**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**HỌ VÀ TÊN:** Công Phương Đông

**KHÓA:** 11

**HỆ ĐÀO TẠO:** Đại học

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH:** Hệ thống thông tin

**TÌM HIỂU LẬP TRÌNH NHÚNG VÀ**

**PHÁT TRIỂN ỨNG DỤNG IoT**

**Hà Nội - Năm 2016**

**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**HỌ VÀ TÊN:** Công Phương Đông

**KHÓA:** 11

**HỆ ĐÀO TẠO:** Đại học

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH**: Hệ thống thông tin

**MÃ SỐ**: 001

**TÌM HIỂU LẬP TRÌNH NHÚNG VÀ**

**PHÁT TRIỂN HỆ THỐNG IoT**

**Cán bộ hướng dẫn:**

**Họ và tên:** Nguyễn Trung Thành

**Cấp bậc:**

**Học vị:** Tiến sĩ

**Hà Nội – Năm 2016**

|  |  |
| --- | --- |
| BỘ QUỐC PHÒNG  **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **KHOA: Công nghệ thông tin**  **Phê chuẩn**  Ngày …. tháng …. năm ……  **CHỦ NHIỆM KHOA** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **ĐỘC LẬP – TỰ DO – HẠNH PHÚC**  Độ mật: ……………..  Số: …………………. |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ và tên: Công Phương Đông Lớp: Tin học 11A Khóa: 11

Ngành: Công nghệ thông tin Chuyên ngành: Hệ thống thông tin

1. Tên đề tài:

Tìm hiểu lập trình nhúng và phát triển hệ thống IoT

2. Các số liệu ban đầu:

3. Nội dung bản thuyết minh:

4. Số lượng, nội dung các bản vẽ (ghi rõ loại, kích thước và cách thực hiện các bản vẽ) và các sản phẩm cụ thể (nếu có):

5. Cán bộ hướng dẫn

\* Họ và tên: Nguyễn Trung Thành

\* Cấp bậc:

\* Chức vụ:

\* Đơn vị:

\* Kiểu hướng dẫn:

|  |  |
| --- | --- |
| 🗹 Toàn bộ | 🞎 Từng phần |

\* Ghi chú:

\* Bảng giải thích các từ viết tắt

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Viết đầy đủ |
| 1 | IoT | Internet of things |

**MỤC LỤC**

[LỜI MỞ ĐẦU 6](#_Toc481426469)

[CHƯƠNG I: TỔNG QUAN 7](#_Toc481426470)

[1. Các mạch nhúng 7](#_Toc481426471)

[1.1. Mạch Arduino 7](#_Toc481426472)

[1.2. Module Sim808 13](#_Toc481426473)

[2. Các công nghệ liên quan 15](#_Toc481426474)

[2.1. Internet of things 15](#_Toc481426475)

[2.2. GPRS 16](#_Toc481426476)

[2.3. GPS 16](#_Toc481426477)

[2.4. Amazon Webservice 19](#_Toc481426478)

[CHƯƠNG II: MÔ TẢ HỆ THỐNG 20](#_Toc481426479)

[1. Nhiệm vụ cơ bản 20](#_Toc481426480)

[2. Mô tả hệ thống 20](#_Toc481426481)

[CHƯƠNG III: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 23](#_Toc481426482)

[1. Phân tích chức năng nghiệp vụ 23](#_Toc481426483)

[1.1. Xác định các chức năng nghiệp vụ 23](#_Toc481426484)

[1.2. Sơ đồ use case 23](#_Toc481426485)

[2. Phân tích dữ liệu nghiệp vụ 23](#_Toc481426486)

[2.1. Xác định kiểu dữ liệu ban đầu 23](#_Toc481426487)

[2.2. Mô hình liên kết thực thể 24](#_Toc481426488)

[2.3. Mô hình quan hệ 27](#_Toc481426489)

[2.4. Đặc tả dữ liệu 30](#_Toc481426490)

[3. Thiết kế giao diện 31](#_Toc481426491)

[3.1. Giao diện quản trị 31](#_Toc481426492)

[3.2. Giao diện người dùng 31](#_Toc481426493)

[CHƯƠNG IV: TRIỂN KHAI PHẦN CỨNG 31](#_Toc481426494)

[1. Triển khai phần cứng 31](#_Toc481426495)

[1.1. Lập trình với Module Sim808 31](#_Toc481426496)

[1.2. Kết hợp Module Sim808 với Arduino UNO 33](#_Toc481426497)

[2. Triển khai server 35](#_Toc481426498)

[CHƯƠNG IV: KẾT LUẬN 36](#_Toc481426499)

# **LỜI MỞ ĐẦU**

Internet of things hay còn được gọi là vạn vật kết nối, đang là một xu hướng mới đang dẫn đầu trong cuộc cách mạng về khoa học kỹ thuật. Thời đại IoT đang tăng tốc và sẽ thay đổi cách chúng ta làm việc, sinh hoạt và giải trí hằng ngày. Với những thiết bị IoT, chúng ta không cần phải mất công đi ra chỗ công tắc đèn, quạt … chỉ để bật chúng lên, thay vào đó, chúng ta có thể chỉ cần dùng một chiếc điện thoại thông minh và điều khiển mọi vật trong nhà.

Và để thực hiện việc IoT hóa các đồ vật xung quanh chúng ta thì chúng ta cần đến những hệ thống nhúng để có thể can thiệp vào cách thức vận hành của chúng. Sức mạnh của IoT có bị hạn chế hay không hoàn toàn phụ thuộc vào các mạch nhúng bên trong các thiết bị trong hệ thống IoT. Chính vì thế mà lập trình nhúng và IoT sẽ đi đôi với nhau, không thể tách rời.

Với đề tài này, em muốn mình có thể tạo ra được một hệ thống giúp người dùng có thể giám sát được các phương tiện của mình thông qua các thông tin như: vị trí hiện tại của xe, lịch sử di chuyển của xe.

Với kiến thức còn hạn chế, tuy đã cố gắng nỗ lực hết mình song đề tài của em vẫn không thể tránh khỏi những thiếu sót. Em mong nhận được những đóng góp về thiếu sót trong đề tài của em từ các thầy, cô và các bạn để đề tài có thể được hoàn thiện hơn nữa.

*Em xin chân thành cảm ơn!!!*

# **CHƯƠNG I: TỔNG QUAN**

## 1. Các mạch nhúng

### 1.1. Mạch Arduino

1.1.1 Lịch sử ra đời

+ Arduino được khởi động vào năm 2005 như là một dự án dành cho sinh viên trại Interaction Design Institute Ivrea (Viện thiết kế tương tác Ivrea) tại Ivrea, Italy. Vào thời điểm đó các sinh viên sử dụng một "BASIC Stamp" (con tem Cơ Bản) có giá khoảng $100, xem như giá dành cho sinh viên. Massimo Banzi, một trong những người sáng lập, giảng dạy tại Ivrea. Cái tên "Arduino" đến từ một quán bar tại Ivrea, nơi một vài nhà sáng lập của dự án này thường xuyên gặp mặt. Bản thân quán bar này có được lấy tên là Arduino, Bá tước của Ivrea, và là vua của Italy từ năm 1002 đến 1014.

