguyên lý hoạt động 13](#_Toc93154922)

[**2.2. Linh kiện phần cứng** 13](#_Toc93154923)

[2.2.1. Arduino uno R3 13](#_Toc93154924)

[2.2.2. Module NodeMCU ESP8266 17](#_Toc93154925)

[2.2.3. Module Sim 900A 22](#_Toc93154926)

[2.2.4. Module GPS NEO-6M 24](#_Toc93154927)

[2.2.5. Mạch hạ áp DC-DC 25](#_Toc93154928)

[2.2.6. Còi báo 27](#_Toc93154929)

[2.2.7. Cảm biến siêu âm HC-SR04 27](#_Toc93154930)

[2.2.8. Động cơ Servo SG90. 31](#_Toc93154931)

[2.2.9. Module Relay 5V 1 kênh. 33](#_Toc93154932)

[**2.3. Thiết kế phần mềm** 34](#_Toc93154933)

[2.3.1. Phần mềm hệ thống Arduino IDE 34](#_Toc93154934)

[2.3.2. App blynk 37](#_Toc93154935)

[**CHƯƠNG 3: THI CÔNG HỆ THỐNG** 44](#_Toc93154936)

[3.1. Sơ đồ thiết bị 44](#_Toc93154937)

[3.2. Sơ đồ nguyên lý 44](#_Toc93154938)

[3.3. Sơ đồ đi dây 45](#_Toc93154939)

[3.4. Lưu đồ thuật toán 46](#_Toc93154940)

[3.5. Sản phẩm 47](#_Toc93154941)

[**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 48](#_Toc93154942)

[**TÀI LIỆU THAM KHẢO** 50](#_Toc93154943)

[**PHỤ LỤC** 51](#_Toc93154944)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1: Cảm biến khoảng cách bằng Laser 12](#_Toc93129522)

[Hình 1.2: Mô tả cảm biến siêu âm 13](#_Toc93129523)

[Hình 1.3: Nguyên lý cảm biến siêu âm 14](#_Toc93129524)

[Hình 1.4:Ứng dụng cảm biến siêu âm phát hiện vật cản 15](#_Toc93129525)

[Hình 1.5: Ứng dụng cảm biến siêu âm trong công nghiệp 16](#_Toc93129526)

[Hình 1.6: Ứng dụng cảm biến siêu âm đo mức nước 17](#_Toc93129527)

[Hình 1.7: Ứng dụng của cảm biến vân tay siêu âm trên Samsung S10 17](#_Toc93129528)

[Hình 1.8: Cảm biến tiệm cận đo khoảng cách 18](#_Toc93129529)

[Hình 2.1: Sơ đồ khối 20](#_Toc93129530)

[Hình 2.2: Arduino Uno R3 21](#_Toc93129531)

[Hình 2.3: Sơ đồ chân Arduino Uno R3 23](#_Toc93129532)

[Hình 2.4: Module thu phát wifi NodeMCU ESP8266 CP2102 25](#_Toc93129533)

[Hình 2.5: Sơ đồ chân module NodeMCU ESP8266 CP2102 27](#_Toc93129534)

[Hình 2.6: Module Sim900A 30](#_Toc93129535)

[Hình 2.7: Sơ đồ chân Module Sim900A 31](#_Toc93129536)

[Hình 2.8: Sơ đồ đấu nối module Sim900A với vi điều khiển 31](#_Toc93129537)

[Hình 2.9: Module GPS NEO-6M 32](#_Toc93129538)

[Hình 2.10: Sơ đồ chân Module GPS NEO-6M 33](#_Toc93129539)

[Hình 2.11: Module hạ áp LM2596 34](#_Toc93129540)

[Hình 2.12: Sơ đồ chân module hạ áp LM2596 34](#_Toc93129541)

[Hình 2.13: Còi báo 35](#_Toc93129542)

[Hình 2.14: Cảm biến siêu âm HC-SR04 36](#_Toc93129543)

[Hình 2.15: Sơ đồ chân cảm biến siêu âm HC-SR04 37](#_Toc93129544)

[Hình 2.16: Động cơ servo 39](#_Toc93129545)

[Hình 2.17: Sơ đồ đấu nối servo với vi điều khiển 40](#_Toc93129546)

[Hình 2.18: Module Relay 5V 1 kênh 41](#_Toc93129547)

[Hình 2.19: Sơ đồ chân module Relay 5V 1 kênh 42](#_Toc93129548)

[Hình 2.20 Giao diện phần mềm Arduino IDE 43](#_Toc93129549)

[Hình 2.21 Chức năng thanh công cụ trong Arduino IDE 43](#_Toc93129550)

[Hình 2.22: Ví dụ minh họa 44](#_Toc93129551)

[Hình 2.23: Kết quả của ví dụ trên màn hình 45](#_Toc93129552)

[Hình 2.24: Tạo tài khoản app Blynk 46](#_Toc93129553)

[Hình 2.25: Tạo project mới trên app blynk 46](#_Toc93129554)

[Hình 2.26: Đặt tên và chọn mạch điều khiển cho project 47](#_Toc93129555)

[Hình 2.27: Sao chép lấy lại hoặc gửi lại mail Auth Token 48](#_Toc93129556)

[Hình 2.28: Giao diện app blynk 48](#_Toc93129557)

[Hình 2.29: Add Widget 49](#_Toc93129558)

[Hình 2.30: Add Energy 49](#_Toc93129559)

[Hình 2.31: Widget Button 50](#_Toc93129560)

[Hình 2.32: Cài đặt Widget Button 51](#_Toc93129561)

[Hình 2.33: Kết quả đạt được khi thiết lập các thông số 51](#_Toc93129562)

[Hình 3. 1: Sơ đồ thiết bị 52](#_Toc93129563)

[Hình 3. 2: Sơ đồ đi dây với vi điều khiển Arduino Uno R3 53](#_Toc93129564)

[Hình 3. 3: Sơ đồ đi dây với vi điều khiển ESP8266 53](#_Toc93129565)

[Hình 3. 4: Lưu đồ thuật toán 54](#_Toc93129566)

[Hình 3. 5: Lưu đồ thuật toán 54](#_Toc93129567)

**DANH MỤC BẢNG**

[Bảng 2. 1: Thông số kỹ thuật của Arduino Uno R3 22](#_Toc93129962)

[Bảng 2. 2: Bảng nối chân giữa HC-SR04 với Arduino Uno R3 37](#_Toc93129963)

[Bảng 2. 3: Bảng đấu nối giữa động cơ Servo với Arduino Uno R3 40](#_Toc93129964)

**DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Tiếng Anh** | **Tiếng Việt** |
| A - MPDU | Aggregate MAC Protocol Data Unit | Đơn vị dữ liệu giao thức MAC tổng hợp |
| A - MSDU | Aggregate MAC Service Data Unit | Đơn vị dữ liệu dịch vụ MAC tổng hợp |
| AES | Advanced Encryption Standard | Thuật toán mã hóa khối |
| CPU | Central Processing Unit |  |
| DCXO | Digital Controlled Crystal Oscillators | Bộ tạo dao động tinh thể được điều khiển kỹ thuật số |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |  |
| GIPO | Global Internet Policy Observatory | Đài quan sát chính sách Internet toàn cầu |
| GND | Ground | Đất |
| GPIO | General Purpose Input Output | Mục đích chung của đầu vào, đầu ra |
| I/O | Hypertext transfer protocol | Giao thức truyền siêu văn bản |
| IC | Integrated circuit | Mạch tích hợp |
| LED | Light Emitting Diode | Điốt phát quang |
| MCU | Micro control unit | Vi điều khiển |
| MIMO | Multiple-input and multipleoutput | Nhiều đầu vào và nhiều đầu ra |
| MISO | Master Input / Slave Output | Truyền dữ liệu từ Master đến Slave |
| MOSI | Master Output / Slave Input | Truyền dữ liệu từ Slave đến Master |
| RAM | Random Access Memory | Bộ nhớ truy cập tạm thời |
| RISC | Reduced Instructions Set Computer | Máy tính với tập lệnh đơn giản hóa |
| SDIO | Secure Digital Input Output | Đầu ra đầu vào kỹ thuật số an toàn |
| SPI | Serial Peripheral Interface | Giao tiếp ngoại vi nối tiếp |
| TKIP | Temporal Key Integrity Protocol | Giao thức về tính toàn vẹn Key Temporal |
| UART | Universal Asynchronous Receiver Transmitter | Chuẩn giao tiếp nối tiếp với sự hỗ trợ của phần cứng |
| WAPI | WLAN Authentication and Privacy Infrastructure | Cơ sở hạ tầng bảo mật và xác thực mạng WLAN |

# **LỜI MỞ ĐẦU**

Hiện nay, khi xã hội ngày càng phát triển, dân cư ngày càng đông đúc đặc biệt là ở các thành phố lớn, các khu công nghiệp từ đó dẫn đến lượng rác thải con người xả ra ngày càng tăng. Do đó, quản lý chất thải là một trong những vấn đề nóng hổi mà thế giới phải đối mặt không phân biệt đó là nước phát triển hay đang phát triển. Vấn đề chính là trong việc quản lý chất thải, thùng rác ở nơi công cộng luôn luôn trong tình trạng đầy và quá tải trước khi được xe rác tới lấy và đưa lượng rác đó ra một vị trí tập trung lượng rác lớn khác để xử lý. Việc thùng rác nơi công cộng luôn đầy và tràn ra lần lượt dẫn đến nhiều mối nguy hiểm khác nhau như bốc mùi, ô nhiễm môi trường cảnh quan và đó cũng là gốc rễ, nguyên nhân chính gây ra những căn bệnh lây nhiễm. Xuất phát từ nguyên nhân đó, **“Thùng rác thông minh”** được ra đời.

Với tính năng tự động gửi cảnh báo đầy về điện thoại sẽ giúp cho việc thu gom rác được kịp thời, nhanh chóng, hạn chế được tình trạng tràn rác.

Trong quá trình thực hiện đề tài **“Thùng rác thông minh”**, mặc dù nhóm thực hiện đề tài đã cố gắng hoàn thành nhiệm vụ đặt ra và đúng thời hạn nhưng do còn hạn chế về kiến thức và thời gian thực hiện nên chắc chắn sẽ không tránh khỏi những thiếu sót, hạn chế, mong quý Thầy/Cô và các bạn sinh viên thông cảm. Nhóm rất biết ơn và mong nhận được những ý kiến đóng góp của quý Thầy/Cô và các bạn sinh viên về đề tài này.

# **LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường Học viện kỹ thuật mật mã, đặc biệt là các thầy cô ở khoa Điện tử viễn thông của trường và chúng em cũng xin chân thành cám ơn cô **Tô Thị Tuyết Nhung**, phụ trách hướng dẫn và nhiệt tình hỗ trợ nhóm chúng em trong quá trình nhóm chúng em hoàn thành và hoàn thiện sản phẩm Đồ án 2.

Trong quá trình thực hiện Đồ án và làm bài báo cáo đồ án, do kiến thức cũng như kinh nghiệm thực tế còn nhiều hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp thầy, cô để em học hỏi được nhiều kĩ năng, kinh nghiệm và sẽ hoàn thành tốt hơn trong Đồ án tốt nghiệp cũng như bài báo cáo tốt nghiệp sắp tới.

Em xin chân thành cảm ơn!

# **CHƯƠNG 1: TÌM HIỂU VỀ LÝ THUYẾT**

* 1. **Cơ sở lý thuyết**

Với sự gia tăng dân số, chúng ta có sự gia tăng rác thải xung quanh các khu vực đô thị. Thùng rác thông minh hoạt động tự động để giúp giải quyết vấn đề này bằng cách sử dụng IOT và mạch dựa trên cảm biến. Các thùng rác thông thường yêu cầu được mở bằng cách nhấn chân vào cần gạt của nó và sau đó ném rác. Ngoài ra một người cần theo dõi khi nào đầy để có thể làm trống và không bị tràn. Nhưng thùng rác thông minh có thể tự làm tất cả những điều này. Khi có người tiếp cận thùng rác, thùng rác sẽ tự động mở và đóng sau một khoảng thời gian nhất định. Ngoài ra, thùng rác còn bao gồm một cảm biến siêu âm cảm nhận mức độ rác trong thùng và tự động phát hiện nếu nó sắp đầy, gửi thông báo qua tin nhắn hoặc cuộc gọi cho người/ quản lí.

* 1. **Tổng quan về cảm biến**

Nền tảng IoT hoạt động và cung cấp các dạng thông tin và dữ liệu có giá trị với việc sử dụng nhiều cảm biến khác nhau. Chúng phục vụ để thu thập dữ liệu, đẩy nó và chia sẻ nó với một mạng lưới toàn bộ các thiết bị kết nối. Tất cả dữ liệu được thu thập này làm cho các thiết bị có thể hoạt động độc lập và toàn bộ hệ sinh thái ngày càng trở nên thông minh hơn.

Các ngành công nhiệp và tổ chức xã hội đã sử dụng nhiều loại cảm biến khác nhau trong một thời gian dài nhưng phát minh ra Internet vạn vật (IoT) đã khiến cho cảm biến đóng vai trò quan trọng và khiến vai trò của cảm biến đến một mức độ hoàn toàn khác.

Bằng cách kết hợp một bộ cảm biến và một mạng truyền thông, các thiết bị chia sẻ thông tin với nhau và đang cải thiện tính hiệu quả và chức năng của chúng. Lấy xe Tesla làm ví dụ, tất cả các cảm biến trên một chiếc xe ghi lại nhận thức của chúng về môi trường xung quanh, tải thông tin lên một cơ sở dữ liệu khổng lồ. Dữ liệu sau đó được xử lý và tất cả các thông tin quan trọng mới được gửi đến tất cả các phương tiện khác. Đây là một quá trình liên tục, thông qua đó một đội xe Tesla đang trở nên thông minh hơn mỗi ngày.

Vậy cảm biến là gì? Chúng ta hãy cùng đi tìm hiểu và xem xét một số cảm biến quan trọng, phổ biến được sử dụng rỗng rãi trong thế giới IoT.

Cảm biến là thiết bị điện tử cảm nhận những trạng thái, quá trình vật lý hay hóa học ở môi trường cần khảo sát và biến đổi thành [tín hiệu](https://bkaii.com.vn/tin-tuc/302-khai-niem-va-phan-loai-tin-hieu) điện để thu thập thông tin về trạng thái hay quá trình đó. Thông tin được xử lý để rút ra [tham số](https://vi.wikipedia.org/wiki/Tham_s%E1%BB%91) định tính hoặc định lượng của môi trường, phục vụ các nhu cầu nghiên cứu khoa học kỹ thuật hay dân sinh và gọi ngắn gọn là đo đạc, phục vụ trong truyền và xử lý thông tin hay trong điều khiển các quá trình khác.

Các đại lượng cần đo thường không có tính chất điện như nhiệt độ, áp suất,… tác động lên cảm biến cho ta một đại lượng đặc trưng mang tính chất điện như điện tích, điện áp, dòng điện,… chứa đựng thông tin cho phép xác định giá trị của đại lượng đo.

Cảm biến thường được đặt trong các vỏ bảo vệ tạo thành đầu thu hay đầu dò, có thể có kèm các mạch điện hỗ trợ và nhiều khi trọn bộ đó lại được gọi luôn là "cảm biến".

