

Tp. HCM, ngày 16 tháng 7 năm 2018

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

| | | |
|-------------------|-------------------------------|----------------|
| Họ tên sinh viên: | Nguyễn Quang Huy | MSSV: 14141130 |
| | Dương Văn Huân | MSSV: 14141120 |
| Chuyên ngành: | Kỹ thuật Điện tử Truyền thông | Mã ngành: 41 |
| Hệ đào tạo: | Đại học chính quy | Mã hệ: 1 |
| Khóa: | 2014 | Lớp: 14141DT |

I. TÊN ĐỀ TÀI: THIẾT KẾ HỆ THỐNG THÔNG MINH CẢNH BÁO CHO XE MÁY

II. NHIỆM VỤ

1. Các số liệu ban đầu:

[1] Phạm Quang Huy, Lê Cảnh Trung, “Lập trình điều khiển với Arduino”, NXB Khoa học và kỹ thuật, 2014.

[2] Trần Thu Hà, “Điện tử cơ bản”, NXB ĐH Quốc Gia Tp.HCM, 2013.

2. Nội dung thực hiện:

- Tìm hiểu nhu cầu thực tiễn cũng như công nghệ của đề tài.
- Tìm hiểu lý thuyết liên quan, các linh kiện sử dụng trong đề tài.
- Tính toán và thiết kế hệ thống điều khiển Arduino.
- Thiết kế mạch, điều khiển thiết bị, lập trình cho hệ thống.
- Đánh giá kết quả đã thực hiện được và tiến tới hoàn thiện đề tài.
- Kết luận về đề tài, đưa ra hướng phát triển trong tương lai.

III. NGÀY GIAO NHIỆM VỤ: 20/3/2018

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 15/7/2018

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: PGS.TS. Nguyễn Thanh Hải

CÁN BỘ HƯỚNG DẪN

BM. ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên 1: Nguyễn Quang Huy

Lớp: 14141DT2B

MSSV:14141130

Họ tên sinh viên 2: Dương Văn Huân

Lớp: 14141DT1B

MSSV:14141120

Tên đề tài: THIẾT KẾ HỆ THỐNG THÔNG MINH CẢNH BÁO CHO XE MÁY

| <i>Tuần</i> | <i>Nội dung</i> | <i>Xác nhận GVHD</i> |
|------------------------|---|--------------------------|
| Tuần 1 | Tìm hiểu công nghệ của đề tài đang thực hiện cũng như các lý thuyết liên quan. | |
| Tuần 2 | Tính toán và thiết kế sơ đồ khối của hệ thống. | |
| Tuần 3 | Tính toán và thiết kế sơ đồ nguyên lý của các mạch cần trong hệ thống, lựa chọn linh kiện cần thiết cho hệ thống. | |
| Tuần 4 | Nghiên cứu về Arduino UNO R3 | |
| Tuần 5 | Nghiên cứu module sim 808A, các cảm biến rung và module gia tốc. | |
| Tuần 6-7 | Lập trình giao tiếp giữa Arduino với module sim 808A, cảm biến rung và module gia tốc | |
| Tuần 8 | Chạy thử nghiệm hệ thống trên testboard, kit. | |
| Tuần 9-10 | Vẽ mạch in, thi công mạch in hệ thống. Lắp ráp và kiểm tra | |
| Tuần 11 | Thiết kế mô hình. | |
| Tuần 12 | Đóng gói hệ thống và chạy thử nghiệm. | |
| Tuần 13-Tuần 16 | Chạy thử nghiệm và cân chỉnh toàn hệ thống. Đánh giá kết quả đạt được, viết báo cáo. | |

GV HƯỚNG DẪN

LỜI CAM ĐOAN

Đề tài này là do chúng tôi tự thực hiện dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Người thực hiện đề tài

Nguyễn Quang Huy

Dương Văn Huân

LỜI CẢM ƠN

Trong thời gian thực hiện đề tài, những người thực hiện được sự giúp đỡ của gia đình, quý thầy cô và bạn bè nên đề tài đã được hoàn thành. Những người thực hiện xin chân thành gửi lời cảm ơn đến:

Thầy Nguyễn Thanh Hải, giảng viên trường Đại Học Sư Phạm Kỹ Thuật Tp.HCM đã trực tiếp hướng dẫn và tận tình giúp đỡ tạo điều kiện để nhóm có thể hoàn thành tốt đề tài.

Những người thực hiện cũng xin chân thành cảm ơn đến các thầy cô trong khoa Điện - Điện tử của trường Đại Học Sư Phạm Kỹ Thuật Tp.HCM đã tận tình dạy dỗ, chỉ bảo, cung cấp cho những người thực hiện những kiến thức nền, chuyên môn làm cơ sở để hoàn thành đề tài này.

Cảm ơn gia đình đã động viên và luôn luôn bên cạnh trong những lúc khó khăn nhất.

Xin gửi lời cảm ơn đến những người bạn sinh viên khoa Điện-Điện tử đã giúp đỡ những người thực hiện đề tài để có thể hoàn thành tốt đề tài này.

Xin chân thành cảm ơn!