+ Lý thuyết phần cứng được đóng góp bởi một sinh viên người Colombia tên là Hernando Barragan. Sau khi nền tảng Wiring hoàn thành, các nhà nghiên cứu đã làm việc với nhau để giúp nó nhẹ hơn, rẻ hơn, và khả dụng đối với cộng đồng mã nguồn mở. Trường này cuối cùng bị đóng cửa, vì vậy các nhà nghiên cứu, một trong số đó là David Cuarlielles, đã phổ biến ý tưởng này.

1.1.2. Các broad chính

+ Phần cứng Arduino gốc được sản xuất bởi công ty Italy tên là Smart Projects. Một vài board dẫn xuất từ Arduino cũng được thiết kế bởi công ty của Mỹ tên là SparkFun Electronics.

+ Sáu phiên bản phần cứng của Arduino cũng đã được sản xuất thương mại tính đến thời điểm hiện tại.

+ Các broad Arduino mẫu:

- Arduino Diecimila in Stoicheia

- Arduino Duemilanove (rev 2009b)

- Arduino UNO

- Arduino Leonardo

- Arduino Mega

- Arduino MEGA 2560 R3

- Arduino Nano

- Arduino Due (nền tảng ARM)

1.1.3. Arduino Uno R3

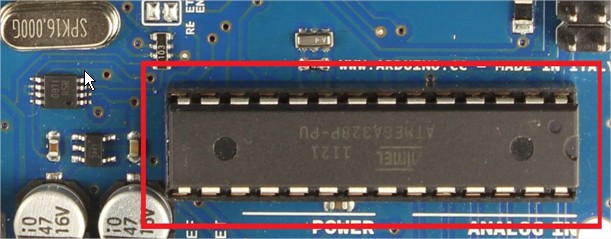


Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà người ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3)

1.1.3.1. Thông số cơ bản

|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 họ 8bit |
| Điện áp hoạt động | 5VDC (Chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16MHz |
| Dòng tiêu thụ | Khoảng 30mA |
| Điện áp vào khuyến dùng | 7 – 12VDC |
| Điện áp vào giới hạn | 6 – 20VDC |
| 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 | 30mA |
| Dòng ra tối đa (5V) | 500mA |
| Dòng ra tối đa (3.3V) | 50mA |
| Bộ nhớ flash | 32KB (ATmega328) với 0.5KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |

1.1.3.2. Vi điều khiển



Vi điều khiển được sử dụng ở đây là ATmega328. Đây là bộ não của Arduino UNO R3, nó có thể xử lý những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa …

1.1.3.3. Năng lượng

+ 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 năng lượng:

- GND: cực âm của nguồn điện cấp cho Arduino UNO.

- 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: nối cực dương của nguồn cấp với chân này để cấp nguồn cho Arduino UNO.

- IOREF: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này.

- RESET: dùng để reset vi điều khiển.

***\* Một vài lưu ý khi cấp nguồn cho Arduino UNO:***

- Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó phải hết sức cẩn thận kiểm tra cực âm – dương trước khi cấp nguồn cho Arduino UNO. Việc làm chập mạch nguồn vào của Arduino UNO sẽ khiến nó bị hỏng, không thể dùng được nữa.

- Các chân 3.3V và 5V dùng để cấp nguồn cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng mạch.

- Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng mạch.

- Cấp điện áp trên 13V vào chân RESET trên mạch có thể làm hỏng vi điều khiển ATmega328.

- Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.

- Cấp điện áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển.

- Cường độ dòng điện qua một chân Digital hoặc Analog bất kỳ của Arduino UNO nếu vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó, nếu không để truyền/nhận dữ liệu, ta phải mắc một điện trở hạn dòng.

1.1.3.4. Bộ nhớ

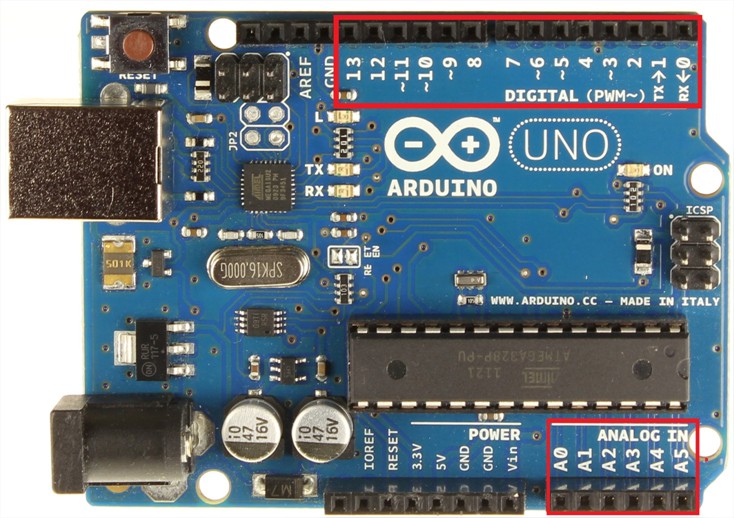
Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

+ 32KB bộ nhớ Flash: nơi lưu trữ những đoạn lệnh. Thường thì sẽ có khoảng vài KB trong số 32KB được dùng cho bootloader.

+ 2KB cho SRAM: giá trị các biến được khai báo khi lập trình sẽ lưu ở đây. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.

+ 1KB cho EEPROM: là nơi dùng để đọc/ghi dữ liệu mà không lo bị mất đi khi mất điện.

1.1.3.5. Các cổng vào ra



+ Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có hai 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. Một số chân digital có 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 các thiết bị khác thông qua hai chân này.

- Chân PWM (gồm các chân: 3, 5, 6, 9, 10, 11): cho phép 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, điện áp ở chân này có thể được điều chỉnh từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.

- Chân giao tiếp SPI (gồm các chân: 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 một đèn led màu cam (ký hiệu chữ L). Khi bấm nút Reset, đèn này sẽ nhấp nháy. Đèn led này được nối với chân 13. Khi chân này được sử dụng, đèn led sẽ sáng.

+ Arduino UNO có 6 chân analog cung cấp độ phân giải tín hiệu 10bit để đọc giá trị điện áp trong khoảng 0 – 5V. Đặc biệt, Arduino UNO có hai chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

1.1.4. Lập trình cho Arduino

+ Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++. Riêng mình thì gọi nó là “ngôn ngữ Arduino”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu.