* 1. **Cảm biến khoảng cách**

Cảm biến khoảng cách được sử dụng để đo khoảng cách xuất phát từ điểm tham chiếu đến vật thể. Cảm biến khoảng cách được sử dụng rộng rãi và phổ biến trong môi trường công nghiệp, kho bãi…

***Phân loại cảm biến đo khoảng cách***

* Cảm biến khoảng cách bằng tia laser
* Xác định khoảng cách bằng cảm biến siêu âm
* Cảm biến tiệm cận đo khoảng cách

***Nguyên lý đo của cảm biến khoảng cách***

Cách đo khoảng cách dựa vào tia laser thì nguyên lý là thu phát tia laser đến vật thể cần đo. Ưu điểm của loại đo này sẽ đo được vật ở khoảng cách xa, vùng đo rộng.

***Khái niệm tín hiệu ngõ ra của cảm biến khoảng cách***

* Đối với dòng cảm biến khoảng cách thì tín hiệu ngõ ra thông thường có dạng là analog, NPN hoặc PNP, rơle,…

Phụ thuộc vào nhu cầu điều khiển giám sát mà ta có thể dễ dàng lựa chọn loại ngõ cho phù hợp. Nhưng loại ngõ ra dạng 4-20mA vẫn là lựa chọn hàng đầu. Với khả năng khiến tín hiệu ít nhiễu hơn, truyền đi với khoảng cách xa khi sử dụng bộ truyền tín hiệu, khuếch đại tín hiệu.

***Sai số của cảm biến đo khoảng cách?***

Mỗi loại cảm biến sẽ có một độ sai số khác nhau tương ứng với từng nhu cầu sử dụng khác nhau. Chính vì vậy ta cần chọn loại có độ sai số cho phù hợp. Một máy cảm biến có sai số phụ thuộc vào thang đo và nguyên lý đo.

Ví dụ: Khoảng cách đo của cảm biến từ ngắn, tính bằng đơn vị mm. Chính vì vậy độ sai số rất thấp.

Cảm biến đo khoảng cách có khả năng đo được khoảng cách xa. Có thể đến hàng trăm, hàng ngàn mét. Vì vậy sai số ở đây tính bằng đơn vị cm, nhưng vẫn trong giới hạn cho phép của phép đo.

***Khoảng cách đo được tối đa là bao nhiêu?***

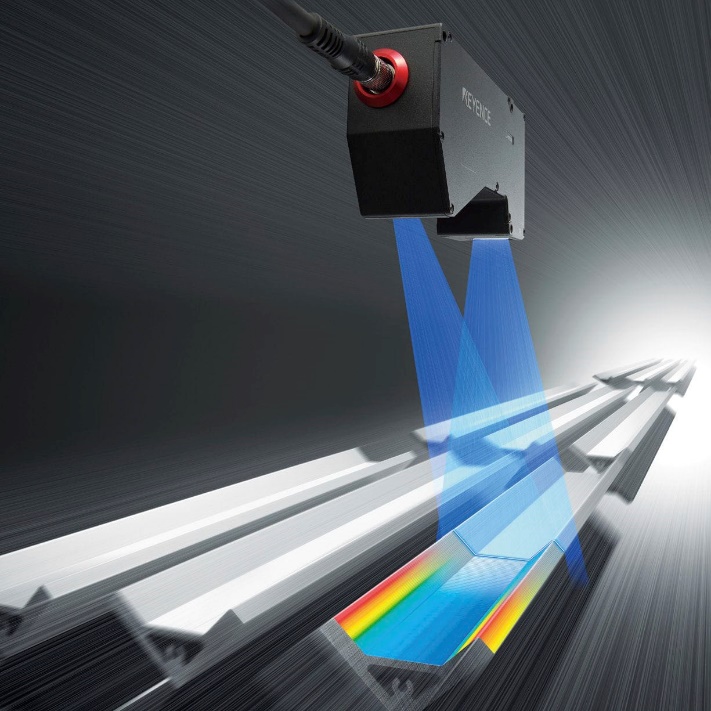
* Dòng cảm biến laser có khả năng đo được khoảng cách xa nhất lên đến 3000m.

Với dòng cảm biến tiệm cận thì khoảng cách đo chỉ dừng lại tối đa 100mm. Chỉ có khả năng phát hiện vật.

***Cảm biến khoảng cách bằng tia laser***

Cảm biến Laser hay còn được biết đến là máy đo khoảng cách Laser, thước đo Laser,… Là một thiết bị phát ra chùm tia Laser rồi thu nhận tín hiệu phản xạ ánh sáng. Thiết bị này dùng để đo khoảng cách, độ dài,…. Chúng cho ra kết quả đo chính xác có độ sai số thấp hơn rất nhiều so với khoảng thước đo thông thường.

Khái niệm hàng trăm, triệu năm ánh sáng cũng có thể cho thấy tác dụng của thước đo laser này. Chúng có thể đo được những khoảng cách rất lớn mà những thước đo cơ bản không thể đo được.



Hình 1.1: Cảm biến khoảng cách bằng Laser

***Nguyên lý hoạt động của cảm biến Laser:***

Người dùng sẽ đặt cố định vị trí nguồn Laser, khi bấm nút khởi động, thiết bị sẽ phát ra chùm sáng Laser. Chính vị trí tia Laser đến mốc cần đo, khi tia sáng chạm vào vật cản chúng sẽ phản xạ ánh sáng quay lại bộ thu phát tín hiệu. Lúc này thiết bị sẽ tiến hành phân tích và hiển thị kết quả dưới dạng thông số người dùng cài đặt.

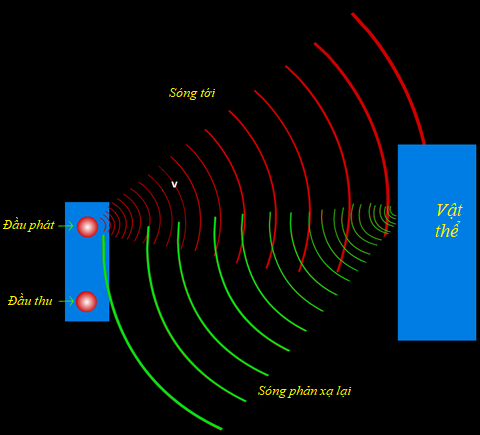
Các bước nói chi tiết sẽ cảm thấy phức tạp, nhưng khi thực hành thực tế, tốc độ xử lý của thiết bị rất nhanh. Gần như là đồng thời với việc chiếu tia Laser đến vị trí cần đo.

***Ưu điểm khi sử dụng cảm biến Laser:***

* Độ chính xác, tốc độ xử lý cao. Cảm biến Laser có thể đưa ra kết quả chính xác trong phạm vi từ 0.05m – 300m chỉ mất tối đa 0.5 giây.
* Thiết kế nhỏ gọn. Mang trong mình công dụng to lớn, nhưng cảm biến Laser có thiết kế rất nhỏ gọn. Người dùng có thể cầm tay, bỏ túi dễ dàng. Việc này hỗ trợ tốt cho những công việc đòi hỏi sự di chuyển.
* Sử dụng dễ dàng, tiết kiệm năng lượng. Máy đo khoảng cách bằng Laser có rất nhiều loại phù hợp với nhiều nhu cầu làm việc riêng. Nhưng chúng đều rất dễ sử dụng. Thường các thiết bị luôn có các nút bấm chức năng ngay trên thân máy. Nhờ vậy người dùng có thể dễ dàng thao tác và thu nhận kết quả đo dễ dàng. Với việc nguồn sử dụng điện không cần quá cao, thiết bị đo thường dùng Pin để cung cấp điện năng. Nên rất tiết kiệm năng lượng và dễ dàng mua và thay thế trong quá trình sử dụng

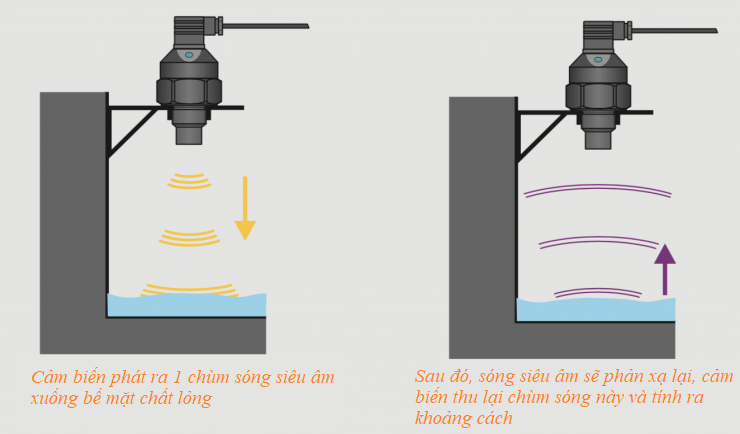
***Cảm biến khoảng cách bằng siêu âm***

Nói một cách đơn giản, [cảm biến siêu âm](https://cambiendo.vn/cam-bien-sieu-am-la-gi/) là một thiết bị cảm biến hoạt động dựa trên sóng siêu âm. Cũng giống như các loại cảm biến áp suất hay cảm biến nhiệt độ, cảm biến siêu âm được dùng chủ yếu là để đo khoảng cách hoặc vận tốc. Ngoài ra thì còn được sử dụng trong các ứng dụng như làm sạch bằng sóng siêu âm hoặc dùng trong siêu âm y khoa (siêu âm chuẩn đoán hình ảnh).



Hình 1.2: Mô tả cảm biến siêu âm

Nguyên lý cảm biến siêu âm:



Hình 1.3: Nguyên lý cảm biến siêu âm

Đầu tiên, đầu cảm biến sẽ phát ra 1 chùm sóng siêu âm xuống bề mặt cần đo khoảng cách. Khi sóng siêu âm gặp bề mặt vật cản sẽ phản xạ ngược lại. Khi đó cảm biến sẽ thu lại các chùm sóng siêu âm này.

Dựa vào thời gian phản xạ và vận tốc của sóng, cảm biến sẽ tính ra được khoảng cách từ cảm biến xuống bề mặt chất lỏng.

Ưu và nhược điểm của cảm biến sóng siêu âm:

**Ưu điểm của cảm biến sóng siêu âm:**

Sử dụng sóng siêu âm nên có thể đo khoảng cách mà không cần tiếp xúc với vật chất cần đo. Vì thế, cảm biến siêu âm thường được dùng để đo mức chất lỏng có độ ăn mòn cao như acid hoặc xăng, dầu,…

Sóng siêu âm là một loại âm thanh có tần số cao nên độ nhạy của cảm biến rất cao, thời gian đáp ứng nhanh.

Độ chính xác của cảm biến siêu âm gần như là tuyệt đối, sai số trung bình khoảng 0,15% đối với khoảng cách 2m trở lại.

*Nhược điểm của cảm biến sóng siêu âm:*

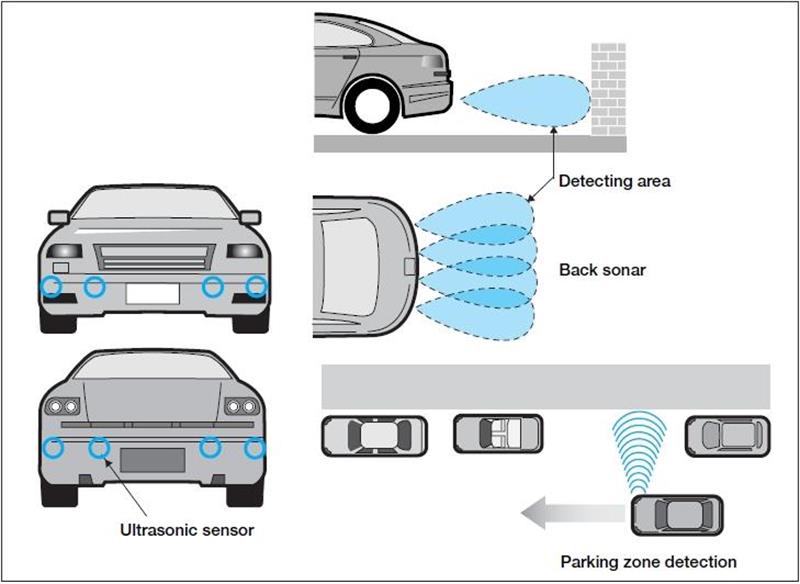
Cảm biến siêu âm chịu ảnh hưởng nhiều bởi nhiệt độ và áp suất. Vì thế nó chỉ hoạt động tốt nhất ở môi trường có nhiệt độ từ 60 độ C trở xuống và áp suất khoảng 1 bar trở lại.

Các loại cảm biến siêu âm đo mức chất lỏng có chi phí đầu tư ban đầu là khá cao so với các loại cảm biến đo mức chất lỏng khác.

Một nhược điểm khác là cảm biến siêu âm rất dễ bị nhiễu tín hiệu nên khi lắp đặt, bạn cần phải lắp theo đúng hướng dẫn của nhà sản xuất.

**Ứng dụng cảm biến siêu âm:**

***Cảm biến siêu âm phát hiện vật cản:***

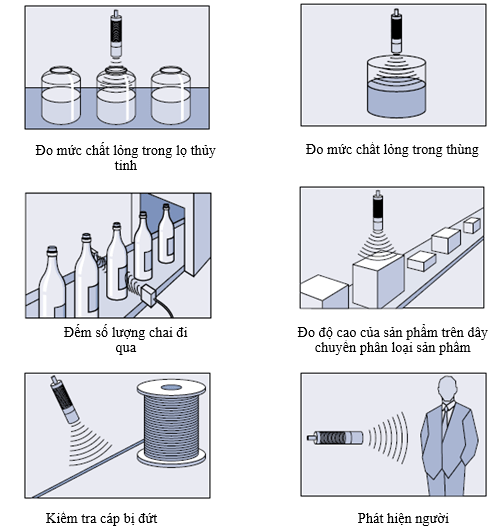


Hình 1.4:Ứng dụng cảm biến siêu âm phát hiện vật cản

Khi phát hiện phần đầu xe hoặc đuôi xe gần va chạm thì cảm biến sẽ xuất tín hiệu báo động về để báo cho tài xế biết là sắp có va chạm.

***Cảm biến siêu âm công nghiệp:***

Trong công nghiệp, cảm biến siêu âm được ứng dụng trong những ứng dụng phát hiện dị tật của sản phẩm; phát hiện sản phẩm bị ngã đổ trên băng chuyền, phát hiện sản phẩm bị nứt hoặc bể,….



Hình 1.5: Ứng dụng cảm biến siêu âm trong công nghiệp

Ngoài ra, người ta còn dùng cảm biến siêu âm để đo mức nhiên liệu trong bể hoặc xác định kích thước của sản phẩm sau sản xuất.