Người thực hiện đề tài

Nguyễn Quang Huy

Dương Văn Huân

MỤC LỤC

| | |
|--|-----|
| NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP | i |
| LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP..... | ii |
| LỜI CAM ĐOAN | iii |
| LỜI CẢM ƠN | iv |
| MỤC LỤC | v |
| DANH MỤC HÌNH | vii |
| DANH MỤC BẢNG | ix |
| TÓM TẮT | x |
| CHƯƠNG 1. TỔNG QUAN | 1 |
| 1.1 Đặt vấn đề..... | 1 |
| 1.2 Mục đích nghiên cứu | 2 |
| 1.3 Nội dung nghiên cứu | 2 |
| 1.4 Giới hạn đề tài | 2 |
| 1.5 Bố cục | 2 |
| CHƯƠNG 2. CƠ SỞ LÝ THUYẾT | 4 |
| 2.1 Tổng quan về gps..... | 4 |
| 2.2 Giới thiệu phần cứng | 4 |
| 2.2.1 Board Arduino | 5 |
| 2.2.2 Module sim 808a | 6 |
| 2.2.3 Cảm biến rung HDX-01 | 8 |
| 2.2.4 Cảm biến gia tốc MPU-6050..... | 8 |
| 2.3 Các chuẩn giao tiếp | 9 |
| 2.3.1 Chuẩn giao tiếp uart..... | 9 |

| | |
|--|-----------|
| 2.3.2 Chuẩn giao tiếp I2C..... | 10 |
| CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ | 11 |
| 3.1 Giới thiệu | 11 |
| 3.2 Tính toán và thiết kế hệ thống | 11 |
| 3.2.1 Thiết kế sơ đồ khối hệ thống | 11 |
| 3.2.2 Tính toán và thiết kế mạch | 12 |
| 3.2.3 Sơ đồ nguyên lý của toàn mạch..... | 18 |
| CHƯƠNG 4. THI CÔNG HỆ THỐNG | 20 |
| 4.1 Giới thiệu | 20 |
| 4.2 Thi công hệ thống | 20 |
| 4.2.1 Thi công bo mạch hệ thống | 20 |
| 4.2.2 Lắp ráp và kiểm tra bo mạch hệ thống | 21 |
| 4.3 Lập trình hệ thống | 22 |
| 4.3.1 Lưu đồ..... | 22 |
| 4.3.2 Phần mềm lập trình cho vi điều khiển..... | 25 |
| 4.4 Viết tài liệu hướng dẫn sử dụng, thao tác..... | 29 |
| CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT - ĐÁNH GIÁ | 30 |
| 5.1 Kết quả..... | 30 |
| 5.2 Nhận xét – đánh giá..... | 36 |
| CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 37 |
| 6.1 Kết luận..... | 37 |
| 6.2 Hướng phát triển..... | 37 |
| TÀI LIỆU THAM KHẢO..... | 38 |
| DANH SÁCH TỪ VIẾT TẮT | 39 |
| PHỤ LỤC | 40 |

DANH MỤC HÌNH

| | |
|---|----|
| Hình 2.1. Board Arduino UNO R3 | 5 |
| Hình 2.2. Module sim 808A thực tế | 7 |
| Hình 2.3. Cảm biến rung HDX-01 | 8 |
| Hình 2.4. Cảm biến gia tốc MPU - 6050 | 8 |
| Hình 2.5. Giá đỡ truyền dữ liệu UART | 10 |
| Hình 3.1. Sơ đồ khối hệ thống | 12 |
| Hình 3.2. Khối xử lý trung tâm sử dụng board Arduino UNO R3 | 13 |
| Hình 3.3. Module Sim 808A | 13 |
| Hình 3.4. Sơ đồ nguyên lý kết nối Module Sim 808A vào Arduino UNO R3 | 14 |
| Hình 3.5. Sơ đồ nguyên lý kết nối module MPU-6050 vào Arduino | 15 |
| Hình 3.6. Sơ đồ nguyên lý kết nối cảm biến rung HDX-01 vào Arduino | 16 |
| Hình 3.7. Sơ đồ nguyên lý kết nối của khối nút nhấn | 17 |
| Hình 3.8. Sơ đồ nguyên lý kết nối của khối cảnh báo | 18 |
| Hình 3.9. Sơ đồ nguyên lý toàn mạch | 19 |
| Hình 4.1. Sơ đồ đi dây đã phủ đồng bo mạch hệ thống | 20 |
| Hình 4.2. Hình dạng 3D lớp bo mạch hệ thống | 21 |
| Hình 4.3. Lưu đồ khi cảm biến rung tác động | 22 |
| Hình 4.4. Lưu đồ khi module gia tốc tác động | 23 |
| Hình 4.5. Lưu đồ chính cho chương trình | 24 |
| Hình 4.6. Giao diện file menu arduino IDE | 25 |
| Hình 4.7. Giao diện Examples menu arduino IDE | 26 |
| Hình 4.8. Giao diện Sketch Menu Arduino IDE | 26 |
| Hình 4.9. Giao diện edit menu arduino IDE | 27 |
| Hình 4.10. Giao diện Tool Menu Arduino IDE | 27 |
| Hình 4.11. Board Arduino sử dụng | 28 |
| Hình 4.12. Arduino Toolbar | 28 |
| Hình 5.1. Hình thực tế khối nút nhấn và cảnh báo | 31 |
| Hình 5.2. Hình thực tế khối cảm biến | 31 |
| Hình 5.3. Hình thực tế gắn linh kiện cho mặt trên | 32 |
| Hình 5.4. Trạng thái led cho biết trạng thái của mạch | 33 |

| | |
|--|----|
| Hình 5.6. Thao tác với module gia tốc | 34 |
| Hình 5.7. Tọa độ GPS mà module sim gửi về điện thoại..... | 34 |
| Hình 5.8. Tra cứu địa chỉ với tọa độ nhận được | 35 |
| Hình 5.9. Lỗi thường gặp khi anten không bắt được tín hiệu..... | 35 |
| Hình 5.10. Tín nhắn cảnh báo chống trộm | 35 |

DANH MỤC BẢNG

| | |
|---|-----------|
| <i>Bảng 2.1. Cấu trúc tổng quát Arduino UNO R3.....</i> | <i>5</i> |
| <i>Bảng 5.1. Số liệu thực nghiệm</i> | <i>36</i> |

TÓM TẮT

Cuộc “Cách mạng Công nghiệp 4.0” đang dần diễn ra tại nhiều nước phát triển và đang phát triển trên toàn thế giới. Công nghệ về điều khiển thông minh cũng phát triển theo, chúng được ứng dụng trong nhiều lĩnh vực đời sống. Đặc biệt, các hệ thống chống trộm, cảnh báo đang được phát triển mạnh mẽ và ngày càng trở nên phổ biến.