+ Người dùng chỉ cần định nghĩa 2 hàm để tạo ra một chương trình vòng thực thi (cyclic executive) có thể chạy được:

- **setup():** hàm này chạy mỗi khi khởi động một chương trình, dùng để thiết lập các cài đặt

- **loop():** hàm này được gọi lặp lại cho đến khi tắt nguồn board mạch

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

#define LED\_PIN 13

void setup () {

pinMode (LED\_PIN, OUTPUT); *// Đặt chân 13 làm đầu ra digital*

}

void loop () {

digitalWrite (LED\_PIN, HIGH); *// Bật LED on*

delay (1000); *// chờ trong 1 giây (1000 mili giây)*

digitalWrite (LED\_PIN, LOW); *// Tắt LED off*

delay (1000); *// chờ trong 1s*

}

### 1.2. Module Sim808

1.2.1. Tổng quát

+ Module Sim 808 là Module GSM/GPS, được xây dựng dựa trên Sim808 của SIMCOM, hỗ trợ GSM/ GPRS với bốn băng tần và công nghệ định vị vệ tinh GPS. Ngoài hai chức năng chính GSM, GPS, Module Sim808 còn hỗ trợ thêm tính năng Bluetooth.

+ Module Sim808 có GPS với độ nhạy cao với 22 kênh theo dõi và 66 kênh tiếp nhận. Bên cạnh đó, nó cũng hỗ trợ công nghệ A-GPS, giúp cho việc định vị được chính xác hơn, ngay cả khi thiết bị ở trong nhà.

+ Đặc tính chung:

- Hoạt động ở 4 băng tần Quad-band 850/900/1800/1900MHz.

- Đã tích hợp module nguồn DC-DC, với nguồn đầu vào 9-24VDC cho đầu ra 4.3VDC cấp cho module sim.

- Hỗ trợ GPRS multi-slot class 12.

- Điều khiển bằng tập lệnh AT qua giao diện UART TTL.

- Tích hợp GPS/CNSS và hỗ trợ A-GPS.

- Dùng được cho cả MCU 3.3V và 5V.

1.2.2. Sơ đồ chân



+ Vin MCU: Là chân input, dùng để tương thích mức điện áp giao tiếp UART giữa vi điều khiển và module. Nguồn của vi điều khiển là 3.3V hay 5V sẽ cấp tới chân Vin MCU.

+ Vin Sim: Là chân input, chân nguồn cấp cho Module Sim808. Với Module Sim 808 đã tích hợp nguồn DC-DC, chân Vin Sim không cần sử dụng.

+ STA: Là chân output, có thể dùng để đọc trạng thái của Module Sim808, để xem Module đã được khởi động hay chưa? Nếu tín hiệu đọc về từ chân STA ở mức cao thì Module đã được khởi động, còn ở mức thấp thì Module đang ngừng hoạt động.

+ PWK: Là chân input, dùng để bật hoặc tắt Module Sim808. Chân PWK được điều khiển từ mức thấp lên mức cao, với thời gian ở mức cao tối thiểu 1 giây thì Module Sim808 sẽ được bật hoặc tắt.

RST: Là chân input, dùng để khởi động lại Module Sim808. Để reset Module Sim808 xuất một xung từ mức thấp lên mức cao tới chân RST, với thời gian ở mức cao tối thiểu 105ms.

+ RXD: Là chân output, được nối với chân RXD của MCU.

+ TXD: Là chân input, được nối với chân TXD của MCU.

+ GND: 0VDC, được nối chung với GND của MCU.

## 2. Các công nghệ liên quan

### 2.1. Internet of things

2.1.1. Internet of things là gì?

+ Internet of things (viết tắt là IoT) là một liên mạng của các thiết bị vật lý, các loại xe cộ, các tòa nhà, và các đồ vật khác mà ở đó, mỗi vật thể đều được gắn với các thiết bị điện tử, các phần mềm, các cảm biến và khả năng kết nối mạng nhằm mục đích giúp cho các vật thể này có thể thu thập và trao đổi dữ liệu với nhau. Nói đơn giản IoT là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.

+ Một vật thể trong IoT có thể là một con người với một trái tim cấy ghép; một con vật ở trang trại với bộ chip sinh học; một chiếc xe với bộ cảm ứng tích hợp cảnh báo tài xế khi bánh xe xẹp hoặc bất kỳ vật thể tự nhiên hay nhân tạo nào mà có thể gán được một địa chỉ IP và cung cấp khả năng truyền dữ liệu thông qua mạng lưới.

+ Điểm quan trọng của IoT đó là các đối tượng phải có thể được nhận biết và định dạng. Nếu mọi đối tượng, kể cả con người, được “đánh dấu” để phân biệt bản thân đối tượng đó với những thứ xung quanh thì chúng ta có thể hoàn toàn quản lý được nó thông qua máy tính. Việc kết nối thì có thể thực hiện qua Wifi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, hồng ngoại…

2.1.2. Ứng dụng của IoT

+ IoT có ứng dụng rộng vô cùng, có thể kể ra một số thứ sau:

- Quản lý chất thải.

- Quản lý và lập kế hoạch quản lý đô thị.

- Quản lý môi trường.

- Mua sắm thông minh.

- Nhà thông minh.

- Quản lý các thiết bị cá nhân.

+ Tác động của IOT rất đa dạng, trên các lĩnh vực: quản lý hạ tầng, y tế, xây dựng và tự động hóa, giao thông…. Cụ thể trong lĩnh vực y tế, Thiết bị IoT có thể được sử dụng để cho phép theo dõi sức khỏe từ xa và hệ thống thông báo khẩn cấp. Các thiết bị theo dõi sức khỏe có thể dao động từ huyết áp và nhịp tim màn với các thiết bị tiên tiến có khả năng giám sát cấy ghép đặc biệt, chẳng hạn như máy điều hòa nhịp hoặc trợ thính tiên tiến.

### 2.2. GPRS

2.2.1. GPRS là gì?

+ GPRS (General Packet Radio Service) là một dịch vụ dữ liệu di động dạng gói dành cho những người dùng Hệ thống thông tin di động toàn cầu (GSM) và điện thoại di động IS-136.

+ Nó cung cấp dữ liệu ở tốc độ từ 56 đến 114 kbps.

+ GPRS có thể được dùng cho những dịch vụ như truy cập Giao thức Ứng dụng Không dây (WAP), Dịch vụ tin nhắn ngắn (SMS), Dịch vụ nhắn tin đa phương tiện (MMS), và với các dịch vụ liên lạc Internet như email và truy cập World Wide Web.

+ Các hệ thống di động 2G kết hợp với GPRS thường được gọi là **"2.5G"**, có nghĩa là, một công nghệ trung gian giữa thế hệ điện thoại di động thứ hai (2G) và thứ ba (3G). Nó cung cấp tốc độ truyền tải dữ liệu vừa phải, bằng cách sử dụng các kênh đa truy cập theo phân chia thời gian (TDMA) đang còn trống, ví dụ hệ thống GSM.