***Cảm biến siêu âm đo khoảng cách:***

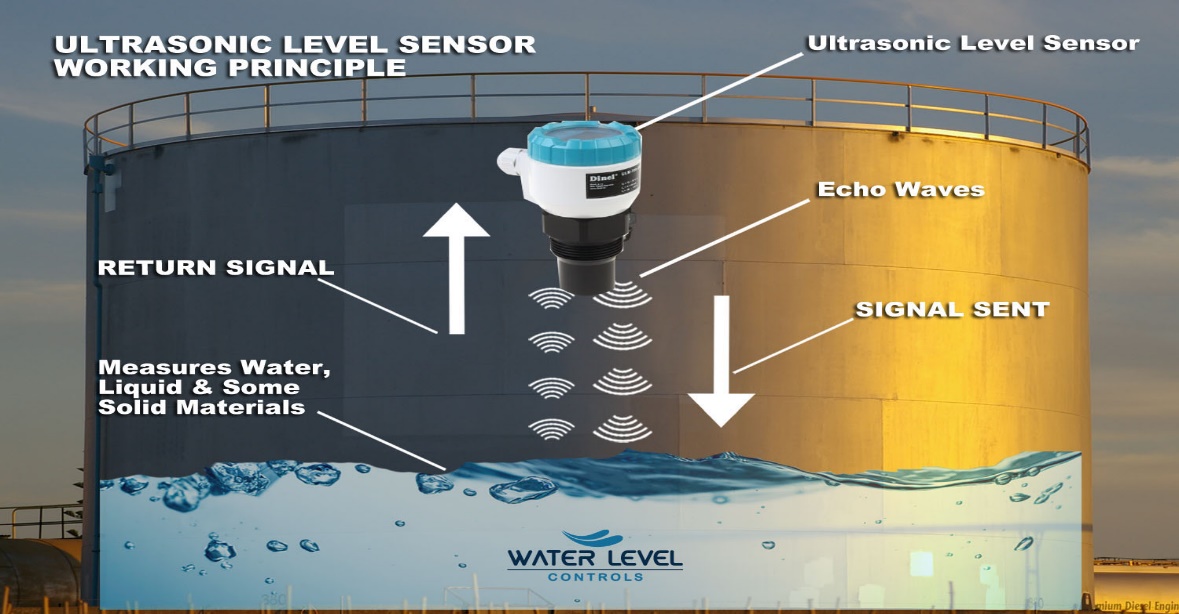
Khi sử dụng cảm biến siêu âm; cảm biến sẽ phát ra 1 chùm tia sóng hình nón xuống vị trí cần đo khoảng cách. Thông thường người ta dùng cảm biến siêu âm để đo khoảng cách từ cảm biến xuống bề mặt sản phẩm trên băng chuyền.

Nếu khoảng cách từ cảm biến xuống bề mặt sản phẩm đột ngột tăng cao thì khả năng là sản phẩm đó đang bị móp, méo hoặc đang bị nứt. Từ đó cảm biến sẽ báo động và loại sản phẩm đó ra khỏi băng chuyền.

***Cảm biến siêu âm đo mức nước:***

Ngoài ứng dụng để đo mức nước, người ta còn dùng cảm biến siêu âm đo mức chất lỏng nói chung. Tức là bao gồm cả dầu ăn, dầu diesel, socola, nước trái cây, đường, muối, acid ăn mòn,….

Lý do là vì dòng cảm biến siêu âm đo mức chất lỏng này có thể đo mức mà không cần tiếp xúc với môi trường đo nên không bị ảnh hưởng bởi độ ăn mòn và cũng đươc dùng trong các ứng dụng đo mức thực phẩm.



Hình 1.6: Ứng dụng cảm biến siêu âm đo mức nước

***Cảm biến vân tay siêu âm:***

Cảm biến này hoạt động dựa trên công nghệ được gọi là “3D Sonic Sensor” . Nó hoạt động bằng cách thu lại sóng âm bị phản hồi lại từ làn da của bạn. Nó sẽ ghi lại thông tin chi tiết của ngón tay xuyên qua nước, kem dưỡng da và dầu mỡ, vào ban đêm hoặc dưới ánh sáng ban ngày.



Hình 1.7: Ứng dụng của cảm biến vân tay siêu âm trên Samsung S10

Khi đặt tay lên bàn hình điện thoại, sóng siêu âm sẽ phát ra từ màn hình đến các vân tay của bạn. Bởi vì các dấu vân tay của bạn là các rãnh lồi, lõm khác nhau nên khi sóng siêu âm tiếp xúc sẽ phản xạ sóng lại có khoảng cách khác nhau.

**Cảm biến tiệm cận đo khoảng cách.**

Cảm biến tiệm cận (còn được gọi là “Công tắc tiệm cận” hoặc đơn giản là “PROX” tên tiếng anh là Proximity Sensors) phản ứng khi có vật ở gần cảm biến. Trong hầu hết các trường hợp, khoảng cách này chỉ là vài mm. Cảm biến tiệm cận thường phát hiện vị trí cuối của chi tiết máy và tín hiệu đầu ra của cảm biến khởi động một chức năng khác của máy. Đặc biệt cảm biến này hoạt động tốt ngay cả trong những môi trường khắc nghiệt.

Cảm biến tiệm cận chuyển đổi tín hiệu về sự chuyển động hoặc xuất hiện của vật thể thành tín hiệu điện. Có 3 hệ thống phát hiện để thực hiện công việc chuyển đổi này: hệ thống sử dụng dòng điện xoáy được phát ra trong vật thể kim loại nhờ hiện tượng cảm ứng điện từ, hệ thống sử dụng sự thay đổi điện dung khi đến gần vật thể cần phát hiện, hệ thống sử dụng nam châm và hệ thống chuyển mạch cộng từ.



Hình 1.8: Cảm biến tiệm cận đo khoảng cách

Đặc điểm:

* Phát hiện vật thể không cần tiếp xúc, không tác động lên vật, khoảng cách xa nhất tới 30mm
* Hoạt động ổn định, chống rung động và chống shock tốt.
* Tốc độ đáp ứng nhanh, tuổi thọ cao so với công tắc giới hạn (limit switch).
* Đầu sensor nhỏ có thể lắp ở nhiều nơi.
* Có thể sử dụng trong môi trường khắc nghiệt

**Nguyên lí hoạt động:**

Cảm biến tiệm cận hoạt động theo nguyên lý trường điện từ phát ra xung quanh cảm biến với khoảng cách tối đa 30mm và gặp vật thể thì nó sẽ phát tín hiệu truyền về bộ xử lý

**Phân loại:**

Có hai loại cảm biến tiệm cận chính có thể kể đến. Đó là loại cảm ứng từ và loại điện dung.

*Cảm biến tiệm cận loại cảm ứng từ*

* Cảm ứng từ loại có bảo vệ (Shielded): Từ trường được tập trung trước mặt sensor nên ít bị nhiễu bởi kim loại xung quanh, tuy nhiên khoảng cách đo ngắn đi.
* Cảm ứng từ loại không có bảo vệ (Un-Shielded): Không có bảo vệ từ trường xung quanh mặt sensor nên khoảng cách đo dài hơn, tuy nhiên dễ bị nhiễu của kim loại xung quanh.

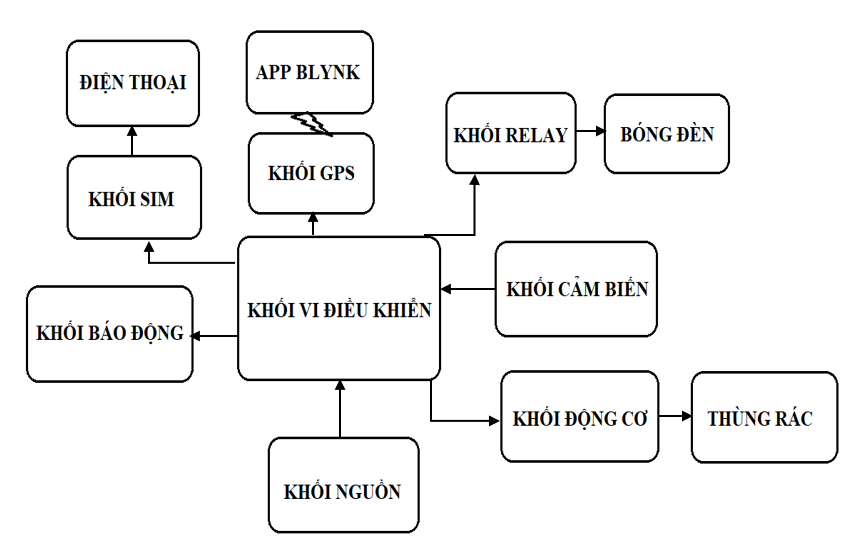
*Cảm biến tiệm cận loại cảm ứng điện dung*

* Cảm ứng này phát hiện theo nguyên tắc tĩnh điện (sự thay đổi điện dung giữa vật cảm biến và đầu sensor), có thể phát hiện tất cả vật thể.

# **CHƯƠNG 2: THIẾT KẾ HỆ THỐNG**

## **2.1. Sơ đồ khối và chức năng các khối**

### 2.1.1. Sơ đồ khối



Hình 2.1: Sơ đồ khối

### 2.1.2. Chức năng các khối

* **Khối nguồn**: là hệ thống điện áp cung cấp cho mạch, giúp cho mạch luôn hoạt động.
* **Khối vi điều khiển**: gồm 2 vi điều khiển là Arduino uno R3 và NodeMCU ESP8266, là phần để người dùng điều khiển, quản lý hệ thống.
* Vi điều khiển Arduino uno R3 có chức năng thu nhận dữ liệu từ cảm biến và điều khiển.
* Vi điều khiển NodeMCU ESP8266 có chức năng thu nhận giữ liệu từ module GPS NEO-6M.
* **Khối cảm biến** : Đo khoảng cách vật rồi gửi dữ liệu về khối vi điều khiển (Arduino uno R3 )
* **Khối báo động**: nhận tín hiệu từ khối vi điều khiển Arduino uno R3 và báo động ra còi.
* **Khối sim**: Báo về điện thoại khi thùng rác đầy.
* **Khối GPS**: Hiển thị vị trí của thùng rác
* **Khối hiển thị**: Hiển thị google map trên app blynk.
* **Khối relay**: Bật tắt đèn và hẹn giờ bật tắt đèn.

### 2.1.3 Nguyên lý hoạt động

Thùng rác thông minh tự động mở lắp khi có người đến gần vứt rác, nếu thùng rác đầy sẽ phát ra báo động tại chỗ và gửi tin nhắn về điện thoại. Thùng rác sẽ được định vị để quan sát được vị trí của thùng rác, thùng rác cũng sẽ được lắp thêm 1 bóng đèn bật tắt, hẹn giờ bật tắt để giúp người vứt rác có thể vứt rác dễ dàng hơn vào buối tối.

## **2.2. Linh kiện phần cứng**

### 2.2.1. Arduino uno R3

Arduino Uno R3 là một board mạch vi điều khiển được phát triển bởi Arduino.cc, một nền tảng điện tử mã nguồn mở chủ yếu dựa trên vi điều khiển AVR Atmega328P.



Hình 2.2: Arduino Uno R3

Phiên bản hiện tại của Arduino Uno R3 đi kèm với giao diện USB, 6 chân đầu vào analog, 14 cổng kỹ thuật số I / O được sử dụng để kết nối với các mạch điện tử, thiết bị bên ngoài. Trong đó có 14 cổng I / O, 6 chân đầu ra xung PWM cho phép các nhà thiết kế kiểm soát và điều khiển các thiết bị mạch điện tử ngoại vi một cách trực quan. Arduino Uno R3 được kết nối trực tiếp với máy tính thông qua USB để giao tiếp với phần mềm lập trình IDE, tương thích với Windows, MAC hoặc Linux Systems, tuy nhiên, Windows thích hợp hơn để sử dụng. Các ngôn ngữ lập trình như C và C ++ được sử dụng trong IDE. Ngoài USB, người dùng có thể dùng nguồn điện ngoài để cấp nguồn cho bo mạch.

Các bo mạch Arduino Uno khá giống với các bo mạch khác trong các loại Arduino về mặt sử dụng và chức năng, tuy nhiên các bo mạch Uno không đi kèm với chip điều khiển FTDI USB to Serial.

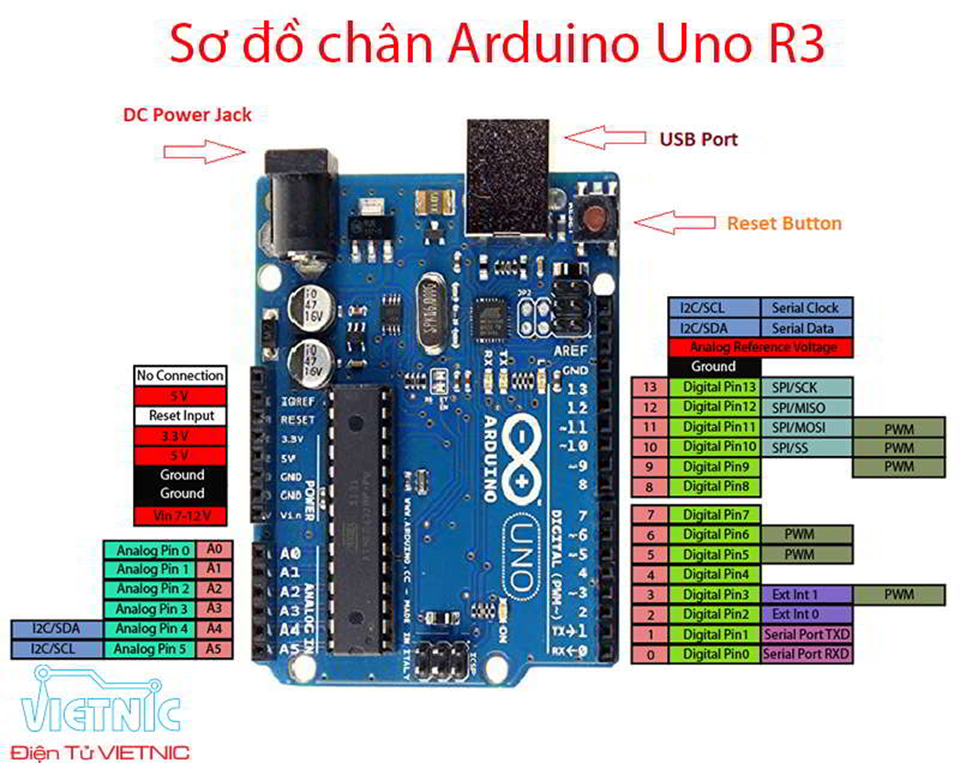
Có rất nhiều phiên bản bo mạch Uno, tuy nhiên, Arduino Nano V3 và Arduino Uno là những phiên bản chính thức nhất đi kèm với vi điều khiển Atmega328 8bit AVR Atmel trong đó bộ nhớ RAM là 32KB. Khi tính chất và chức năng của nhiệm vụ trở nên phức tạp, thẻ nhớ SD Mirco có thể được kết nối thêm vào Arduino để lưu trữ được nhiều thông tin hơn.

***Một vài thông số của Arduino UNO R3:***

Bảng 2. 1: Thông số kỹ thuật của Arduino Uno R3

|  |  |
| --- | --- |
| 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 | 1 KB (ATmega328) |

***Sơ đồ chân của Arduino UNO R3:***



Hình 2.3: Sơ đồ chân Arduino Uno R3

* 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.

*Các cổng vào/ra:*

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối). Một số chân digital có các chức năng đặc biệt như sau:

* 2 chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết
* Chân PWM (~): 3, 5, 6, 9, 10, và 11: cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 → 28-1 tương ứng với 0V → 5V) bằng hàm analogWrite(). Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
* Chân giao tiếp SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
* LED 13: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

Arduino UNO có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 210-1) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân AREF trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V → 2.5V với độ phân giải vẫn là 10bit.