DATN này thiết kế, thi công một mô hình hệ thống cảnh báo chống trộm và tai nạn cho người dùng được gắn trên xe máy. Các cảnh báo và giá trị vị trí GPS cũng được cập nhật và gửi tới điện thoại của người dùng. Người dùng có thể điều khiển mạch bằng chính điện thoại của mình.

CHƯƠNG 1. TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Với sự phát triển mạnh mẽ của khoa học và công nghệ ngày nay đi cùng với công cuộc công nghiệp hóa - hiện đại hóa đất nước đã thúc đẩy nền công nghiệp sản xuất nước nhà phát triển. Các phương tiện giao thông như xe máy, ô tô cũng được chế tạo ra không những ngày càng tinh tế mang tính hiện đại mà còn thông minh tăng tính tự động hóa nhằm nâng cao chất lượng sống của chúng ta. Theo quy hoạch phát triển ngành công nghiệp xe máy Việt Nam giai đoạn 2006 - 2015 có xét đến 2020, Bộ Công Thương ước tính lượng xe máy lưu thông toàn thị trường đến năm 2015 sẽ đạt 31 triệu xe và tăng lên 33 triệu xe vào 2020. Trong đó tăng bình quân hàng năm từ 2010-2020 sẽ đạt từ 1,8 đến 2,2 triệu xe mỗi năm [1]. Số lượng xe máy lớn như vậy đi kèm với một bộ phận ý thức người dân còn thấp sẽ dẫn đến mỗi nguy tai nạn giao thông ngày càng tăng cao. Với một đất nước mà xe máy là phương tiện di chuyển chính và còn là tài sản quý giá của mỗi con người, mỗi gia đình thì ngoài việc đảm bảo cho chiếc xe khỏi sự dòm ngó của những tên trộm như hệ thống chống trộm, cũng cần đòi hỏi một chiếc xe có tính thông minh, hàm lượng tự động hóa cao, đảm bảo an toàn cho người điều khiển phương tiện tránh khỏi những rủi ro đáng tiếc như va quệt hay tai nạn và kịp thời cứu chữa khi xảy ra tai nạn bằng hệ thống thông báo tới người thân, bệnh viện nhằm đưa số lượng các vụ tai nạn, số lượng người chết giảm xuống mức tối đa có thể.

Ngày nay, công nghệ kỹ thuật đang phát triển mạnh mẽ, các phương pháp lập trình rất đa dạng và lập trình nhúng là một hướng đi của xu thế hiện nay. Trong lập trình nhúng thì ngôn ngữ lập trình với Arduino là một trong những phương pháp sử dụng nhiều nhất. Arduino là một bo mạch xử lý hay còn gọi là vi điều khiển được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, màn hình, bluetooth, GPS. Tuy chỉ là một bo mạch nhỏ gọn, chi phí thấp vì bo đã được thi công sẵn, tuy nhiên Arduino có thể được ứng dụng vào rất nhiều lĩnh vực khác nhau, ta có thể đo đặc nhiệt độ để điều khiển tưới tiêu trong các nhà vườn, hoặc có thể dùng để điều khiển xe cân bằng rất phổ biến hiện nay, hay có thể ứng dụng nó vào việc điều khiển cánh tay Robot. Ngoài những lợi ích to lớn trên, việc sử dụng và lập trình trên

CHƯƠNG 1. TỔNG QUAN

Arduino cũng rất đơn giản, môi trường phát triển đa dạng, hỗ trợ trên nhiều hệ điều hành khác nhau, có thể học 1 cách nhanh chóng. Vì những điểm mạnh ưu việt như vậy nên nhóm đã quyết định chọn Arduino làm bộ xử lý trung tâm cho việc “THIẾT KẾ HỆ THỐNG THÔNG MINH CẢNH BÁO CHO XE MÁY”.

1.2 MỤC ĐÍCH NGHIÊN CỨU

Thiết kế hệ thống thông minh cảnh báo cho xe máy để hệ thống cảnh báo chạy ổn định, báo vị trí người dùng chính xác, gửi tin nhắn về số điện thoại nhanh.

1.3 NỘI DUNG NGHIÊN CỨU

- Đọc và tìm hiểu các nguồn tài liệu.
- Tìm hiểu cách kết nối giữa Arduino với module sim 808A, cảm biến rung và module gia tốc.
- Viết chương trình cho khối điều khiển Arduino, khối module sim 808A, khối cảm biến rung và nút nhấn.
- Vẽ mạch và thi công mạch
- Kết nối các khối lại với nhau.
- Chạy thử nghiệm sản phẩm .
- Viết luận văn.
- Báo cáo đề tài tốt nghiệp .

1.4 GIỚI HẠN ĐỀ TÀI

- Chỉ thiết kế mô hình nhỏ, gọn (dài 20cm, rộng 7cm) nên không thể áp dụng lên xe máy thực tế được.
- Định vị vị trí thông qua App Google Map nên điện thoại cần phải có mạng mới có thể tra cứu vị trí.
- Anten phải để ngoài trời mới có thể nhận được tín hiệu tốt.

1.5 BỐ CỤC

Với đề tài thiết kế hệ thống xe máy thông minh thì bố cục đồ án như sau:

- **Chương 1:** Tổng Quan

Trình bày đặt vấn đề dẫn nhập lý do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bố cục đồ án.