2.2.2. Phân loại các thiết bị sử dụng GPRS

Các thiết bị được sử dụng GPRS được chia thành 3 loại:

+ **Loại A:** Có thể kết nối vào dịch vụ GPRS và dịch vụ GSM (thoại, SMS) cùng lúc.

**+ Loại B:** Có thể kết nối vào dịch vụ GPRS và dịch vụ GSM (thoại, SMS) nhưng chỉ dùng một trong hai dịch vụ vào một thời điểm.

**+ Loại C:** Được kết nối với hoặc dịch vụ GPRS hoặc dịch vụ GSM (thoại, SMS). Việc chuyển đổi giữa hai dịch vụ phải thực hiện thủ công.

### 2.3. GPS

2.3.1. GPS là gì?

+ **GPS**là viết tắt của "Global Positioning System" (hệ thống định vị toàn cầu), thực chất là một mạng lưới bao gồm 27 vệ tinh quay xung quanh trái đất. Trong số 27 vệ tinh này, 24 vệ tinh đang hoạt động, 3 vệ tinh còn lại đóng vai trò dự phòng trong trường hợp 1 trong số 24 vệ tinh chính bị hư hỏng.

+ Dựa vào cách sắp đặt của các vệ tinh này, khi đứng dưới mặt đất, bạn có thể nhìn được ít nhất là 4 vệ tinh trên bầu trời tại bất kì thời điểm nào.

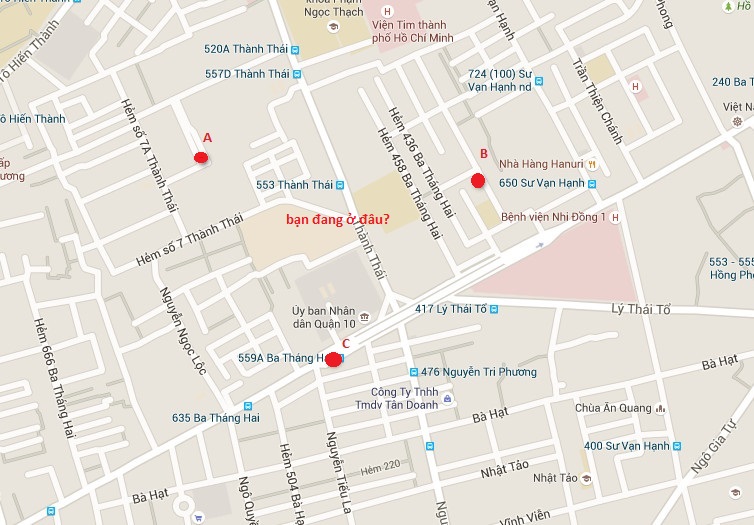
2.3.2. Cơ chế hoạt động của GPS

+ Các vệ tinh **GPS**bay hai vòng trong một ngày theo một quỹ đạo đã được tính toán chính xác và liên tục phát các tín hiệu có thông tin xuống Trái Đất. Các máy thu GPS nhận các tín hiệu này và giải mã bằng các phép tính lượng giác, qua đó sẽ tính toán và hiển thị được vị trí của người dùng.

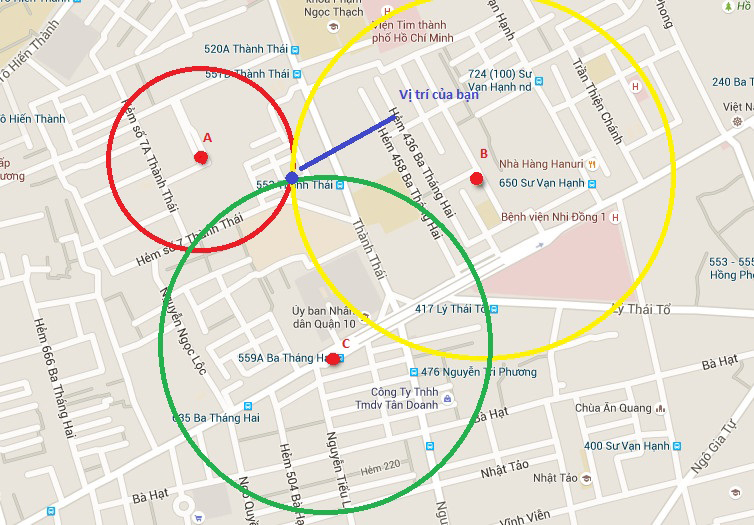
+ Các đầu thu GPS (các thiết bị smartphone…) thu dữ liệu từ các vệ tinh GPS ở trên bầu trời. Nói một cách đơn giản, mỗi vệ tinh cho bạn biết khoảng cách chính xác từ vị trí của bạn đến vệ tinh đó hoặc một điểm nào đó trên trái đất

+ Cơ chế hoạt động của GPS rất đơn giản, bạn có thể tưởng tượng như sau:

- Trên bản đồ có 3 điểm cố định A, B, C. Dữ liệu GPScho bạn biết khoảng cách lần lượt từ điểm A, B, C đến nơi bạn đứng là 1km, 3km, 2km



- Sau đó bạn vẽ 3 vòng tròn có tâm là A, B, C với bán kính lần lượt là 1km, 3km và 2km.



- Vị trí giao nhau của ba vòng tròn chính là vị trí của bạn

+ Thiết bị thu phải nhận được tín hiệu của ít nhất ba vệ tinh để cho ra vị trí hai chiều và để theo dõi được chuyển động của bạn. Khi nhận được tín hiệu của ít nhất bốn vệ tinh, máy sẽ cho ra được vị trí ba chiều. Một khi vị trí của bạn đã tính được thì thiết bị thu có thể tính các thông tin khác, như tốc độ di chuyển, hướng chuyển động, bám sát di chuyển, khoảng hành trình, khoảng cách đích đến và nhiều thứ khác nữa.

+ Để đưa ra vị trí chính xác, rất nhiều thiết bị GPS kết nối tới ít nhất là 4 vệ tinh. Đó là lý do vì sao đôi khi để tìm ra vị trí chính xác của bạn, hệ thống GPS lại mất nhiều thời gian tới vậy. Đó cũng là lý do vì sao đôi khi bạn bị mất sóng GPS: thiết bị di động của bạn có thể đã kết nối tới 1 hoặc 2 vệ tinh, song 2 vệ tinh vẫn là không đủ.

2.3.3. Độ chính xác của GPS

+ Các máy thu GPS ngày nay cực kì chính xác, nhờ vào thiết kế nhiều kênh hoạt động song song của chúng. Các máy thu 12 kênh song song (của Garmin) nhanh chóng khóa vào các quả vệ tinh khi mới bật lên và duy trì kết nối bền vững.

+ Trạng thái của khí quyển và các nguồn gây sai số khác có thể ảnh hưởng tới độ chính xác của máy thu GPS. Các máy thu GPS có độ chính xác trung bình trong vòng 15 mét.