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

### 2.2.2. Module NodeMCU ESP8266

Kit RF thu phát Wifi ESP8266 NodeMCU 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 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. Kit RF thu phát Wifi ESP8266 NodeMCU sử dụng chip nạp và giao tiếp UART mới và ổn định nhất là CP2102 có khả năng tự nhận Driver trên tất cả các hệ điều hành Window và Linux, đây là phiên bản nâng cấp từ các phiên bản sử dụng IC nạp giá rẻ CH340.



Hình 2.4: Module thu phát wifi NodeMCU ESP8266 CP2102

**Thông số kỹ thuật**

* Hỗ trợ Arduino IDE 1 và Arduino ESP8266
* Sử dụng module wifi ESP – 12E
* Nguồn vào: Cấp nguồn 5V và chương trình thông qua cổng USB
* Kích thước: 49 x 24.5 x 13mm
* IC chính: ESP8266 Wifi SoC - Phiên bản firmware: Node MCU
* Chip nạp và giao tiếp UART: CH340 - 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

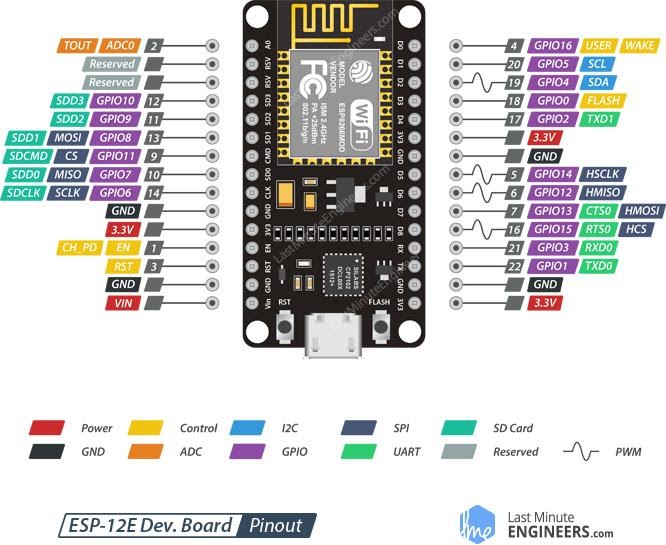
**Các thông số khác:**

* Tích hợp TCP / IP protocol stack
* Tích hợp TR switch, balun, LNA, power amplifier and matching network
* Tích hợp bộ nhân tần số, ổn áp, DCXO and power management units
* 25 dBm output power in 802.11b mode
* Power down leakage current of < 100A
* Integrated low power 32 - bit CPU could be used as application processor
* SDIO 1.1 / 2.0, SPI, UART + STBC, 1x1 MIMO, 2x1 MIMO
* A - MPDU & A - MSDU aggregation & 0.4ms guard interval
* Wake up and transmit packets in < 2ms
* Dòng tiêu thụ ở Standby Mode < 1,0mW (DTIM3)

**Các chẩn giao tiếp và các thông tin khác:**

* SDIO 2.0. SPI, UART
* Integrated RF switch, balun, 24dBm PA, DCXO and PMU
* Integrated RISC processor, on - chip memory and external memory
* MAC / baseband processors
* Quality of Service management
* 128 interface for high fidelity audio applications
* On - chip low - dropout linear regulators for all internal supplies
* Proprietary spurious - free clock generation architecture
* Intergrated WEB, TKIP, AES, and WAPI enginesv.

**Sơ đồ chân**



Hình 2.5: Sơ đồ chân module NodeMCU ESP8266 CP2102

* **GPIO được kết nối với Chip Flash**

GPIO6 đến GPIO11 thường được kết nối với chip flash trong bo mạch ESP8266. Vì vậy, những chân này không được khuyến khích sử dụng.

* **Chân được sử dụng trong khi khởi động**

ESP8266 có thể bị ngăn không cho khởi động nếu một số chân được kéo MỨC THẤP hoặc MỨC CAO. Danh sách sau đây cho thấy trạng thái của các chân khi khởi động:

* GPIO16: chân ở mức cao khi khởi động
* GPIO0: lỗi khởi động nếu kéo mức thấp
* GPIO2: chân ở mức cao khi khởi động, không khởi động được nếu kéo mức thấp
* GPIO15: lỗi khởi động nếu kéo mức cao
* GPIO3: chân ở mức cao khi khởi động
* GPIO1: chân ở mức cao khi khởi động, không khởi động được nếu kéo mức thấp
* GPIO10: chân ở mức cao khi khởi động
* GPIO9: chân ở mức cao khi khởi động
* **Chân mức cao khi khởi động**

Có một số chân xuất ra tín hiệu 3.3V khi ESP8266 khởi động. Điều này sẽ là vấn đề cần phải quan tâm nếu bạn có relay hoặc thiết bị ngoại vi khác được kết nối với các GPIO đó. Các GPIO sau xuất tín hiệu mức cao khi khởi động:

* GPIO16
* GPIO3
* GPIO1
* GPIO10
* GPIO9

Ngoài ra, các GPIO khác, ngoại trừ GPIO5 và GPIO4, có thể xuất ra tín hiệu điện áp thấp khi khởi động, có thể có vấn đề nếu chúng được kết nối với transistor hoặc relay.

* **Đầu vào analog**

ESP8266 chỉ hỗ trợ đọc analog trong một GPIO. GPIO đó được gọi là ADC0 và nó thường được đánh dấu trên màn lụa là A0.

Điện áp đầu vào tối đa của chân ADC0 là 0 đến 1V nếu bạn đang sử dụng chip trần ESP8266. Nếu bạn đang sử dụng bo phát triển như bộ ESP8266 12-E NodeMCU, thì dải điện áp đầu vào là 0 đến 3,3V vì bo này có bộ chia điện áp bên trong.

* **Đèn LED trên bo mạch**

Hầu hết các bo phát triển ESP8266 đều có đèn LED tích hợp. Đèn LED này thường được kết nối với GPIO2.

Đèn LED hoạt động với logic ngược. Gửi tín hiệu CAO để tắt và tín hiệu THẤP để bật.

* **Chân RST**

Khi chân RST được kéo THẤP, ESP8266 sẽ reset. Thao tác này cũng giống như nhấn nút reset trên bo mạch.

* **GPIO0**

Khi GPIO0 được kéo THẤP, nó sẽ đặt ESP8266 vào chế độ bộ nạp khởi động. Thao tác này cũng giống như nhấn nút FLASH / BOOT trên bo mạch.

* **GPIO16**

GPIO16 có thể sử dụng để đánh thức ESP8266 khỏi chế độ ngủ sâu. Để đánh thức ESP8266 khỏi chế độ ngủ sâu, GPIO16 phải được kết nối với chân RST.

* **I2C**

ESP8266 không có chân I2C phần cứng, nhưng nó có thể được triển khai trong phần mềm. Vì vậy, bạn có thể sử dụng bất kỳ GPIO nào làm I2C. Thông thường, các GPIO sau được sử dụng làm chân I2C:

* GPIO5: SCL
* GPIO4: SDA
* **SPI**

Các chân được sử dụng làm SPI trong ESP8266 là:

* GPIO12: MISO
* GPIO13: MOSI
* GPIO14: SCLK
* GPIO15: CS
* **Các chân PWM**

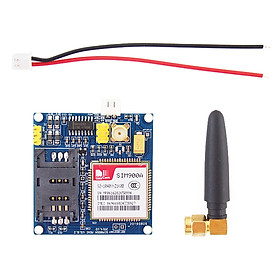
ESP8266 cho phép phần mềm PWM ở tất cả các chân I/O: GPIO0 đến GPIO16. Tín hiệu PWM trên ESP8266 có độ phân giải 10-bit.

* **Chân ngắt**

ESP8266 hỗ trợ chân ngắt trong bất kỳ GPIO nào, ngoại trừ GPIO16.

### 2.2.3. Module Sim 900A

[Module GSM GPRS Sim900A](https://nshopvn.com/product/module-gsm-gprs-sim900a/) có IC đệm (GSM sim900a module ) được thiết kế cho các ứng dụng cần độ bền và độ ổn định cao, mạch có kích thước nhỏ gọn nhưng vẫn giữ được các yếu tố cần thiết của thiết kế Sim900 cũ như: Mạch chuyển mức tín hiệu logic sử dụng Mosfet, IC giao tiếp RS323 MAX232, tụ ổn định nguồn đầu vào, khe sim chuẩn và các đèn led báo hiệu. Ngoài ra mạch còn đi kèm dây cáp nguồn và Anten GSM.

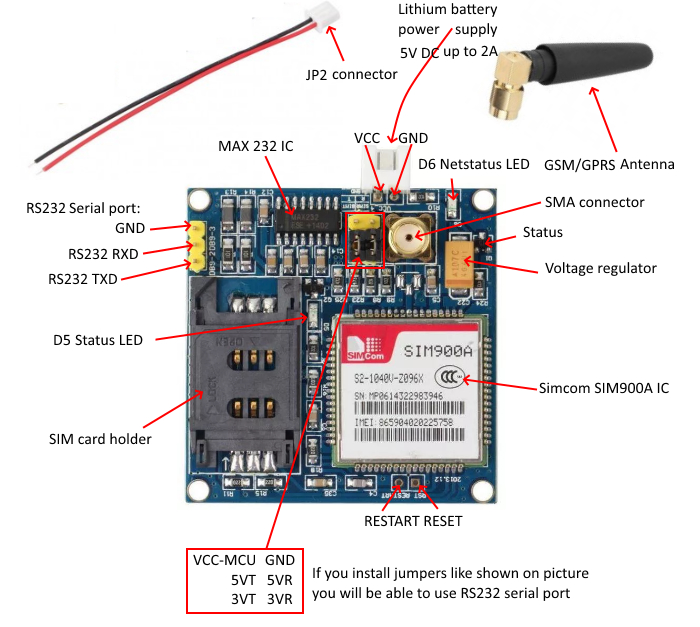


Hình 2.6: Module Sim900A

**Thông số kỹ thuật**

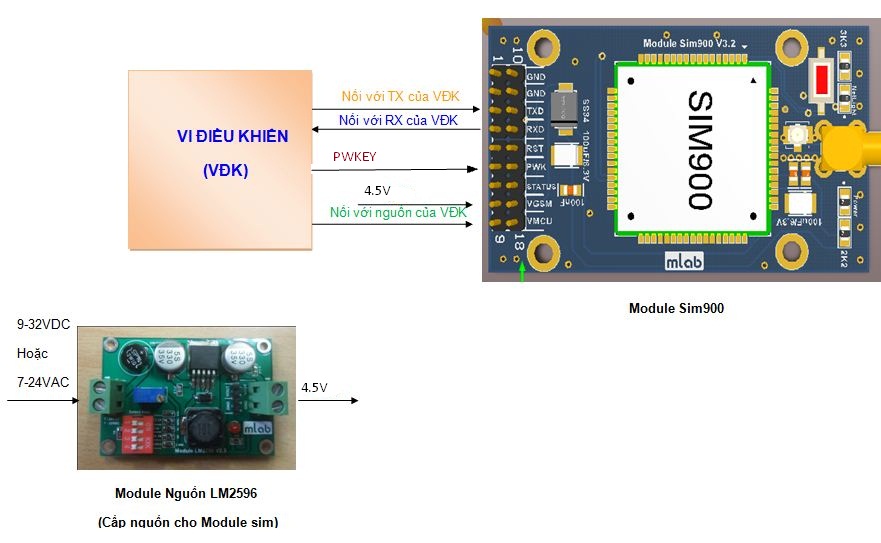
* Điện áp hoạt động: 4.7-5V
* Điện năng tiêu thụ thấp: 1.5mA (ở chế độ ngủ)
* Nhiệt độ hoạt động: -40 - 85 °C
* Điều khiển qua tập lệnh AT (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)
* Băng tần kép 900/ 1800 MHz
* GPRS multi-slot class 10/8
* GPRS mobile station class B
* Phù hợp với GSM giai đoạn 2/2+

**Sơ đô chân**



Hình 2.7: Sơ đồ chân Module Sim900A

**Sơ đồ đấu nối với vi điều khiển**



Hình 2.8: Sơ đồ đấu nối module Sim900A với vi điều khiển

### 2.2.4. Module GPS NEO-6M

- Xác định tọa độ (kinh tuyến, vĩ tuyến) hiện tại của module trên bề mặt trái đất với sai số nhỏ nhất < 1m.              
- Xác định thời gian quốc tế được cấp bởi đồng hồ nguyên tử trên vệ tinh gửi về. Từ đó bạn cũng có thể suy ra thời gian đồng hồ nơi ở của bạn theo tắc trừ múi giờ. Khỏi cần module RTC.   
- Chỉ cần 3 vệ tinh là bạn có thể xác định được tọa độ, chỉ cần 4 vệ tinh là bạn có thể xác định được độ cao hiện tại so với mực nước biển.  
- Có thể tính toán ra tốc độ di chuyển, hướng di chuyển của vật thể được gắn module GPS.    
- Giải các bài toán về tính toán giữa 2 điểm bất kì, tính diện tích ở một không gian cực kì rộng lớn.

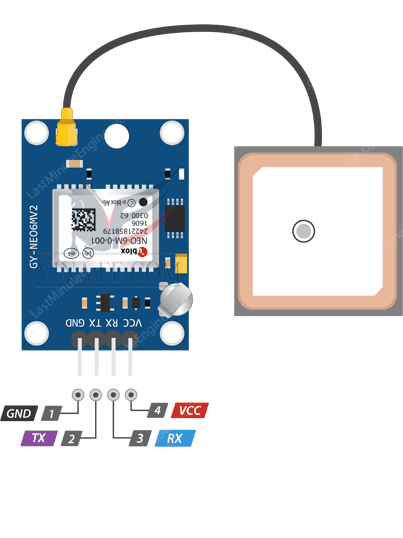


Hình 2.9: Module GPS NEO-6M

**Thông số kỹ thuật**

* Nguồn cung cấp: 3~5 VDC
* Mô-đun có ăng-ten bằng sứ, thu phát tín hiệu mạnh
* Có EEPROM lưu tham số cấu hình khi mất nguồn
* Có pin dự phòng lưu dữ liệu
* Có đèn LED hiển thị tín hiệu
* Kích thước ăng ten 12x12 mm
* Kích thước module: 23x30 mm
* Tốc độ baud mặc định: 9600
* Nhiệt độ hoạt động: -40°C ~ 85°C
* Trở kháng TXD / RXD: 510Ω

**Sơ đồ chân**

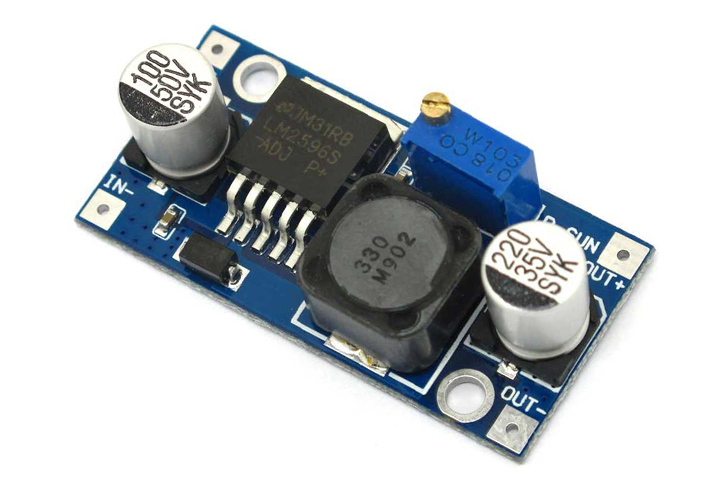


Hình 2.10: Sơ đồ chân Module GPS NEO-6M

* **VCC:** 3V-5V
* **TX:** truyền dữ liệu
* **RX:** nhận dữ liệu
* **GND:** nối đất

### 2.2.5. Mạch hạ áp DC-DC

Module dùng để hạ điện áp đầu vào , dải điện áp đầu ra lớn ( từ 1.23V đến 30V), nhỏ gọn, dễ dàng sử dụng, ứng dụng trong các mạch biến đổi nguồn DC - DC hay mạch cần giảm điện áp.