- **Chương 2: Cơ Sở Lý Thuyết**

Trình bày các lý thuyết liên quan đến các vấn đề mà đề tài sẽ dùng để thực hiện thiết kế, thi công cho đề tài, bao gồm các thông số, hình ảnh các linh kiện sử dụng trong đồ án.

- **Chương 3: Thiết Kế và Tính Toán**

Trình bày sơ đồ khối, chức năng từng khối. Tính toán và thiết kế từng khối để có thể vẽ sơ đồ nguyên lý toàn mạch.

- **Chương 4: Thi công hệ thống**

Trình bày kết quả thi công phần cứng và hình ảnh mô phỏng của mạch, các bước tiến hành thi công mạch. Ngoài ra, trong chương này còn đưa vào lưu đồ và các chương trình lập trình chính.

- **Chương 5: Kết quả - Nhận xét – đánh giá**

Trình bày kết quả của cả quá trình nghiên cứu, thời gian thực hiện và kiến thức trong suốt thời gian nhận được trong quá trình làm đồ án tốt nghiệp và cuối cùng là đưa ra nhận xét, đánh giá khách quan của bản thân về sản phẩm tự tay mình làm.

- **Chương 6: Kết luận và hướng phát triển**

Trình bày kết quả thu được dựa vào phương pháp và mục đích ban đầu, các việc đã và chưa làm được. Ngoài ra, trong chương này, trình bày hướng phát triển và mở rộng quy mô của đồ án tốt nghiệp này vào thực tế.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 TỔNG QUAN VỀ GPS

Hệ thống Định vị Toàn cầu (tiếng Anh: Global Positioning System - GPS) là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh. GPS là một hệ thống gồm 27 vệ tinh (kể cả 3 cái sơ cua) chuyển động trên các quỹ đạo chung quanh trái đất. Mỗi vệ tinh nặng khoảng 2 tấn, sử dụng năng lượng mặt trời, chuyển động cách mặt đất khoảng 19300km [2]. Mỗi vệ tinh quay quanh trái đất 2 vòng một ngày đêm. Quỹ đạo của các vệ tinh được tính toán sao cho ở bất kỳ nơi nào trên trái đất, vào bất kỳ thời điểm nào, cũng có thể “nhìn thấy” tối thiểu 4 vệ tinh. Công việc của một máy thu GPS là xác định vị trí của 4 vệ tinh hay hơn nữa, tính toán khoảng cách từ các vệ tinh và sử dụng các thông tin đó để xác định vị trí của chính nó. Quá trình này dựa trên một nguyên lý toán học đơn giản.

- Vĩ độ (ký hiệu: φ) của một điểm bất kỳ trên mặt Trái Đất là góc tạo thành giữa đường thẳng đứng (phương của dây dọi, có đỉnh nằm ở tâm hệ tọa độ-chính là trọng tâm của địa cầu) tại điểm đó và mặt phẳng tạo bởi xích đạo. Đường tạo bởi các điểm có cùng vĩ độ gọi là vĩ tuyến, và chúng là những đường tròn đồng tâm trên bề mặt Trái Đất.

- Kinh độ (ký hiệu: λ) của một điểm trên bề mặt Trái Đất là góc tạo ra giữa mặt phẳng kinh tuyến đi qua điểm đó và mặt phẳng kinh tuyến gốc. Những đường thẳng tạo bởi các điểm có cùng kinh độ gọi là kinh tuyến.

Bằng cách phối hợp hai góc này, ta có thể xác định được vị trí nằm ngang của bất kỳ điểm nào trên Trái Đất.

2.2 GIỚI THIỆU PHẦN CỨNG

- **Thiết bị đầu vào:** Nút nhấn đơn không chống dội, module cảm biến rung HDX-01, module Sim 808A, Cảm biến gia tốc MPU-6050.
- **Thiết bị đầu ra:** Buzzer, Led.

- **Thiết bị điều khiển trung tâm:** Board Arduino UNO R3.
- **Các chuẩn truyền dữ liệu:** UART, I2C.

2.2.1 Board Arduino

Arduino thật ra là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm. Các ứng dụng nổi bật của board mạch Arduino như robot đơn giản, điều khiển nhiệt độ, phát hiện chuyển động, game tương tác.

Board Arduino có rất nhiều phiên bản với hiệu năng và mục đích sử dụng khác nhau như: Arduino Mega, Arduino LilyPad. Trong số đó, Arduino Uno R3 là một trong những phiên bản được sử dụng rộng rãi nhất. Arduino UNO R3 là một dòng Arduino đủ mạnh về tính năng, 20 chân, bộ nhớ đủ cho hầu hết các ứng dụng thông thường.



Hình 2.1. Board Arduino UNO R3

Cấu trúc tổng quát của Arduino UNO R3 như sau:

Bảng 2.1. Cấu trúc tổng quát Arduino UNO R3

| | |
|-------------------------------|------------|
| Vi điều khiển | Atmega328P |
| Điện áp hoạt động | 5V |
| Điện áp đầu vào (khuyến dùng) | 7-12V |
| Điện áp đầu vào (giới hạn) | 6-20V |

| | |
|-----------------------|--|
| Chân Digital I/O | 14 (Với 6 chân PWM output) |
| Chân PWM Digital I/O | 6 |
| Chân đầu vào Analog | 6 |
| Dòng sử dụng I/O Pin | 20 mA |
| Dòng sử dụng 3.3V Pin | 50 mA |
| Bộ nhớ Flash | 32 KB (ATmega328P) với 0.5KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| LED_BUILTIN | 13 |
| Chiều dài | 68.6 mm |
| Chiều rộng | 53.4 mm |
| Trọng lượng | 25 |

2.2.2 Module sim 808a

Module Sim808A là Module GSM/GPS, xây dựng dựa trên Sim808 của SIMCOM, hỗ trợ GSM/GPRS với 4 băng tần và công nghệ định vị vệ tinh GPS. Ngoài 2 chức năng chính GSM/GPS, Module Sim808A còn hỗ trợ thêm tính năng Bluetooth.