+ Các máy thu mới hơn với khả năng WAAS (*Wide Area Augmentation System*) có thể tăng độ chính xác trung bình tới dưới 3 mét. Không cần thêm thiết bị hay mất phí để có được lợi điểm của WAAS bằng các máy phát hiệu. Để thu được tín hiệu đã sửa lỗi, người dùng phải có máy thu tín hiệu vi sai bao gồm cả ăn-ten để dùng với máy thu GPS của họ.

2.3.4. Ứng dụng của GPS

GPS đã được ứng dụng rất nhiều trong đời sống sản xuất. Có thể kể đến một số ứng dụng như:

+ Ứng dụng trong dân dụng: giám sát quản lý vận tải; theo dõi vị trí, tốc độ, hướng di chuyển…

+ Ứng dụng trong quân sự: bom thông minh JDAM, tên lửa không đối đất, tên lửa tấn công đất liền, tên lửa hành trình, tên lửa đất đối đất…

2.4. Amazon Webservice

+ Amazon web services là một trong số các dịch vụ của Amazon hoạt động dựa trên nền tảng cloud computing (điện toán đám mây).

+ Tổng quan kiến trúc:

**- Global Infrastructure (cơ sở hạ tầng nằm trên toàn cầu).**

**- Các dịch vụ cơ sở – Lưu trữ: S3 (Simple Storage Service).**

**- Các dịch vụ cơ sở – Database.**

**- Các dịch vụ cơ sở – Mạng: VPC.**

**- Application Services (Các dịch vụ ứng dụng).**

+ Amazon Elastic Compute Cloud (Amazon EC2) là một dịch vụ IaaS (Infrastructure as a Service) trên hạ tầng cơ sở của Amazon. Có nghĩa là bạn có thể tạo hoặc tạm dừng ra một máy chủ ảo (Virtual Machine) với các cấu hình thông số phần cứng có thể thay đổi rất nhanh với giao diện trực quan, đặc biệt là chi phí giá thành khá rẻ với dịch vụ on-demand tức là dùng bao nhiêu tính tiền bấy nhiêu. So với việc bạn tự xây dựng infrastructure, thì việc sử dụng EC2 sẽ rất tiện lợi và tiết kiệm chi phí và nhân sự rất nhiều.

# **CHƯƠNG II: MÔ TẢ HỆ THỐNG**

## 1. Nhiệm vụ cơ bản

Hệ thống được xây dựng nhằm mục đích giúp người dùng có thể giám sát các phương tiện của mình một cách dễ dàng, thuận lợi nhất. Thông qua hệ thống, người dùng có thể xem được vị trí hiện tại của phương tiện của mình, xem lịch sử di chuyển của phương tiện đó.

## 2. Mô tả hệ thống

Hệ thống được thiết kế giành cho việc giám sát phương tiện của người dùng. Với hệ thống này vai trò của mỗi người tham gia hệ thống như sau:

\* Mạch Arduino + Sim808: Chịu trách nhiệm gửi lên server tọa độ của phương tiện được giám sát.

\* Con người:

+ Quản trị viên:

- Thực hiện đăng ký thiết bị và tài khoản sử dụng cho người dùng. Với thiết bị thì tiến hành nhập biển số xe đăng ký cùng thiết bị, sau đó nạp chương trình cho thiết bị. Với mỗi thiết bị thì chỉ có thể đăng ký cho một xe.

- Hỗ trợ chuyển đổi thiết bị sang phương tiện khác cho người dùng.

- Giải đắp các thắc mắc của người dùng.

+ Kỹ thuật viên:

- Hỗ trợ lắp đặt thiết bị vào phương tiện cho người dùng khi có yêu cầu.

- Sữa chữa thiết bị.

- Bảo trì server thường xuyên để hệ thống có thể hoạt động trơn tru, không bị gián đoạn.

+ Người dùng:

- Mua thiết bị và đăng ký với nhà sản xuất để lấy tài khoản sử dụng trong phần mềm di động. Thông tin đăng ký bao gồm: biển số xe, loại xe, hãng xe, địa chỉ email, số điện thoại, địa chỉ nhà, tên đăng nhập. Sau khi mua thiết bị cũng như đăng ký phương tiện sử dụng thiết bị thì người dùng tiến hành lắp đặt thiết bị vào phương tiện.

- Với người dùng lần đầu đăng nhập vào ứng dụng, người dùng sẽ bắt buộc phải cập nhật đầy đủ thông tin yêu cầu bao gồm: tên, tuổi, số điện thoại, địa chỉ … Một tài khoản có thể theo dõi được nhiều thiết bị.

- Sau khi đăng nhập thành công, người dùng có thể chọn xe được giám sát và sau đó có thể giám sát vị trí hiện tại của xe.

- Người dùng có thể xem lại lịch sử di chuyển của từng xe theo ngày trên bản đồ, xem những đoạn đường mà xe đã đi qua.

- Người dùng có thể tìm đường đến vị trí hiện tại của xe. Thêm nữa, người dùng cũng có thể tìm đường đến một vị trí bất kỳ.

- Ngoài ra, người dùng có thể bật/tắt chức năng ghi lại lịch sử di chuyển ở trên thiết bị giám sát. Khi bắt đầu di chuyển thì người dùng bật chức năng ghi lại lịch sử di chuyển của phương tiện bằng cách ấn vào nút đen trên thiết bị. Lúc này, đèn led trên thiết bị sẽ sáng báo hiệu việc ghi lại lịch sử di chuyển đang diễn ra. Lúc không di chuyển nữa thì người dùng sẽ ấn vào nút đen trên thiết bị để tắt chức năng ghi lại lịch sử di chuyển. Lúc này đèn led trên thiết bị sẽ tối đi.

- Nếu muốn thay đổi xe được giám sát của một thiết bị đã đăng ký với xe khác trước đó, người dùng sẽ phải mang thiết bị đến cửa hàng nhờ quản trị viên thay đổi giúp. Lúc này quản trị viên sẽ tiến hành nhập biển số xe mới và nạp lại chương trình cho thiết bị.

# **CHƯƠNG III: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG**

## 1. Phân tích chức năng nghiệp vụ

### 1.1. Xác định các chức năng nghiệp vụ

Từ mô tả hệ thống, ta có thể xác định các chức năng nghiệp vụ của hệ thống như sau:

+ Đăng ký.

+ Đăng nhập.

+ Cập nhập thông tin cá nhân.

+ Thêm phương tiện được giám sát.

+ Chọn phương tiện để giám sát.

+ Xem vị trí hiện tại của phương tiện được chọn.

+ Xem lịch sử di chuyển của phương tiện được chọn theo ngày.

+ Bật tắt ghi lại lịch sử di chuyển.