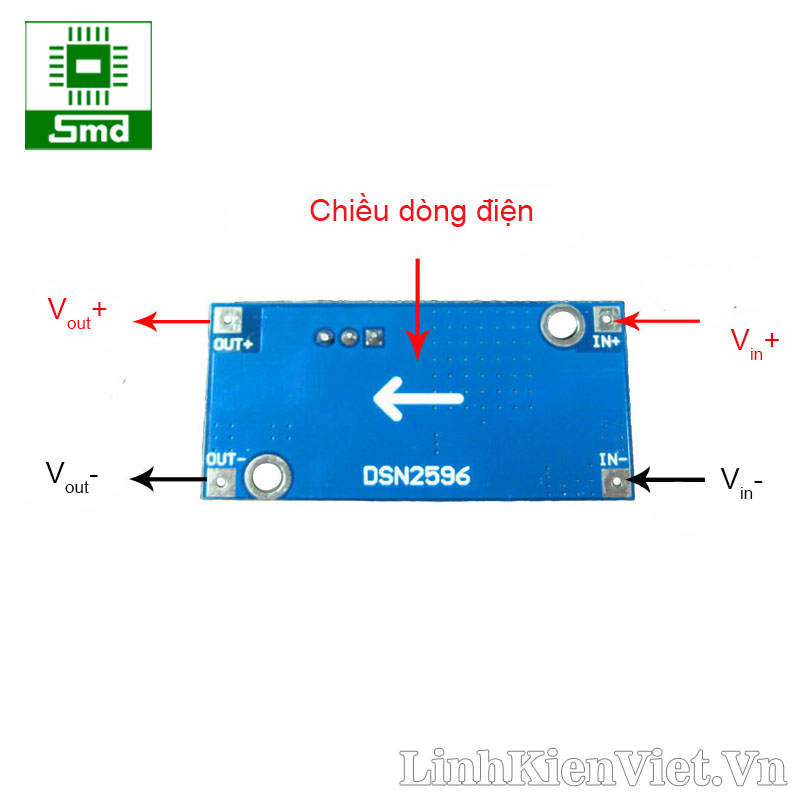


Hình 2.11: Module hạ áp LM2596

**Thông số kỹ thuật**

* Điện áp đầu vào: 4V-35V
* Điện áp đầu ra: 1.23V-30V
* Dòng đầu ra: 3A (max)
* Hiệu suất chuyển đổi: 92% (tối đa)
* Tần số hoạt động module hạ áp: 150kHz
* Nhiệt độ hoạt động: -40 ℃ đến + 85 ℃
* Kích thước: 23x14x8(mm)
* Sai số điện áp: ±2,5%

**Sơ đồ chân**



Hình 2.12: Sơ đồ chân module hạ áp LM2596

### 2.2.6. Còi báo

Buzzer hay còn gọi là còi chíp hoặc còi xung, là thiết bị phát ra âm thanh (tiếng bíp bíp) hay dùng trong các mạch điện tử.

Cấu tạo của Buzzer gồm 2 chân, chân dài là chân (+) và chân ngắn là chân (-). Trong quá trình sử dụng cần chú ý mắc đúng chân để tránh làm hỏng buzzer.

Ứng dụng này sử dụng Buzzer 5V, có khả năng phát ra âm thanh có tần số tối đa 2.5kHz.



Hình 2.13: Còi báo

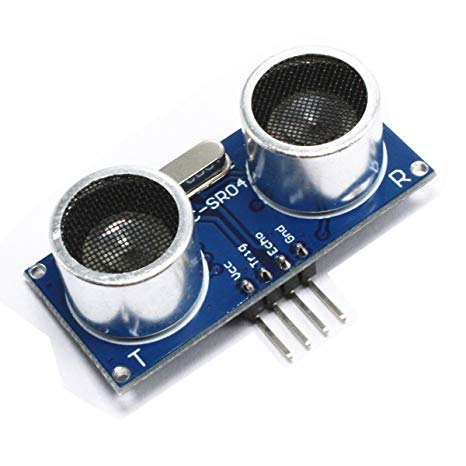
***Thống số kỹ thuật***

* Điện áp hoạt động: 3.5V – 5.5V.
* Dòng hoạt động: <25mA.
* Tần số âm thanh: 2500Hz.
* Nhiệt độ hoạt động: -20oC ~ 70o C.
* Nhiệt độ bảo quản: -30oC ~ 105o C

### 2.2.7. Cảm biến siêu âm HC-SR04

Cảm biến siêu âm HC-SR04 là một dạng cảm biến module. Cảm biến này thường chỉ là một bản mạch, hoạt động theo nguyên lý thu phát sóng siêu âm bởi 2 chiếc loa cao tần.

[Cảm biến siêu âm](https://huphaco.vn/cam-bien-sieu-am/) HC-SR04 thường được kết hợp với các bộ arduino, PIC, AVR,… để chạy một số ứng dụng như : phát hiện vật cản trên xe robot, đo khoảng cách vật,…

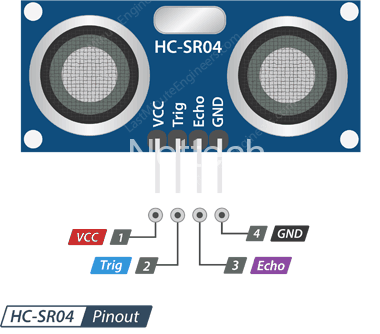


Hình 2.14: Cảm biến siêu âm HC-SR04

**Thông số kỹ thuật.**

* Model: HC-SR04
* Điện áp làm việc: 5VDC
* Dòng điện: 15mA
* Tần số: 40 KHZ
* Khoảng cách phát hiện: 2cm – 4m
* Tín hiệu đầu ra: Xung mức cao 5V, mức thấp 0V
* Góc cảm biến: Không quá 15 độ.
* Độ chính xác cao: Lên đến 3mm
* Chế độ kết nối: VCC / Trig (T-Trigger) / Echo (R-Receive) / GND

**Sơ đồ chân HC-SR04**



Hình 2.15: Sơ đồ chân cảm biến siêu âm HC-SR04

* **VCC** : 5V
* **trig** : chân điều khiển phát
* **echo** : chân nhận tín hiệu phản hồi
* **GND** : nối đất

***Sơ đồ nối chân giữa HC-SR04 với Arduino***

Bảng 2. 2: Bảng nối chân giữa HC-SR04 với Arduino Uno R3

|  |  |
| --- | --- |
| Vcc | 5V |
| Trig | Một chân Digital output |
| Echo | Một chân Digital input |
| GND | GND |

**Nguyên lý hoạt động của module HC-SR04**

Cảm biến siêu âm HC-SR04 có hai chế độ hoạt động.

***Chế độ 1: Sử dụng chân Trigger để kích hoạt phát sóng siêu âm và chân Echo để nhận tín hiệu phản hồi.***

Ở chế độ này, hai chân Trigger và Echo được sử dụng độc lập với nhau.Để sử dụng chế độ này chân Out để hở mạch (không kết nối).

Để kích hoạt cảm biến HC-SR04 phát ra sóng siêu âm, cần tạo 1 xung có độ rộng mức 1 tối thiểu là 10us trên chân Trigger. Sau đó cảm biến siêu âm HC-SR04 sẽ tạo ra 8 xung để phát ra sóng siêu âm. Sau đó khi sóng siêu âm được phát ra, chân Echo ngay lập tức được kéo lên mức cao. Nếu sóng siêu âm gặp vật cản và phản hồi ngược lại cảm biến siêu âm, chân Echo sẽ được đưa xuống mức 0. Nếu độ rộng xung đo được lớn hơn 30ms, nghĩa là không có vật cản, hoặc vật cản nằm ngoài khoảng cho phép của module.

***Chế độ 2: Sử dụng 1 chân Trigger để kích hoạt phát sóng siêu âm và nhận tín hiệu phản hồi.***

Ở chế độ này, chỉ có 1 chân Trigger được sử dụng đồng thời 2 chức năng kích hoạt phát sóng siêu âm và nhận tín hiệu phản hồi.Để sử dụng chế độ này, cần kết nối chân Out với 0V.

Tương tự chế độ 1, để kích hoạt phát sóng siêu âm, cần tạo xung có độ rộng tối thiểu 10us trên chân Trigger, sau đó, cảm biến siêu âm sẽ tạo ra 8 xung để phát sóng siêu âm. Sau khi sóng siêu âm được phát đi, chân Trigger được kéo lên mức 1, trong khoảng thời gian 100us-25ms, nếu có sóng siêu âm phản hồi, chân Trigger sẽ được đưa xuống mức 0. Nếu độ rộng xung đo được lớn hơn 30ms, nghĩa là không có vật cản, hoặc vật cản nằm ngoài khoảng cho phép của module.

**Tính toán khoảng cách**

Việc xác định khoảng cách sẽ được xác định bằng cách xác định độ rộng mức 1 của xung tín hiệu phản hồi trên chân Echo (ở chế độ 1) hoặc chân Trigger (ở chế độ 2).

Từ khoảng thời gian độ rộng mức 1, ta tính ra khoảng cách (Distance) từ vật cản đến cảm biến siêu âm HC-SR04 dựa vào công thức sau:

Gọi thời gian độ rộng xung đo được là **t(us).**

Thời gian đo độ rộng xung là thời gian từ lúc sóng truyền đi, gặp vật cản, và phản hồi lại. Nên thời gian từ cảm biến đến vật cản là: t/2(us).

Ta có vận tốc âm thanh trong không khí **= 343.2m/s = 0.03432cm/us.**

**Distance = 0.03432\*t(us)/2 (cm)**

**Distance ≃ t(us)/58 (cm)**

Cảm biến siêu âm có thể kích hoạt phát sóng siêu âm sau mỗi 50ms. Do vậy sau mỗi lần đo, chúng ta nên chờ 50ms rồi mới thực hiện lần đo tiếp theo.

### 2.2.8. Động cơ Servo SG90.

Động cơ Servo Tower SG90 là loại thường được dùng nhiều nhất trong các thiết kế robot hoặc dẫn hướng xe. Động cơ có lực kéo mạnh, khớp bánh răng làm hoàn toàn bằng kim loại nên có độ bền cao, động cơ được tích hợp sẵn driver điều khiển động cơ bên trong theo cơ chế phất xung quay góc nên rất dễ sử dụng.

Servo Tower Pro 9G là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ –



Hình 2.16: Động cơ servo

**Thông số kỹ thuật:**

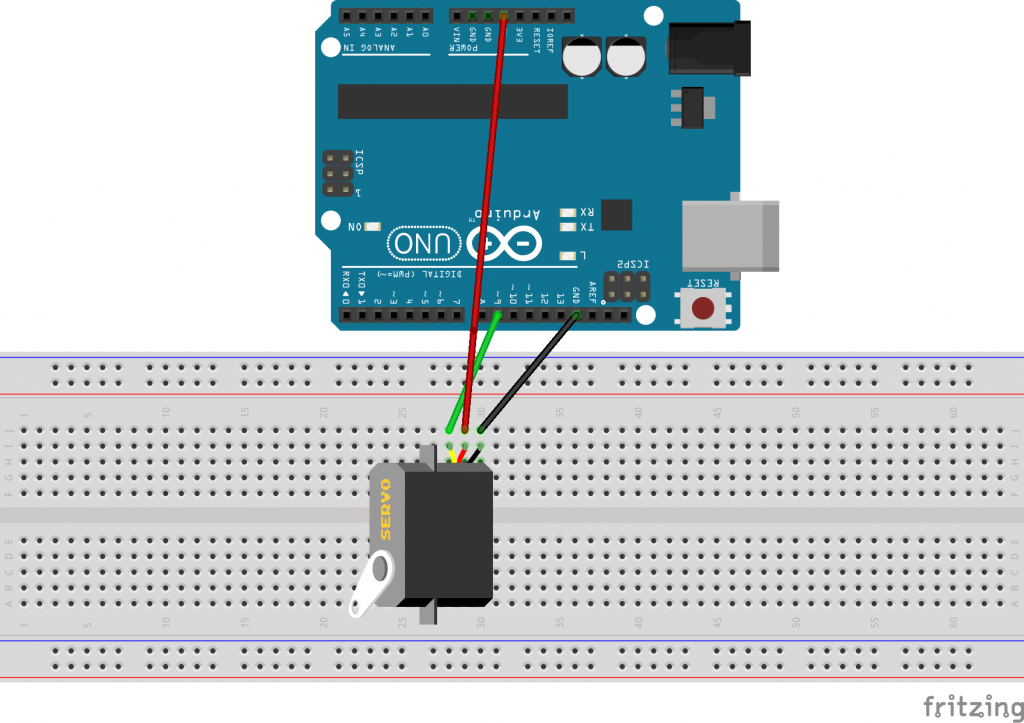
* Khối lượng: 9g
* Kích thước: 22.2x11.8.32 mm
* Momen xoắn: 1.8kg/cm
* Tốc độ hoạt động: 60 độ trong 0.1 giây
* Điện áp hoạt động: 4.8V(~5V)
* Nhiệt độ hoạt động: 0 ºC – 55 ºC

**Sơ đồ mắc dây:**

Bảng 2. 3: Bảng đấu nối giữa động cơ Servo với Arduino Uno R3

|  |  |
| --- | --- |
| **Arduino** | **Động cơ Servo** |
| 5V | Dây màu đỏ |
| GND | Dây màu đen |
| D6 | Dây màu vàng |

Kết nối dây màu đỏ với 5V, dây màu nâu với mass, dây màu cam với chân phát xung của vi điều khiển. Ở chân xung cấp một xung từ 1ms-2ms theo để điều khiển góc quay theo ý muốn.