Module Sim808A 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.

Module được thiết kế tối ưu, loại bỏ đi những tính năng không cần thiết để giảm giá thành và phục vụ chủ yếu cho việc giám sát, điều khiển các thiết bị từ xa thông qua GSM/GPRS/GPS.



Hình 2.2. Module sim 808A thực tế

- **Vin MCU:** Chân input, dùng để tương thích với mức điện áp giao tiếp UART giữa vi điều khiển với Module Sim808. Nguồn của vi điều khiển là 3.3V hay 5V sẽ cấp đến chân Vin Arduino.
- **Vin Sim:** Chân input, là chân nguồn cấp cho Module Sim808, nguồn cấp trong dải từ 3.7 đến 4.5V. Cấp nguồn áp quá mức này module sẽ bị hỏng. Với Module Sim808 đã tích hợp nguồn DC-DC, chân Vin Sim không cần sử dụng, do nguồn đã được cấp tới chân này từ module nguồn DC-DC.
- **ST:** Chân output, được dùng để đọc trạng thái của Module Sim808, để xem module đã khởi động được hay chưa? Nếu tín hiệu đọc về từ chân STA ở mức cao thì module khởi động, ở mức thấp thì module đang ngừng hoạt động.
- **PWK:** Chân Input, dùng để bật/tắt Module Sim808. Chân PWK dùng để điều khiển từ mức thấp lên mức cao, với thời gian ở mức cao là ít nhất 1 giây thì Module Sim808 sẽ được bật/tắt.
- **RST:** 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:** Chân output, nối với chân RXD của Arduino
- **TXD:** Chân input, nối với chân TXD của Arduino
- **GND:** 0 VDC, nối chung GND của Arduino

2.2.3 Cảm biến rung HDX-01

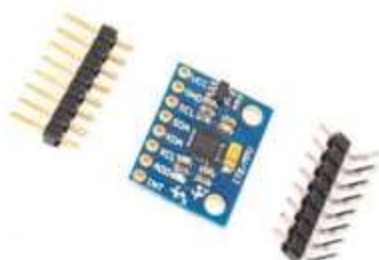


Hình 2.3. Cảm biến rung HDX-01

Chức năng : Cảm biến rung này sử dụng để đo độ rung. Nó có thể kích hoạt từ mọi góc độ và thường được sử dụng cho việc đo cảm ứng chạm, sốc. Có thể được sử dụng trong các thiết bị chống trộm, khóa điện tử, cơ khí phát hiện thiết bị rung.

- Điện áp hoạt động: < 24VDC
- Dòng điện: < 1mA
- Nhiệt độ: < 80°C

2.2.4 Cảm biến gia tốc MPU-6050



Hình 2.4. Cảm biến gia tốc MPU - 6050

Module cảm biến gia tốc MPU-6050 GY-521 tích hợp gia tốc 3 trục + con quay hồi chuyển 3 trục giúp kiểm soát cân bằng hoặc định hướng chuyển động cho robot, máy bay, drone, tay cầm chơi game, hệ thống giữ thăng bằng cho camera/máy ảnh, nhận biết sự rơi, rung, lắc.

Chức năng: Cảm biến gia tốc được ứng dụng nhiều trong nhiều lĩnh vực:

- Kỹ thuật: Cảm biến gia tốc có thể đo gia tốc của các phương tiện, đo gia tốc ô tô, máy móc, nhà cửa, các hệ thống tự động hóa và lắp đặt an toàn.

- Công nghiệp: Cảm biến gia tốc được sử dụng trong việc bảo quản máy móc để báo cáo sự dao động và biến đổi của các trục quay ở bi... kịp thời.
- Xây dựng: Cảm biến gia tốc được sử dụng để đo các chuyển động và dao động của một cấu trúc chịu ảnh hưởng của các hệ thống chuyển động.
- Các thiết bị cá nhân: Được tích hợp trong các smartphone, tablet... để điều khiển giao diện người dùng (xoay màn hình ngang/dọc).
- Y học: Vài năm gần đây, có nhiều công ty đã sản xuất các đồng hồ đeo tay thể thao có chân đế gắn các cảm biến gia tốc giúp cho việc theo dõi tốc độ và vị trí chạy của người đeo nó.

Thông số kỹ thuật

- Chip: MPU-6050 (16bit ADC, 16bit data out)
- Giá trị Gyroscopes trong khoảng: ± 250 500 1000 2000 degree/sec
- Giá trị Acceleration trong khoảng: $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$
- Giao tiếp: I2C
- Nguồn sử dụng: 3V - 5V (DC)

2.3 CÁC CHUẨN GIAO TIẾP

2.3.1 Chuẩn giao tiếp uart

UART : Universal Asynchronous Receiver/Transmitter, là kiểu truyền thông tin nối tiếp không đồng bộ, UART thường được dùng trong máy tính công nghiệp, truyền thông, vi điều khiển, hay một số các thiết bị truyền tin khác.

Một số thông số:

Baud rate (tốc độ Baud) : Khi truyền nhận không đồng bộ để hai mô đun hiểu được nhau thì cần quy định một khoảng thời gian cho 1 bit truyền nhận , nghĩa là trước khi truyền thì tốc độ phải được cài đặt đầu tiên . Theo định nghĩa thì tốc độ baud là số bit truyền trong một giây.