### 1.2. Sơ đồ use case

## 2. Phân tích dữ liệu nghiệp vụ

### 2.1. Xác định kiểu dữ liệu ban đầu

Từ mô tả hệ thống, ta xác định được những thực thể sau:

NGƯỜI DÙNG (Email, Tên đăng nhập, Mật khẩu, Họ tên, Địa chỉ nhà, Số điện thoại, Vai trò).

PHƯƠNG TIỆN (Biển số xe, Tên người sở hữu, Hãng xe, Loại xe, Mô tả, Vĩ độ, Kinh độ, Ngày, Thời gian bắt đầu, Thời gian kết thúc).

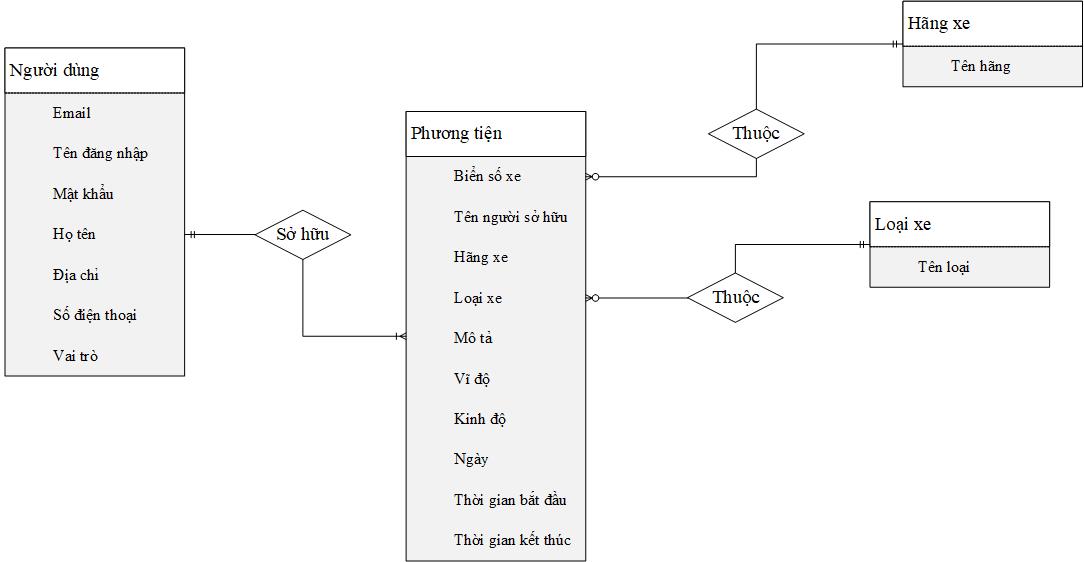
HÃNG XE (Tên hãng xe).

LOẠI XE (Tên loại xe).

VỊ TRÍ (Vĩ độ, Kinh độ, Biển số xe, Ngày, Thời gian bắt đầu, Thời gian kết thúc).

### 2.2. Mô hình liên kết thực thể

2.2.1. Mô hình liên kết thực thể mở rộng

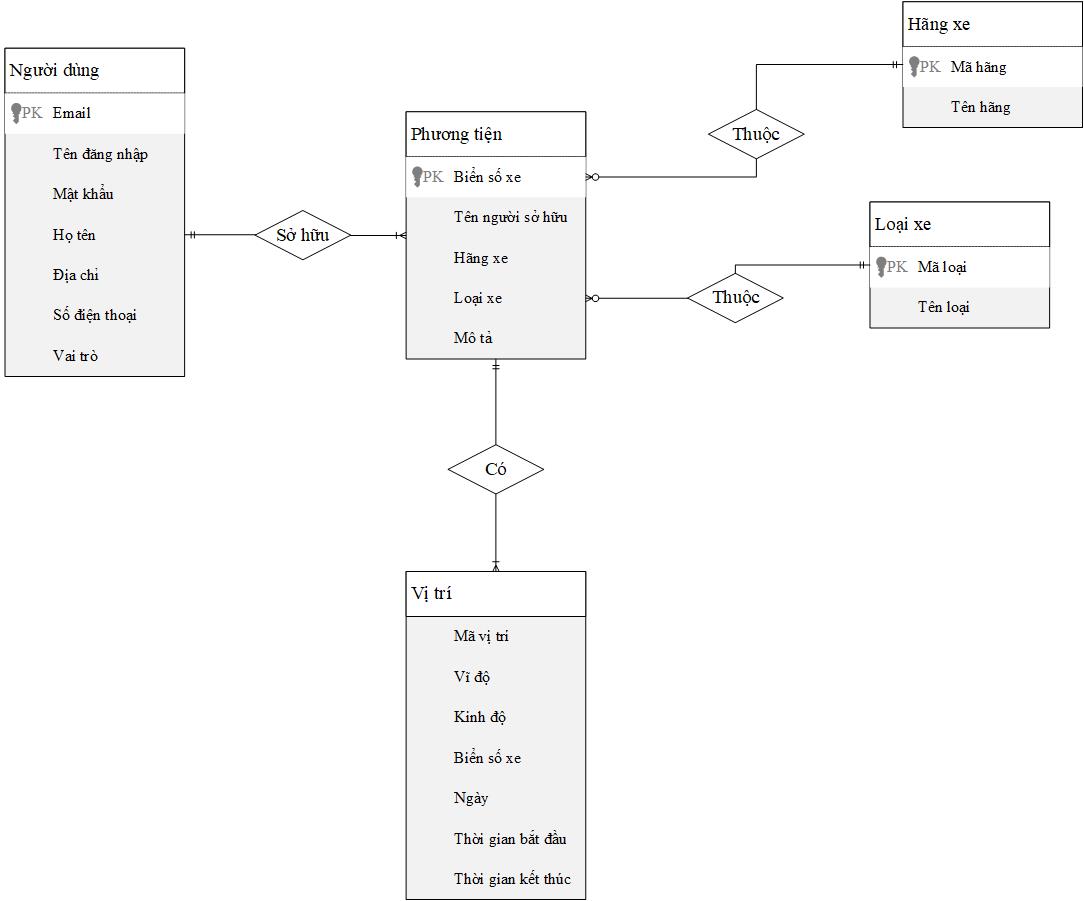


2.2.2. Mô hình liên kết thực thể kinh điển

Từ mô hình liên kết thực thể mở rộng, ta chuyển sang mô hình liên kết thực thể kinh điển như sau:

+ Các bảng có thuộc tính đa trị sẽ được tách thành các bảng riêng.

+ Thêm các thuộc tính cũng như xác định khóa chính cho các bảng.