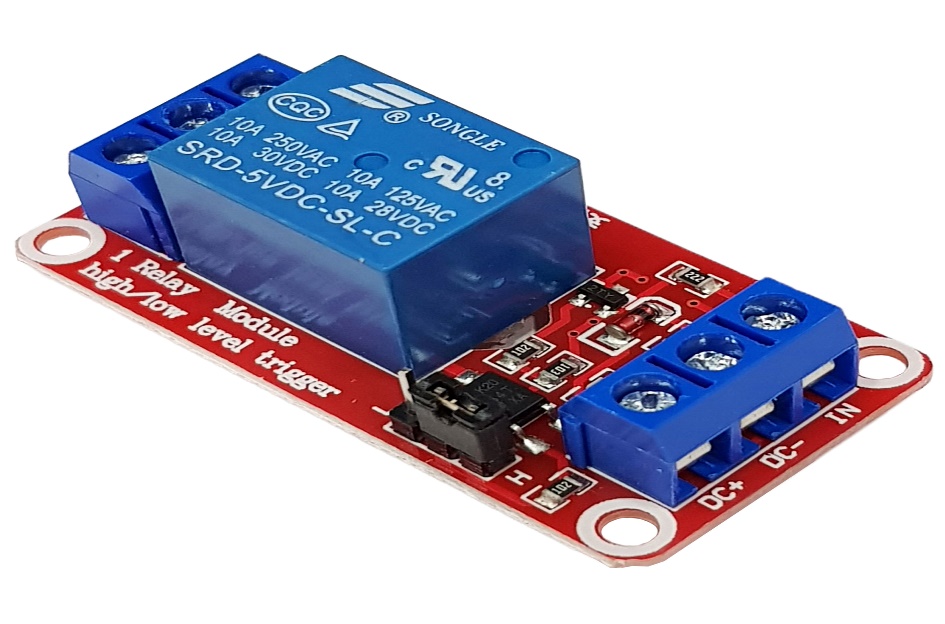


Hình 2.17: Sơ đồ đấu nối servo với vi điều khiển

### 2.2.9. Module Relay 12V 1 kênh.

Giới thiệu chung về Module Relay 12V 1 kênh:

* Module Relay 12V 1 Kênh được dùng như một công tắc điện, dùng để điều khiển các thiết bị công suất lớn (đèn, động cơ, ...)
* Module Relay 12V 1 Kênh gồm 1 rơ le hoạt động tại điện áp 12VDC chịu được hiệu điện thế lên đến 250VAC 10A. Module relay 1 kênh được thiết kế chắc chắn, khả năng cách điện tốt.



Hình 2.18: Module Relay 12V 1 kênh

**Thông số kỹ thuật:**

* Điện áp hoạt động: 12V
* Kích thước: 43mm x 17.3mm x 17mm (dài x rộng x cao)
* Trọng lượng: 15g

Đầu vào:

* Điện áp nuôi: 5VDC
* Tín hiệu vào điều khiển: 0V
* Tín hiệu là 0: thì Relay đóng
* Tín hiệu là: thì Relay mở

Đầu ra:

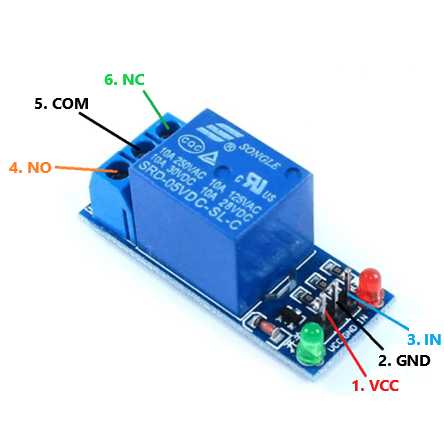
* Tiếp điểm relay 220V 10A (Lưu ý tiếp điểm, không phải điện áp ra)
* NC: Thường đóng
* NO: Thường mở
* COM: Chân chung

Ký hiệu nguồn:

* VCC, GND là nguồn nuôi Relay
* in là chân tín hiệu điều khiển

Các chân của Module Relay:

* DC +: Kết nối cực dương của nguồn điện.
* DC-: Kết nối cực âm của nguồn điện.
* IN: Theo cài đặt của người dùng, nó có thể ở mức cao hoặc thấp.
* NO: Cổng Thường Mở
* COM: Cổng chung rơle
* NC: Cổng Thường Đóng

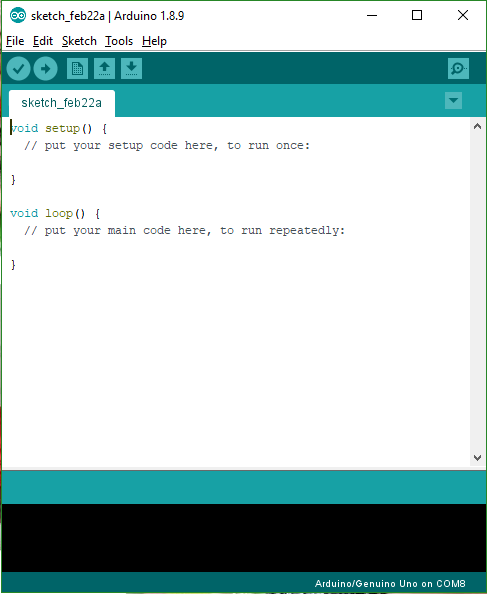


Hình 2.19: Sơ đồ chân module Relay 5V 1 kênh

## **2.3. Thiết kế phần mềm**

### 2.3.1. Phần mềm hệ thống Arduino IDE

Arduino IDE được viết tắt (Arduino Integrated Development Environment) là một trình soạn thảo văn bản, giúp bạn viết code để nạp vào bo mạch Arduino.

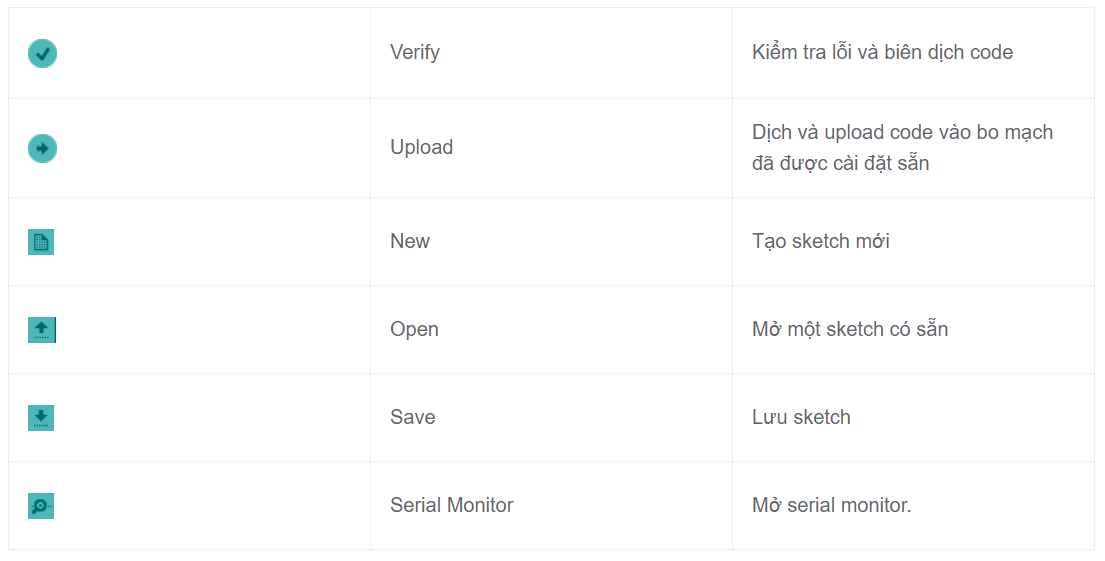


Hình 2.20 Giao diện phần mềm Arduino IDE

**Sketch**

Là một chương trình viết bởi Arduino IDE được gọi là sketch, sketch được lưu dưới định dạng .ino.

**Làm quen với giao diện Arduino IDE**



Hình 2.21 Chức năng thanh công cụ trong Arduino IDE

**Cổng Com**

Cổng nối tiếp (Serial port) là một cổng thông dụng trong các máy tính trong các máy tính truyền thống dùng kết nối các thiết bị ngoại vi với máy tính như: bàn phím, chuột điều khiển, modem, máy quét...Cổng nối tiếp còn có tên gọi khác như: Cổng COM, communication.

Ngày nay, do tốc độ truyền dữ liệu chậm hơn so với các cổng mới ra đời nên các cổng nối tiếp đang dần bị loại bỏ trong các chuẩn máy tính hiện nay, chúng được thay thế bằng các cổng có tốc độ nhanh hơn như: USB, FireWire

**Serial Monitor**

Serial Monitor là thành phần của Arduino IDE, giúp bo mạch và máy tính có thể gửi và nhận dữ liệu với nhay qua giao tiếp USB.

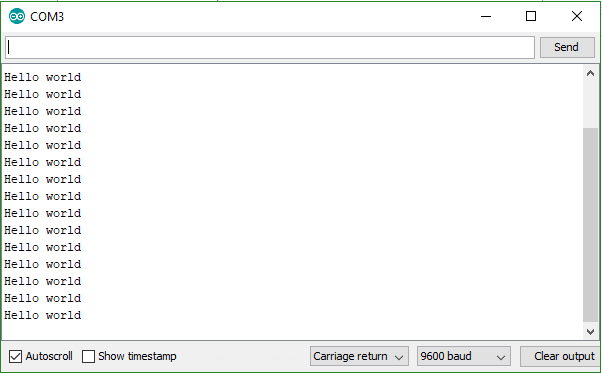
Để mở màn hình Serial Monitor, chúng ta chọn Tool > Serial Monitor.

**Hello World**

Để giao tiếp với máy tính, chúng ta cần phải thiết lập tốc độ kết nối qua cổng usb bằng hàm Serial.begin(). Sau đó dùng hàm Serial.print() để hiện thị nội dung ra màn hình Serial Monitor.



Hình 2.22: Ví dụ minh họa



Hình 2.23: Kết quả của ví dụ trên màn hình

### 2.3.2. App blynk

Blynk là một nền tảng với các ứng dụng iOS và Android để điều khiển Arduino, Raspberry Pi và các ứng dụng tương tự qua Internet.

Nó là một bảng điều khiển kỹ thuật số nhờ đó bạn có thể xây dựng giao diện đồ họa cho dự án của mình bằng cách kéo và thả các widget.

Việc thiết lập mọi thứ rất đơn giản và bạn sẽ bắt đầu sau chưa đầy 5 phút.

Blynk không bị ràng buộc với một số bo hoặc shield cụ thể. Thay vào đó, nó hỗ trợ phần cứng mà bạn lựa chọn. Cho dù Arduino hoặc Raspberry Pi của bạn được liên kết với Internet qua Wi-Fi, Ethernet hoặc chip ESP8266, Blynk sẽ giúp bạn online và sẵn sàng cho IoT

***Cách sử dụng app blynk***

a. Tạo tài khoản

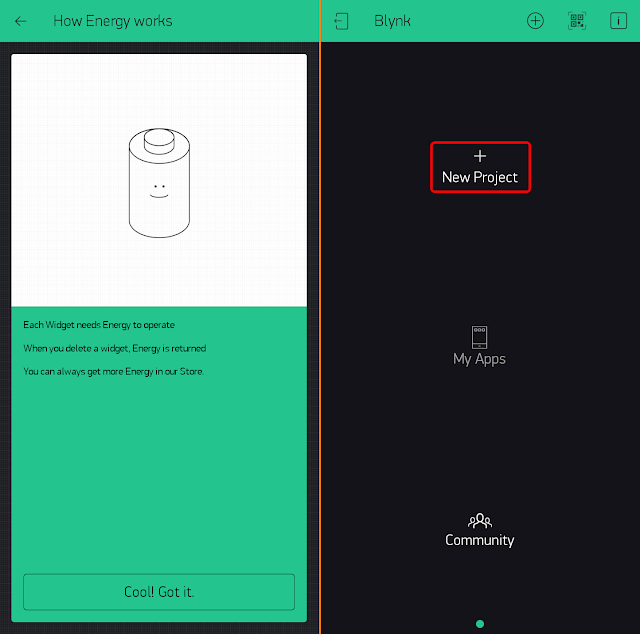
Sau khi tại ứng dụng về, tiếp theo cần tạo tài khoản để có thể sử dụng ứng dụng.



Hình 2.24: Tạo tài khoản app Blynk

b. Tạo project

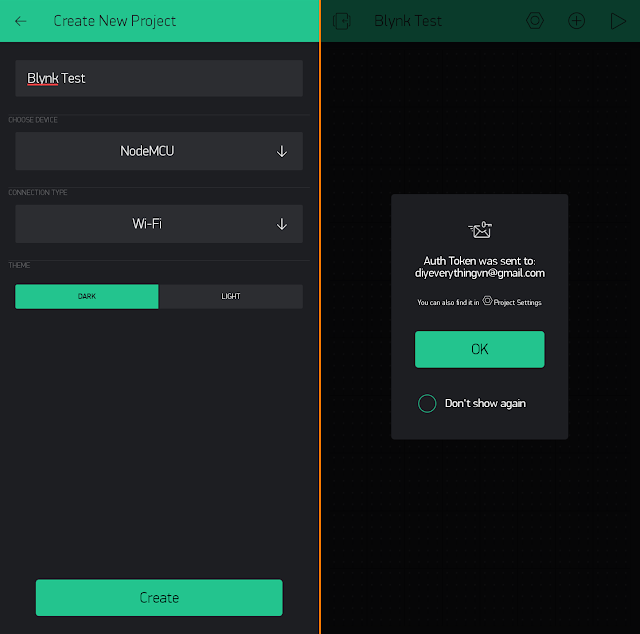
Chọn New Project để tạo 1 Project mới



Hình 2.25: Tạo project mới trên app blynk

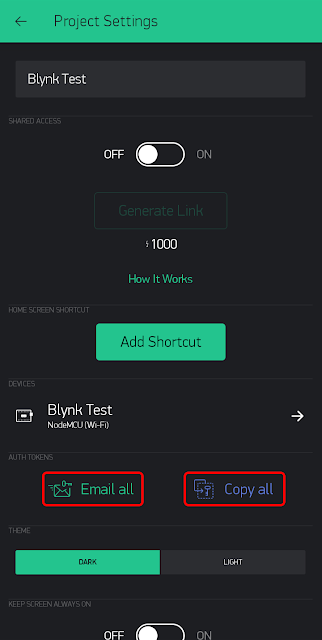
Tiếp theo Đặt tên, Chọn mạch điều khiển (như đã nói phía trên hướng dẫn này mình dùng mạch NodeMCU ESP8266, nên mình chọn NodeMCU), Chọn kiểu kết nối qua Wifi hay mạng dây

Sau khi chọn Create thì sẽ có mail gửi **Auth Token** gửi tới Email bạn vừa sử dụng để tạo tài khoản.



Hình 2.26: Đặt tên và chọn mạch điều khiển cho project

***\*Lưu ý:*** Auth Token bắt buộc phải có để có thể kết nối giữa mạch điều khiển tới server và bạn có thể điều khiển qua App. Nên nếu bạn không nhận được, hoặc không may xóa mail mà chưa kịp lưu thì bạn có thể vào phần cài đặt để sao chép lấy lại hoặc gửi lại mail Auth Token



Hình 2.27: Sao chép lấy lại hoặc gửi lại mail Auth Token

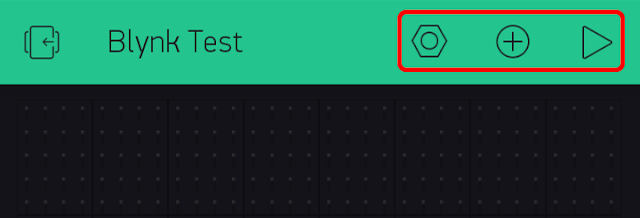
**c. Giao diện**

Phía trên bên phài, theo thứ tự từ trái qua phải là

• Cài đặt

• Add Widget

• Khởi chạy App điều khiển

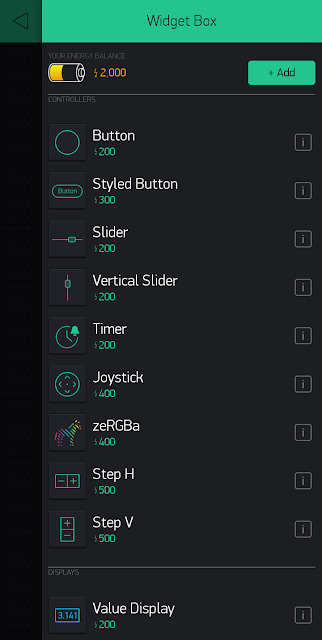


Hình 2.28: Giao diện app blynk

**d.  Add Widget**

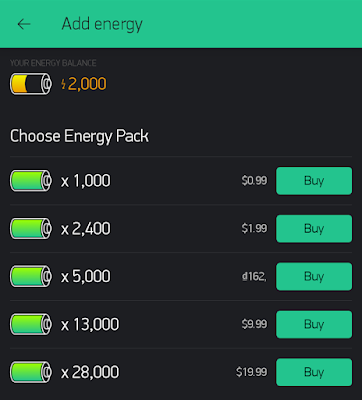
Thêm các Widget tính năng, điều khiển vào App

Mỗi Widget sẽ cần 1 lượng **Energy** tương ứng.