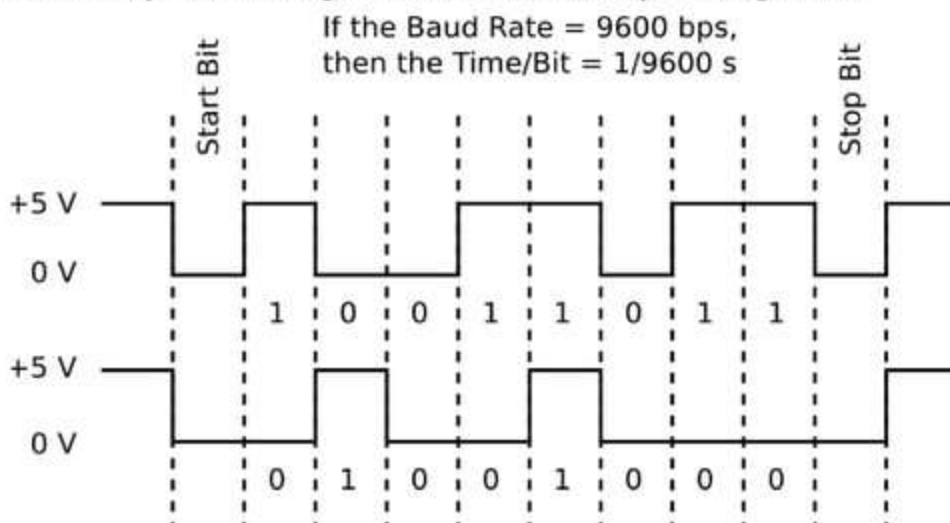
Frame (khung truyền) : Do kiểu truyền thông nối tiếp này rất dễ mất dữ liệu nên ngoài tốc độ , khung truyền cũng được cài đặt từ ban đầu để tránh bớt sự mất mát dữ liệu này . Khung truyền quy định số bit trong mỗi lần truyền , các bit báo như start , stop , các bit kiểm tra như parity, và số bit trong một data .

Bit Start : Là bit bắt đầu trong khung truyền Bit này nhằm mục đích báo cho thiết bị nhận biết quá trình truyền bắt đầu . trên AVR bit Start có trạng thái là 0 .

Data : Dữ liệu cần truyền Data không nhất thiết phải 8 bit, có thể là 5, 6, 7, 8, 9 . Trong UART bit LSB được truyền đi trước, Bit MSB được truyền đi sau .

Parity bit : Là bit kiểm tra dữ liệu đúng không . có 2 loại parity : chẵn (even parity) , lẻ (old parity) . Parity chẵn là bit parity thêm vào để số bit 1 trong data + parity = chẵn, parity lẻ là bit parity thêm vào để số bit 1 trong data + parity = lẻ. Bit Parity là không bắt buộc nên có thể dùng hoặc không.

Stop : là bit báo cáo kết thúc khung truyền, thường là mức 5V và có thể có 1 hoặc 2 bit stop . Giản đồ trong hình 2.10 mô tả dữ liệu truyền đi bằng UART.



Hình 2.5. Giản đồ truyền dữ liệu UART

2.3.2 Chuẩn giao tiếp I2C

I2C là 1 chuẩn truyền nối tiếp theo mô hình chủ – tớ. Một thiết bị chủ có thể giao tiếp với nhiều thiết bị tớ. Muốn giao tiếp với thiết bị nào, thiết bị chủ phải gửi đúng địa chỉ để kích hoạt thiết bị đó rồi mới được phép ghi hoặc đọc dữ liệu.

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

3.1 GIỚI THIỆU

Hệ thống thông minh cảnh báo cho xe máy bao gồm các chức năng:

+ Chức năng 1 chế độ cảnh báo trộm: Khi xe được cấp nguồn và bật chế độ chống trộm lên, nếu xe bị rung, mạch sẽ xử lý và gửi 1 tin nhắn “xe của bạn bị rung” về số điện thoại chủ xe nếu là chủ xe thì bỏ qua tin nhắn, còn không phải chủ xe thì thực hiện chức năng thứ 2.

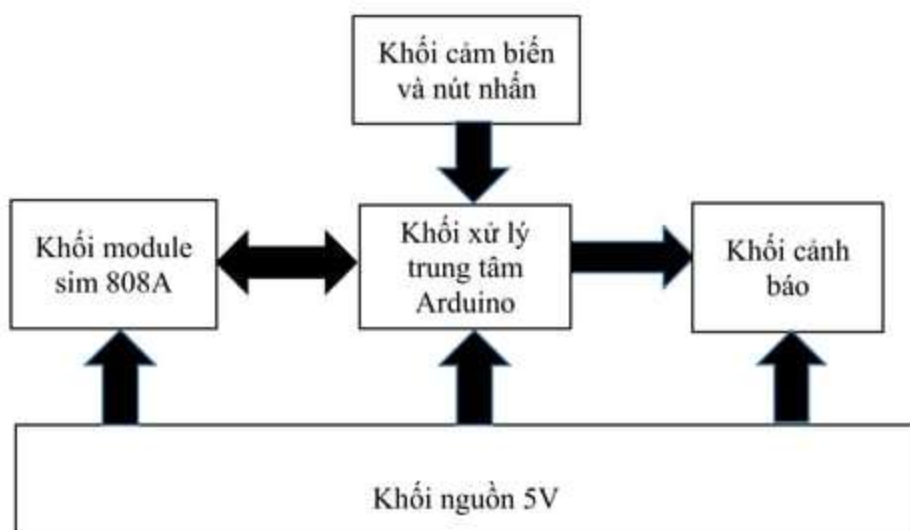
+ Chức năng 2 chống trộm bật, tắt báo động, và định vị xe máy: Khi không phải là chủ xe tác động vào xe thì chủ xe có thể gọi điện thoại đến số điện thoại được gắn trong mạch, mạch nhận cuộc gọi kiểm tra xem có đúng cú pháp không, đúng mạch xử lý và bắt đầu hú còi đồng thời ngắt nguồn xe ngăn không cho xe tiếp tục chạy, và gửi địa chỉ về mạch. Chức năng bật báo động cũng có thể được ứng dụng trong việc tìm xe trong các bãi đỗ xe rộng bằng cách bật báo động xe hú còi qua đó dễ dàng tìm thấy xe khi ở trong bãi đỗ xe.