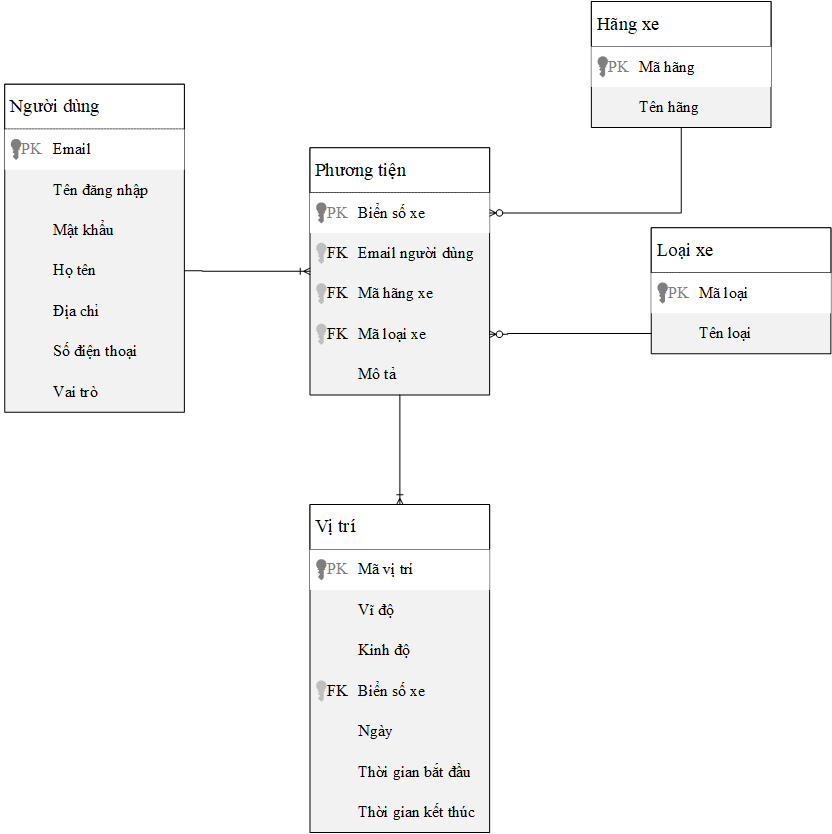


2.2.3. Mô hình liên kết thực thể hạn chế

Từ mô hình liên kết thực thể kinh điển, ta chuyển sang mô hình liên kết thực thể hạn chế như sau:

+ Thêm các thuộc tính khóa ngoại vào các bảng.

+ Bổ sung đầy đủ khóa chính cho các bảng.



### 2.3. Mô hình quan hệ

Từ mô hình liên kết thực thể hạn chế, ta chuyển sang mô hình quan hệ như sau:

+ Chuẩn hóa lại tên gọi:

- Thực thể “Người dùng”:

|  |  |  |
| --- | --- | --- |
| Người dùng 🠢 User | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Email | Email |
| 2 | Tên đăng nhập | Username |
| 3 | Password | Password |
| 4 | Họ tên | Name |
| 5 | Địa chỉ | Address |
| 6 | Số điện thoại | PhoneNumber |
| 7 | Vai trò | Role |

- Thực thể “Phương tiện”:

|  |  |  |
| --- | --- | --- |
| Phương tiện 🠢 Vehicle | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Biển số xe | NumberPlate |
| 2 | Email người dùng | UserEmail |
| 3 | Mã hãng xe | BranchId |
| 4 | Mã loại xe | TypeId |
| 5 | Mô tả | Description |

- Thực thể “Hãng xe”:

|  |  |  |
| --- | --- | --- |
| Hãng xe 🠢 Branch | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Mã hãng | Id |
| 2 | Tên hãng | Name |

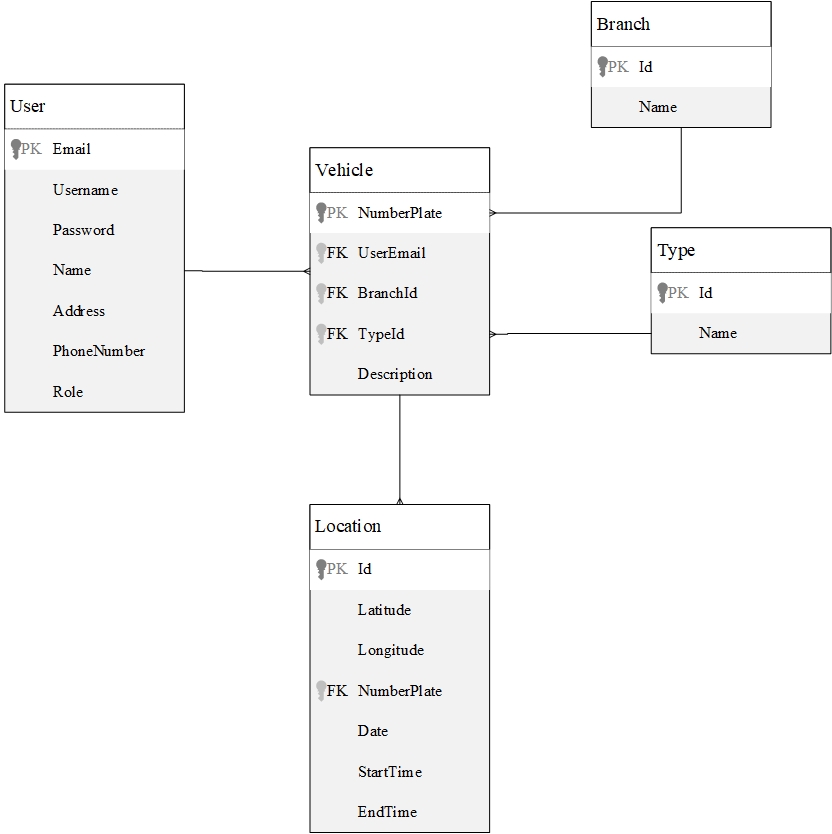
- Thực thể “Loại xe”:

|  |  |  |
| --- | --- | --- |
| Loại xe 🠢 Type | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Mã loại | Id |
| 2 | Tên loại | Name |

- Thực thể “Vị trí”:

|  |  |  |
| --- | --- | --- |
| Vị trí 🠢 Location | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Mã vị trí | Id |
| 2 | Vĩ độ | Latitude |
| 3 | Kinh độ | Longitude |
| 4 | Biển số xe | NumberPlate |
| 5 | Ngày | Date |
| 6 | Thời gian bắt đầu | StartTime |
| 7 | Thời gian kết thúc | EndTime |

*Mô hình quan hệ:*



### 2.4. Đặc tả dữ liệu

+ Bảng “User”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | Email | varchar(80) | x |  | Email đăng ký tài khoản |
| 2 | Username | varchar(20) |  |  | Tên đăng nhập hiển thị |
| 3 | Password | varchar(20) |  |  | Mật khẩu đăng nhập |
| 4 | Name | nvarchar(100) |  |  | Tên thật |
| 5 | Address | nvarchar(100) |  |  | Địa chỉ nhà |
| 6 | PhoneNumber | varchar(20) |  |  | Số điện thoại của người dùng |
| 7 | Role | int |  |  | Vai trò của người dùng:  1 – Quản trị viên  2 – Người dùng |