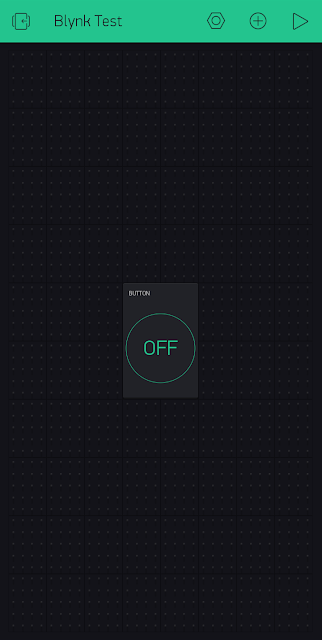
Hình 2.29: Add Widget

Với mặc định khi tạo tại khoản trên server mặc định của Blynk, khi đó lượng Energy được cho giới hạn là 2000, để có thêm Energy thì cần mua các gói. Điều này giúp bên phát triển của Blynk giới hạn số Widget có thể sử dụng và lấy kinh phí phát triển thêm (vì như các bạn đã biết thì Blynk là dự án mã nguồn mở được kêu gọi trên Kickstarter)



Hình 2.30: Add Energy

Ở đây ta sẽ thêm Widget Button để điều khiển LED bật/tắt ta có thể nhấn giữ Widget để thay đổi kích thước và di chuyển thay đổi vị trí các Widget



Hình 2.31: Widget Button

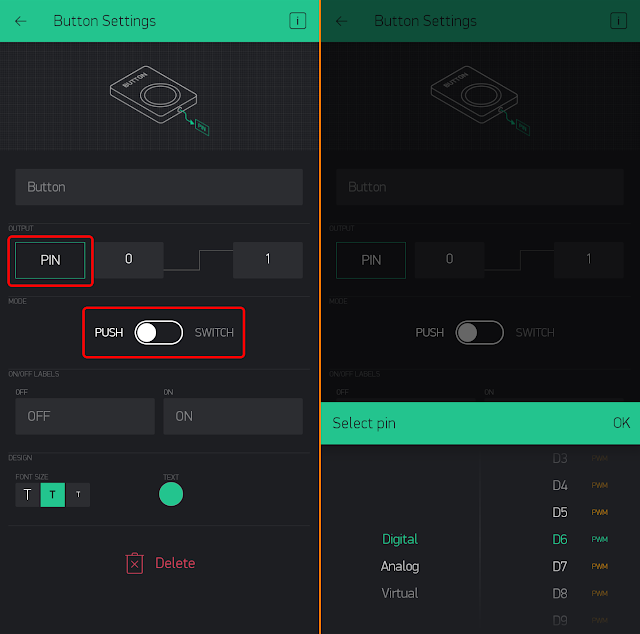
Sau đó ta chọn Widget để cài đặt cho Widget đó

Ở đây ta sử dụng chân **D6** trên mạch NodeMCU để điều khiển đèn LED

Ta thấy Button sẽ có 2 chế độ hoạt động

• Push: Có chức năng như các công tắc nhấn nhả

• Switch: Có chức năng như các công tắc thông thường, nhấn và giữ trạng thái



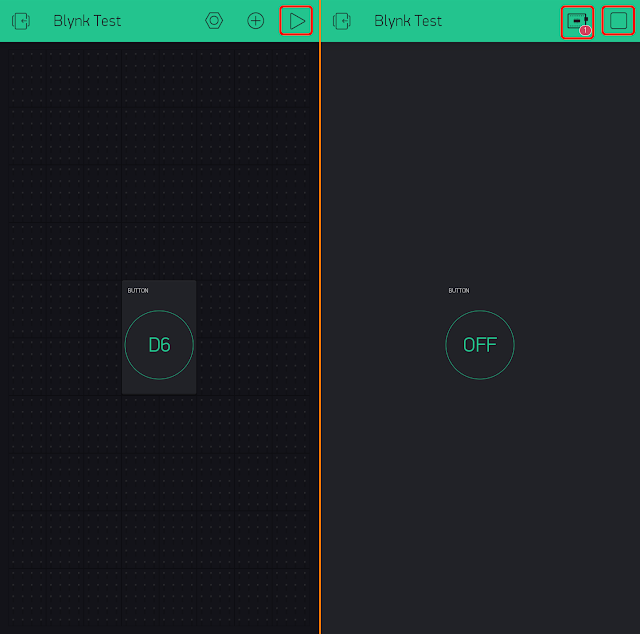
Hình 2.32: Cài đặt Widget Button

Sau khi cài đặt xong thì ta có thể chọn Run để chuyến sang chế độ điều khiển

Khi đó ta sẽ thấy 2 icon trên góc phải

• Theo dõi xem tình trạng kết nối của mạch điều khiển

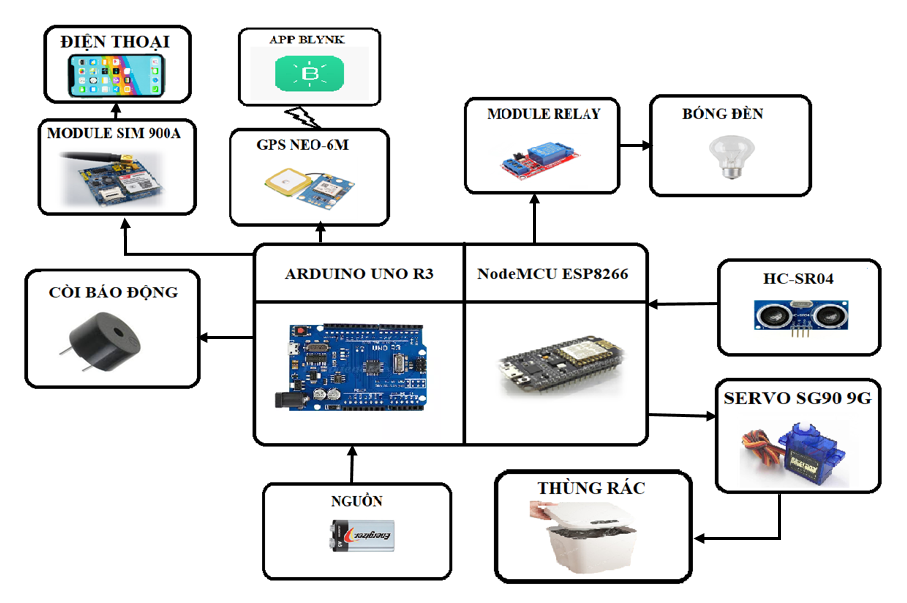
• Stop và quay lại chế độ cài đặt



Hình 2.33: Kết quả đạt được khi thiết lập các thông số

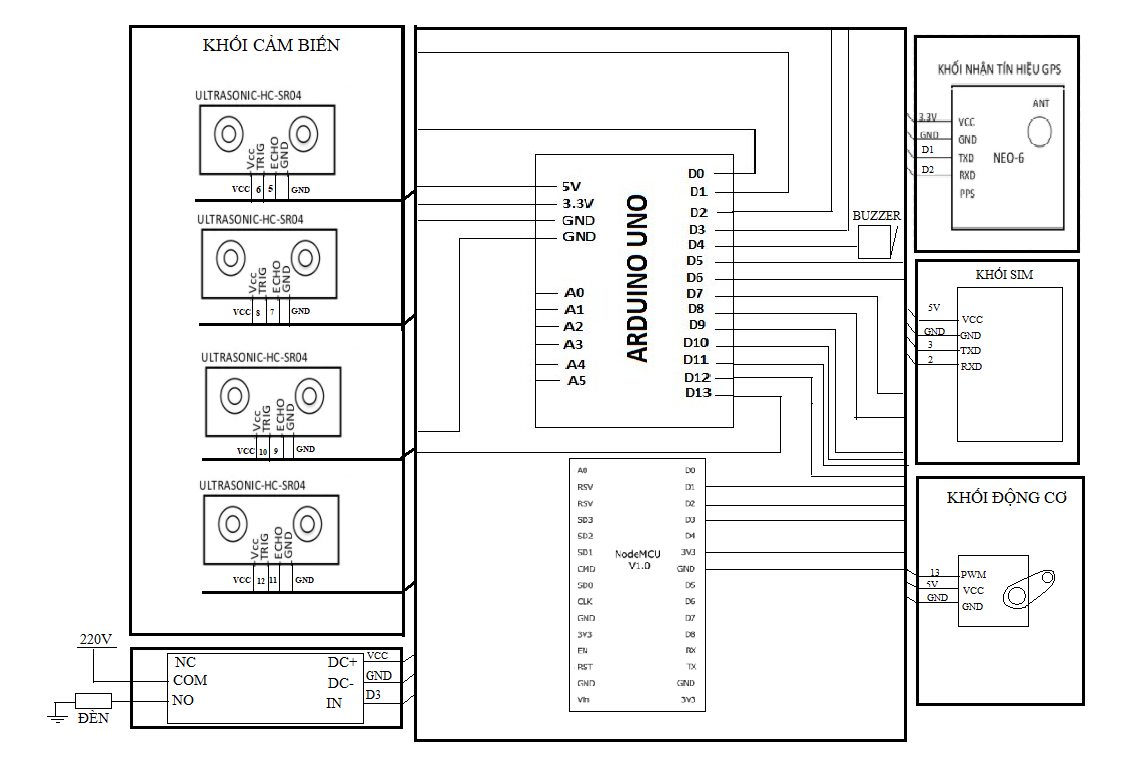
# **CHƯƠNG 3: THI CÔNG HỆ THỐNG**

## 3.1. Sơ đồ thiết bị



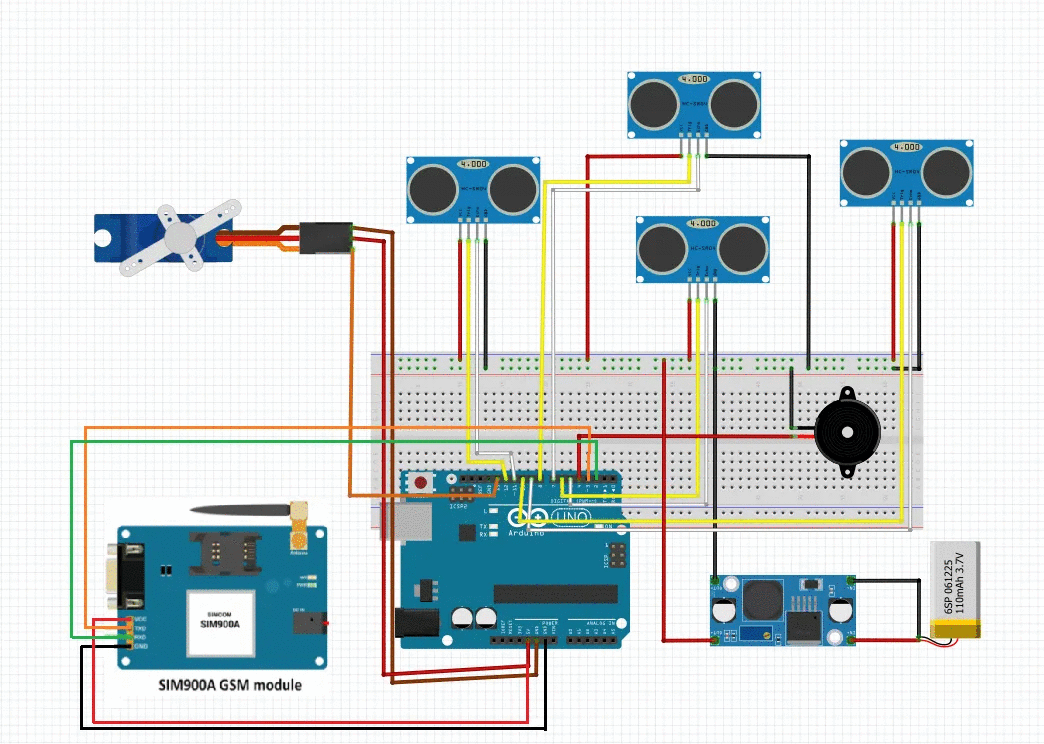
Hình 3. 1: Sơ đồ thiết bị

## 3.2. Sơ đồ nguyên lý

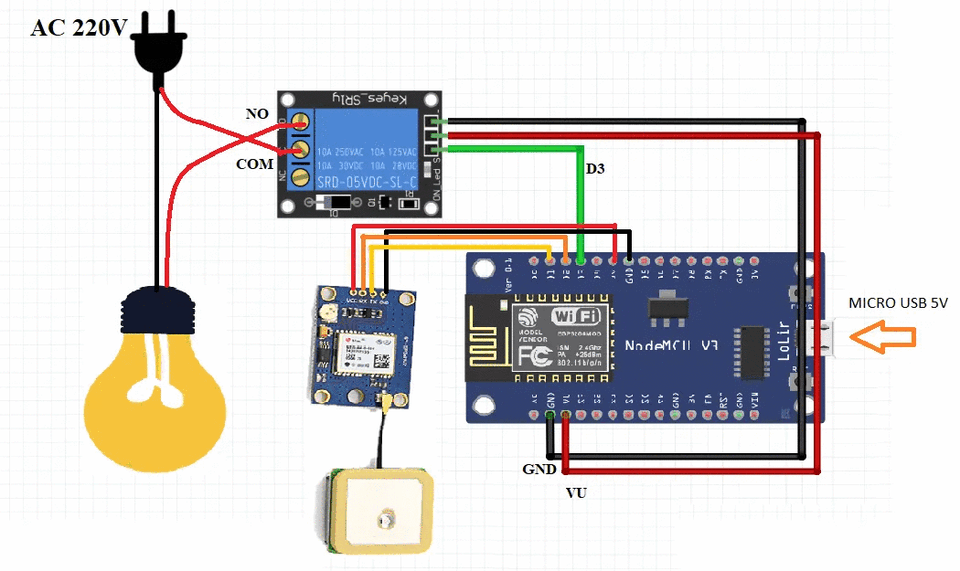


Hình 3. 2: Sơ đồ nguyên lý

## 3.3. Sơ đồ đấu nối



Hình 3. 3: Sơ đồ đi dây với vi điều khiển Arduino Uno R3



Hình 3. 4: Sơ đồ đi dây với vi điều khiển ESP8266

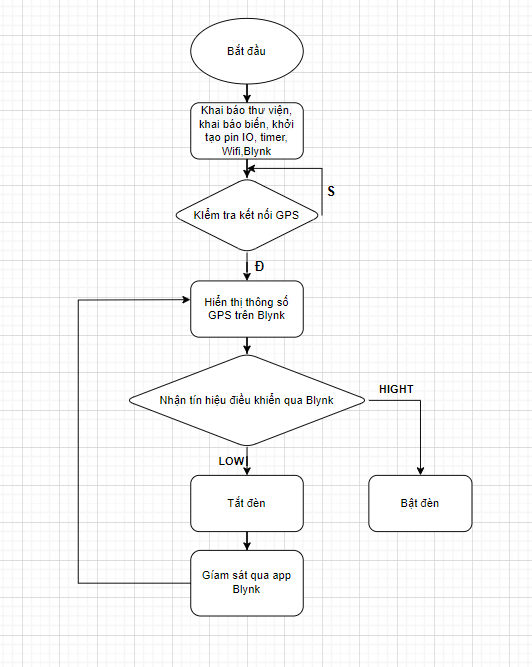
## 3.4. Lưu đồ thuật toán

***Lưu đồ thuật toán cho khối vi điều khiển Arduino UNO R3***



Hình 3. 5: Lưu đồ thuật toán

***Lưu đồ thuật toán cho khối vi điều khiển ESP8266***



Hình 3. 6: Lưu đồ thuật toán

## 3.5. Sản phẩm

a. Phần cứng

|  |  |
| --- | --- |
| C:\Users\Admin\Downloads\270286333_355593005965247_6955119417417087496_n.jpg | C:\Users\Admin\Downloads\b412c1d718f0d5ae8ce1.jpg |