+ Chức năng 3 giám sát xe và báo tai nạn: Khi xe gặp tai nạn hay bị ngã, mạch sẽ tự động tắt xe và đồng thời nháy xi nhan liên tục sau đó gọi điện thoại về cho người giám sát.

3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

3.2.1 Thiết kế sơ đồ khối hệ thống

Hệ thống thông minh cảnh báo cho xe máy được thiết kế gồm 5 khối kết nối lại với nhau theo nhiều hướng tạo nên một hệ thống hoạt động ổn định được trình bày trong sơ đồ khối hình 3.1 như sau:



Hình 3.1. Sơ đồ khối hệ thống

➤ **Chức năng từng khối:**

- Khối xử lý trung tâm Arduino: dùng để xử lý các tín hiệu vào và xuất tín hiệu ra, điều khiển mọi hoạt động của hệ thống
- Khối module sim: nhận tín hiệu từ vệ tinh gửi về điện thoại vị trí GPS và tín hiệu cảnh báo.
- Khối cảnh báo: Xử lý chuông kêu, đèn sáng khi nhận tín hiệu từ khối Arduino
- Khối nguồn : cung cấp nguồn cho các khối khác.
- Khối cảm biến và nút nhấn: Nút nhấn có chức năng bật/tắt mạch, chế độ chống trộm và reset mạch. Cảm biến khi được tác động sẽ gửi trạng thái cho khối xử lý trung tâm Arduino xử lý.

3.2.2 Tính toán và thiết kế mạch

a. Khối xử lý trung tâm

Arduino UNO sử dụng 3 vi điều khiển chính thuộc họ 8bit AVR là ATmega328, ATmega168, ATmega8. Với 3 dòng VDK này, mạch có thể xử lý những tác vụ đơn giản như điều khiển nhấp nháy đèn LED, thiết lập ứng dụng đo độ ẩm - nhiệt độ sau đó truyền và hiển thị lên LCD , tiếp nhận và xử lý tín hiệu cho ứng dụng điều khiển xe từ xa .. Nó có 16 chân digital I/O, 6 chân đầu vào tương tự

(Analog Inputs), một thạch anh dao động 16 MHz, kết nối USB, một jack cắm điện, một đầu ICSP và một nút reset như trong hình 3.2.



Hình 3.2. Khối xử lý trung tâm sử dụng board Arduino UNO R3

Trong quá trình kết nối các module và lập trình cho hệ thống, tổng số chân I/O sử dụng là 15 chân, công thức tính dòng tiêu thụ I có thể được tính như sau:

$$I = P * I_0 = 15 * 20\text{mA} = 300\text{mA} \quad (3.1)$$

Trong đó, P là số chân và I_0 là dòng ngõ ra mỗi chân I/O.

b. Khối Module Sim 808A



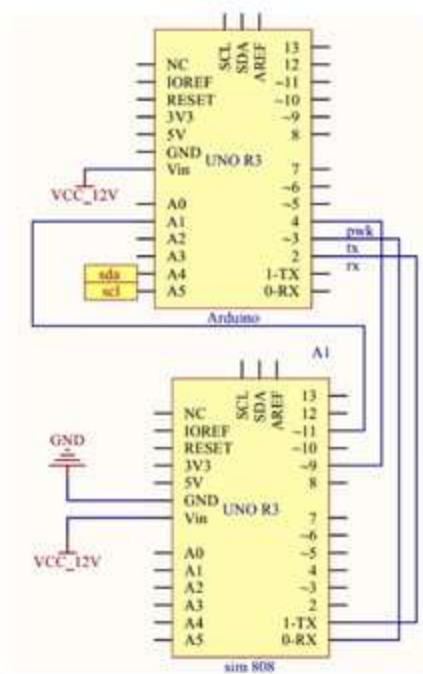
Hình 3.3. Module Sim 808A

Vì chỉ gửi trạng thái các cảm biến và báo vị trí GPS chính xác mà anten bắt được từ vệ tinh nên ta chỉ dùng 8 chân của module sim 808A như hình 3.3 là VCC, GND, TXD, RXD, ST, PWK được kết nối với Arduino Mega như sau:

- Hai chân nguồn VCC và GND được kết nối với Adapter 12V và 2A để dòng điện cho sim hoạt động ổn định lâu dài.

- TXD được nối vào chân RX và RXD nối vào TX của Arduino UNO để truyền nhận dữ liệu theo chuẩn UART.
- Chân PMK được nối với chân số 4 của Arduino. Đây là chân điều khiển bật/tắt Module sim 808A
- Chân ST kết nối với chân A1 của Arduino để đọc trạng thái hoạt động/ nghỉ của module sim 808A.

Hình 3.4 thể hiện sơ đồ nguyên lý kết nối sim vào Arduino UNO R3 và sử dụng nguồn riêng 12V 2A.