+ Bảng “Vehicle”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | NumberPlate | varchar(20) | x |  | Biển số xe đăng ký với thiết bị |
| 2 | UserEmail | varchar(80) |  | x | Email của người dùng |
| 3 | BranchId | int |  | x | Mã hãng xe |
| 4 | TypeId | int |  | x | Mã loại xe |
| 5 | Description | nvarchar(300) |  |  | Mô tả về thiết bị |

+ Bảng “Branch”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | Id | int | x |  | Id |
| 2 | Name | nvarchar(100) |  |  | Tên hãng xe |

+ Bảng “Type”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | Id | int | x |  | Id |
| 2 | Name | nvarchar(100) |  |  | Tên loại xe |

+ Bảng “Location”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | Id | int | x |  | Id |
| 2 | Latitude | varchar(20) |  |  | Vĩ độ |
| 3 | Longitude | varchar(20) |  |  | Kinh độ |
| 4 | NumberPlate | varchar(20) |  | x | Biển số xe |
| 5 | Date | varchar(20) |  |  | Ngày |
| 6 | StartTime | varchar(20) |  |  | Thời gian bắt đầu ghi lịch sử |
| 7 | EndTime | varchar(20) |  |  | Thời gian kết thúc ghi lịch sử |

## 3. Thiết kế giao diện

### 3.1. Giao diện quản trị

### 3.2. Giao diện người dùng

# **CHƯƠNG IV: TRIỂN KHAI PHẦN CỨNG**

## 1. Triển khai phần cứng

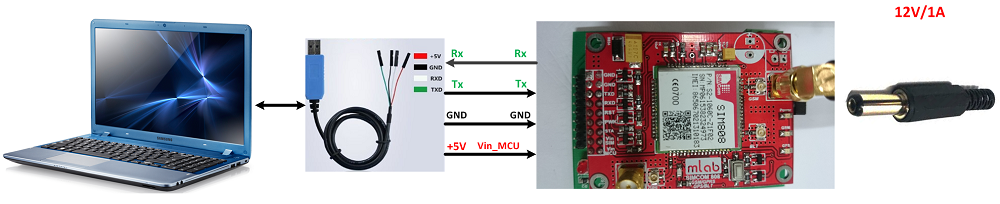
### 1.1. Lập trình với Module Sim808

\* Yêu cầu:

+ 1 nguồn 12V – 2A.

+ 1 dây USB to TTL.

\* Cài đặt như sau



\* Lập trình

+ Các lệnh chung

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Giá trị trả về** |
| AT | Kiểm tra phản hồi của Module Sim808 | OK => Module hoạt động bình thường |

+ Lệnh điều khiển GPS

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Giá trị trả về** |
| AT+CGNSPWR=[x] | Điều khiển bật tắt GPS  x = 1: bật  x = 0: tắt |  |
| AT+CGNSINF | Yêu cầu thông báo thông tin GPS |  |
| AT+CGNSURC=[x] | Cài đặt chu kỳ báo cáo thông tin GPS  x là thời gian theo mili giây. |  |

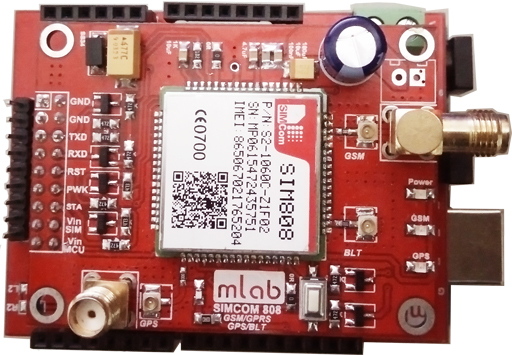
+ Lệnh điều khiển GPRS

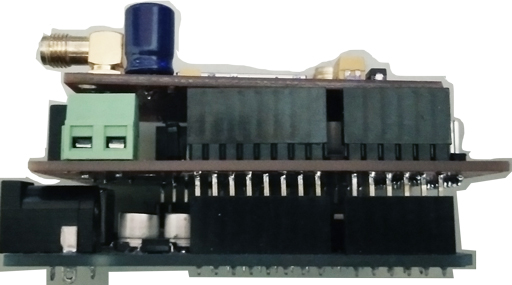
|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Giá trị trả về** |
| AT+CGATT? | Xem trạng thái đã được gắn GPRS hay chưa | + CGATT: <state>  OK  <state>: 0 – chưa gắn  1 – đã gắn |
| AT+CGACT=<state>,<cid> | Kích hoạt hoặc hủy bỏ PDP Context | OK  <state>: 0 – hủy bỏ  1 – kích hoạt  <cid>: PDP context id |

### 1.2. Kết hợp Module Sim808 với Arduino UNO

\* Yêu cầu: 1 nguồn 12V – 2A

\* Cài đặt:





\* Lập trình

+ Hàm lấy thông tin GPS

|  |
| --- |
| try\_get\_gps:  // Read gps if it's available  sendCommand("AT+CGNSINF\r");  readData(&response, 200);  if (response.length() > 0)  Serial.println(response);  // Split Gps Info  clearBuffer();  // Get number of words after splitting the response string  int numberOfWords = 1;  for (int i = 0; i < response.length(); i++) {  if (response[i] == ',')  numberOfWords++;  }  // If there are more than two words, then execute  if (numberOfWords > 1) {  String result[numberOfWords];  split(response, result, ',');  for (int i = 0; i < numberOfWords; i++) {  Serial.println(result[i]);  clearBuffer();  }  // Check if gps state is ok (1). If it's not ok, start over from try\_get\_gps  if (result[GPS\_STATE\_INDEX] == "0") {  delay(5000);  goto try\_get\_gps;  }  gpsInformation.state = (result[GPS\_STATE\_INDEX] == "1");  gpsInformation.latitude = result[GPS\_LAT\_INDEX];  gpsInformation.longitude = result[GPS\_LONG\_INDEX];  gpsInformation.gpsSpeed = result[GPS\_SPEED\_INDEX].toFloat();  gpsInformation.orientation = result[GPS\_ORIENTATION\_INDEX].toFloat();  Serial.println(gpsInformation.latitude);  clearBuffer();  Serial.println(gpsInformation.longitude);  clearBuffer();  Serial.println(gpsInformation.gpsSpeed);  clearBuffer();  Serial.println(gpsInformation.orientation);  } |

## 2. Triển khai server

# **CHƯƠNG IV: KẾT LUẬN**

**5. Tài liệu tham khảo**

[1]. <https://www.arduino.cc/en/Guide/HomePage>

[2]. <http://www.webopedia.com/TERM/I/internet_of_things.html>

[3]. <https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_l%C6%B0%E1%BB%9Bi_v%E1%BA%A1n_v%E1%BA%ADt_k%E1%BA%BFt_n%E1%BB%91i_Internet>