*Hình 3. 7: Sản phẩm thực tế*

b. Phần mềm

|  |  |
| --- | --- |
| C:\Users\Admin\Downloads\271291365_696072841411131_1133203833448744184_n.png | C:\Users\Admin\Downloads\271433962_1119386972144205_4236391589621073693_n.jpg |

Hình 3. 8: Giao diện app hiển thị google map và điều khiển đèn

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**1. Kiến thức đạt được.**

Sau một khoảng thời gian tìm hiểu và nghiên cứu với sự hỗ trợ của cô Tô Thị Tuyết Nhung thì nhóm chúng em đã hoàn thành quyển đồ án và thi công mô hình theo những yêu cầu đặt ra ban đầu. Trong quá trình thực hiện, nhóm đã thu được những kết quả nhất định:

* Có kiến thức về hệ thống định vị toàn cầu GPS.
* Có kiến thức vững chắc về cảm biến siêu âm, cách lập trình Arduino Uno R3 và ESP8266.
* Thiết kế thành công mô hình khá cân đối, gọn gàng tuy nhiên chưa thực sự thẩm mỹ. Khi rác đầy sẽ có báo động tại chỗ và báo động đến chủ nhà hoặc nhân viên.

**2. Kết luận**

* **Ưu điểm**
* Thùng rác có thể tự động mở nắp khi cách tay người khoảng 10cm và tự động đóng lại sau 3 - 5 giây.
* Thùng rác sẽ báo động khi rác đầy.
* Thùng rác có gắn định vị để nhân viên có thể giám sát
* Có kết hợp với đèn để thuận tiện cho mọi người khi vứt rác vào buổi tối.
* **Nhược điểm**
* Mô hình hiện tại sử dụng không phù hợp với môi trường sử dụng ngoài trời mà phù hợp sử dụng trong văn phòng, trong nhà hơn.
* Thi thoảng gặp trục trặc về vấn đề cảm biến siêu âm không đo được khoảng cách mặc dù cảm biến hoạt động bình thường.
* Sai số vị trí vẫn còn tương đối lớn, do module GPS cập nhật vị trí còn chưa được chính xác.
* Thời lượng pin sử dụng chưa được lâu.

**3. Hướng phát triển**

* Sử dụng nguồn điện bằng pin năng lượng mặt trời
* Kết hợp với việc phân loại rác thải
* Kết hợp thêm các động cơ bánh xe để thùng rác có thể tự di chuyển
* Kết hợp thêm các cảm biến để thùng rác thông minh hơn…

# **TÀI LIỆU THAM KHẢO**

* 1. <https://thegioidienco.vn/cam-bien-do-khoang-cach.html?fbclid=IwAR0ZyNzM3jEiZud6H64DFQVoWG-yE5DBWbIRJ4jBKGqRFjcgQHp3eebKvLU>
  2. <https://baoanjsc.com.vn/du-an/cam-bien-do-khoang-cach-la-gi_2_70_31443_vn.aspx?fbclid=IwAR0jGS-Bh6aICTFu-iwbP2LaAzKVYFnuqjngjv-ceYuZZ5YQZUUH_cRnr_k>
  3. <https://cambiendo.vn/cam-bien-sieu-am-la-gi/>

# **PHỤ LỤC**

***Mã nguồn arduino***

|  |
| --- |
| #include <SoftwareSerial.h>  #include <Servo.h>  SoftwareSerial SIM900A**(**3**,** 2**);**  Servo myServo**;**  int servoPin **=** 13**;** //(D13)  int Buzzer **=** 4**;** // (D4)  const int trig\_1 **=** 12**;**  const int echo\_1 **=** 11**;**  const int trig\_2 **=** 10**;**  const int echo\_2 **=** 9**;**  const int trig\_3 **=** 8**;**  const int echo\_3 **=** 7**;**  const int trig\_4 **=** 6**;**  const int echo\_4 **=** 5**;**  void setup**()** **{**  Serial**.**begin**(**9600**);**  pinMode**(**Buzzer**,** OUTPUT**);**  myServo**.**attach**(**servoPin**);**    pinMode**(**trig\_1**,**OUTPUT**);**  pinMode**(**echo\_1**,**INPUT**);**  pinMode**(**trig\_2**,**OUTPUT**);**  pinMode**(**echo\_2**,**INPUT**);**  pinMode**(**trig\_3**,**OUTPUT**);**  pinMode**(**echo\_3**,**INPUT**);**  pinMode**(**trig\_4**,**OUTPUT**);**  pinMode**(**echo\_4**,**INPUT**);**  SIM900A**.**begin**(**9600**);**  Serial**.**println **(**"SIM900A Ready"**);**  **}**  void loop**()** **{**  //Cảm biến siêu âm 1  long duration\_1**,** distance\_1**;**  /\* Phát xung từ chân trig \*/  digitalWrite**(**trig\_1**,** LOW**);**  delayMicroseconds**(**2**);**  digitalWrite**(**trig\_1**,** HIGH**);**  delayMicroseconds**(**10**);**  digitalWrite**(**trig\_1**,** LOW**);**  /\* Tính toán thời gian \*/  duration\_1 **=** pulseIn**(**echo\_1**,** HIGH**);**  distance\_1 **=** **(**duration\_1**/**2**)** **/** 29.1**;**    //Cảm biến siêu âm 2  long duration\_2**,** distance\_2**;**  digitalWrite**(**trig\_2**,** LOW**);**  delayMicroseconds**(**2**);**  digitalWrite**(**trig\_2**,** HIGH**);**  delayMicroseconds**(**10**);**  digitalWrite**(**trig\_2**,** LOW**);**  duration\_2 **=** pulseIn**(**echo\_2**,** HIGH**);**  distance\_2 **=** **(**duration\_2**/**2**)** **/** 29.1**;**    //Cảm biến siêu âm 3  long duration\_3**,** distance\_3**;**  digitalWrite**(**trig\_3**,** LOW**);**  delayMicroseconds**(**2**);**  digitalWrite**(**trig\_3**,** HIGH**);**  delayMicroseconds**(**10**);**  digitalWrite**(**trig\_3**,** LOW**);**  duration\_3 **=** pulseIn**(**echo\_3**,** HIGH**);**  distance\_3 **=** **(**duration\_3**/**2**)** **/** 29.1**;**  //Cảm biến siêu âm 4( nắp thùng rác để báo đầy )  long duration\_4**,** distance\_4**;**  digitalWrite**(**trig\_4**,** LOW**);**  delayMicroseconds**(**2**);**  digitalWrite**(**trig\_4**,** HIGH**);**  delayMicroseconds**(**10**);**  digitalWrite**(**trig\_4**,** LOW**);**  duration\_4 **=** pulseIn**(**echo\_4**,** HIGH**);**  distance\_4 **=** **(**duration\_4**/**2**)** **/** 29.1**;**  **if** **((**distance\_1 **>=** 100 **||** distance\_1 **<=** 0**)** **^** **(**distance\_2 **>=** 100 **||** distance\_2 **<=** 0**)** **^** **(**distance\_3 **>=** 100 **||** distance\_3 **<=** 0**))**  **{**  Serial**.**println**(**"Ngoài phạm vi"**);**  myServo**.**write**(**0**);**  **if** **(**distance\_4 **>=** 6 **||** distance\_4 **<=** 0**)**  **{**  Serial**.**println**(**"khoang cach cam bien 4: "**);**  Serial**.**print**(**distance\_4**);**  Serial**.**println**(**" cm"**);**  digitalWrite**(**Buzzer**,** LOW**);**  delay**(**500**);**  **}**  **else** **if** **(**distance\_4 **<** 6 **||** distance\_4 **>** 0**)**  **{**  Serial**.**println**(**" TRASH FULL"**);**    digitalWrite**(**Buzzer**,** HIGH**);**  delay**(**5000**);**  SendMessage**();**    **}**  **}**  **else**  **{**  /\* In kết quả ra Serial Monitor \*/  Serial**.**println**(**"khoang cach cam bien 1: "**);**  Serial**.**print**(**distance\_1**);**  Serial**.**println**(**" cm"**);**  Serial**.**println**(**"khoang cach cam bien 2: "**);**  Serial**.**print**(**distance\_2**);**  Serial**.**println**(**" cm"**);**  Serial**.**println**(**"khoang cach cam bien 3: "**);**  Serial**.**print**(**distance\_3**);**  Serial**.**println**(**" cm"**);**  myServo**.**write**(**90**);** //Mở thùng rác  delay**(**5000**);** //Delay 5s  **}**  delay**(**500**);**    **}**  /\* Nhắn tin báo về điện thoại khi rác đầy.\*/  void SendMessage**()**  **{**  Serial**.**println **(**"Sending Message"**);**  SIM900A**.**println**(**"AT+CMGF=1"**);** //Sets the GSM Module in Text Mode  delay**(**1000**);**  Serial**.**println **(**"Set SMS Number"**);**  SIM900A**.**println**(**"AT+CMGS=\"+84868199092\"\r"**);** //Mobile phone number to send message  delay**(**1000**);**  Serial**.**println **(**"Set SMS Content"**);**  SIM900A**.**println**(**"Trash bin full"**);**// Messsage content  delay**(**100**);**  Serial**.**println **(**"Finish"**);**  SIM900A**.**println**((**char**)**26**);**// ASCII code of CTRL+Z  delay**(**1000**);**  **}**  void updateSerial**()**  **{**  delay**(**500**);**  **while** **(**Serial**.**available**())**  **{**  SIM900A**.**write**(**Serial**.**read**());**//Forward what Serial received to Software Serial Port  **}**  **while(**SIM900A**.**available**())**  **{**  Serial**.**write**(**SIM900A **.**read**());**//Forward what Software Serial received to Serial Port  **}**  **}** |

***Mã nguồn ESP8266***

|  |
| --- |
| #include <SoftwareSerial.h>  #define BLYNK\_PRINT Serial  //Install the following Libraries  #include <TinyGPS++.h>  #include <ESP8266WiFi.h>  #include <BlynkSimpleEsp8266.h>  //GPS RX to D1 & GPS TX to D2 and Serial Connection  const int RXPin **=** 4**,** TXPin **=** 5**;** // RX = D2; TX = D1  const uint32\_t GPSBaud **=** 9600**;**  SoftwareSerial gps\_module**(**RXPin**,** TXPin**);**  TinyGPSPlus gps**;**  WidgetMap myMap**(**V0**);** //V0 - virtual pin for Map  WidgetLED LedConnect**(**V5**);**  WidgetLED StatusD3**(**V3**);**  BlynkTimer timer**;**  //Variable to store the speed, no. of satellites, direction  float gps\_speed**;**  float no\_of\_satellites**;**  String satellite\_orientation**;**  char auth**[]** **=** "5gD3yF6-CCdqJ2iJ2TXlxCWNiaN52LDL"**;**  char ssid**[]** **=** "Android"**;**  char pass**[]** **=** "khongcomk"**;**  //unsigned int move\_index;  unsigned int move\_index **=** 1**;**    void setup**()**  **{**  Serial**.**begin**(**115200**);**  Serial**.**println**();**  gps\_module**.**begin**(**GPSBaud**);**  Blynk**.**begin**(**auth**,** ssid**,** pass**);**  timer**.**setInterval**(**5000L**,** checkGPS**);**    timer**.**setInterval**(**1000L**,** blinkLedWidget**);**  pinMode**(**D3**,**OUTPUT**);**  digitalWrite**(**D3**,**HIGH**);**  **}**  void checkGPS**(){**  **if** **(**gps**.**charsProcessed**()** **<** 10**)**  **{**  Serial**.**println**(**F**(**"No GPS detected: check wiring."**));**  Blynk**.**virtualWrite**(**V4**,** "GPS ERROR"**);**  **}**  **}**  BLYNK\_WRITE**(**V4**)**  **{**  int pinValue **=** param**.**asInt**();**  digitalWrite**(**D3**,**pinValue**);**  **if** **(!**digitalRead**(**D3**))** **{**  StatusD3**.**off**();**  **}** **else** **{**  StatusD3**.**on**();**  **}**  **}**  void blinkLedWidget**()**  **{**  **if** **(**LedConnect**.**getValue**())** **{**  LedConnect**.**on**();**  **}** **else** **{**  LedConnect**.**off**();**  **}**  **}**  void loop**(){**    **while** **(**gps\_module**.**available**()** **>** 0**)**  **{**  //displays information every time a new sentence is correctly encoded.  **if** **(**gps**.**encode**(**gps\_module**.**read**()))**  displayInfo**();**  **}**    Blynk**.**run**();**  timer**.**run**();**    **}**  void displayInfo**()**  **{**  **if** **(**gps**.**location**.**isValid**())**  **{**  //Storing the Latitude. and Longitude  float latitude **=** **(**gps**.**location**.**lat**());**  float longitude **=** **(**gps**.**location**.**lng**());**    //Send to Serial Monitor for Debugging  Serial**.**print**(**"LAT: "**);**  Serial**.**println**(**latitude**,** 6**);** // float to x decimal places  Serial**.**print**(**"LONG: "**);**  Serial**.**println**(**longitude**,** 6**);**    Blynk**.**virtualWrite**(**V1**,** String**(**latitude**,** 6**));**  Blynk**.**virtualWrite**(**V2**,** String**(**longitude**,** 6**));**  myMap**.**location**(**move\_index**,** latitude**,** longitude**,** "GPS\_Location"**);**    //get speed  gps\_speed **=** gps**.**speed**.**kmph**();**  Blynk**.**virtualWrite**(**V6**,** gps\_speed**);**    //get number of satellites  no\_of\_satellites **=** gps**.**satellites**.**value**();**  Blynk**.**virtualWrite**(**V7**,** no\_of\_satellites**);**  // get the satellite orientation/direction  satellite\_orientation **=** TinyGPSPlus**::**cardinal**(**gps**.**course**.**value**());**  Blynk**.**virtualWrite**(**V8**,** satellite\_orientation**);**  **}**    Serial**.**println**();**  **}** |