Hình 3.4. Sơ đồ nguyên lý kết nối Module Sim 808A vào Arduino UNO R3

c. Khối cảm biến và nút nhấn

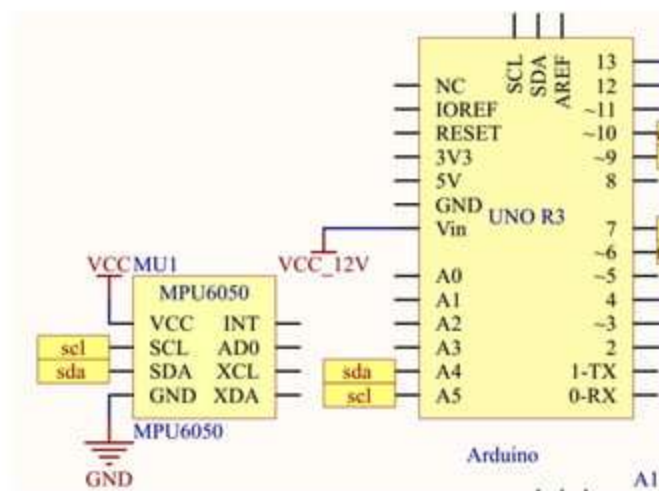
➤ Cảm biến gia tốc MPU-6050

- Chip: MPU-6050 (16bit ADC, 16bit data out)
- Giá trị Gyroscopes trong khoảng: +/- 250 500 1000 2000 degree/sec
- Giá trị Acceleration trong khoảng: +/- 2g, +/- 4g, +/- 8g, +/- 16g
- Giao tiếp: I2C
- Nguồn sử dụng: 3V - 5V (DC)

CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Cách kết nối giữa cảm biến gia tốc MPU-6050 và Arduino UNO R3, được thể hiện trong hình 3.5 với các chân được mô tả sau đây :

- Chân số SCL và SDA nối vào chân của Arduino để truyền nhận dữ liệu và tạo xung. Chân SCL là chân Clock, có tác dụng đồng bộ hóa việc truyền dữ liệu giữa các thiết bị, và việc tạo ra xung clock. Chân SDA là chân truyền dữ liệu.
- Chân GND nối với GND của Arduino
- Chân số 1 nối với nguồn 5V.

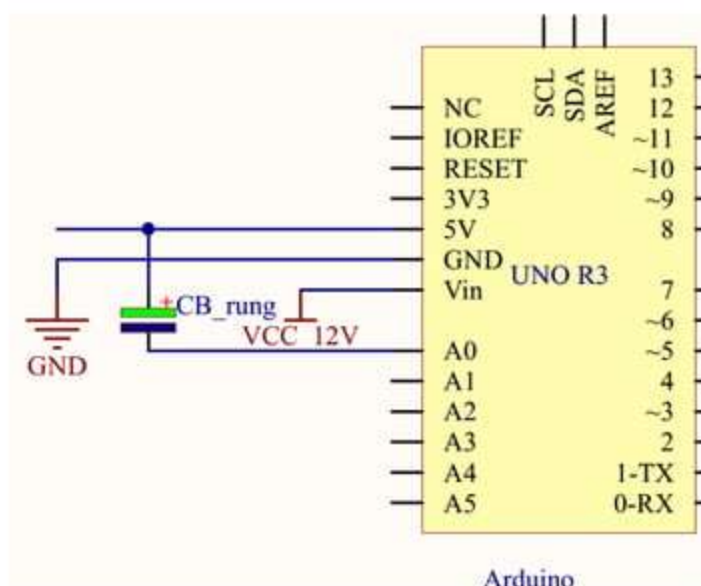


Hình 3.5. Sơ đồ nguyên lý kết nối module MPU-6050 vào Arduino

➤ Cảm biến rung HDX-01

- Điện áp hoạt động: < 24VDC.
- Dòng điện: < 1mA.
- Nhiệt độ: < 80°C.

Cách kết nối giữa cảm biến gia tốc HDX-01 và Arduino UNO R3, được thể hiện trong hình 3.6 sau đây :



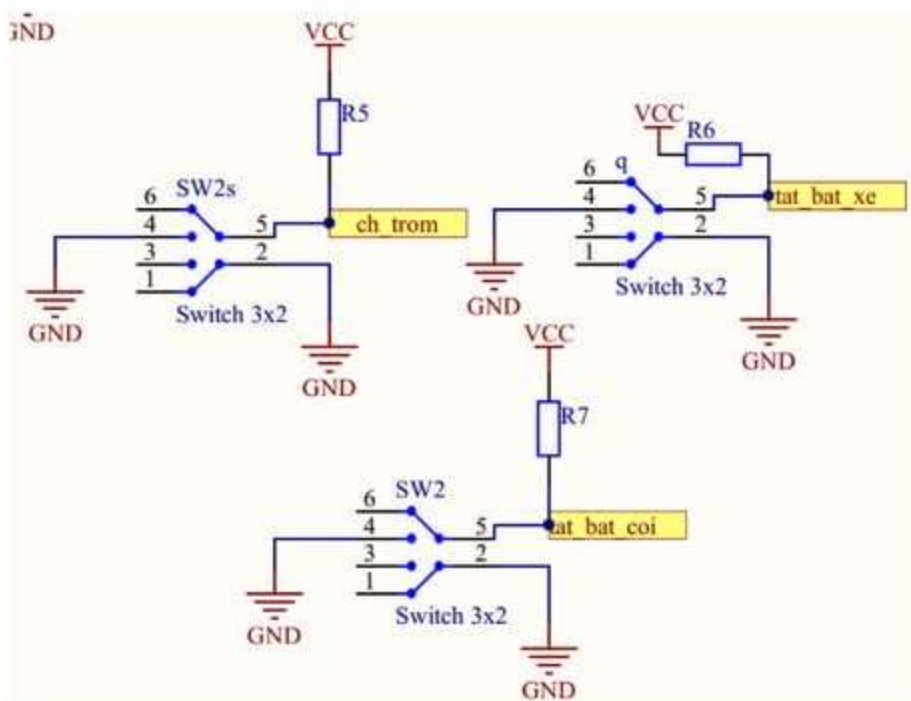
Hình 3.6. Sơ đồ nguyên lý kết nối cảm biến rung HDX-01 vào Arduino

➤ Khối nút nhấn có dòng đi qua là 0.4mA, điện áp đầu vào là 5V. Giá trị điện trở được tính bằng công thức: $R_n = \frac{V}{I_n} = \frac{5V}{0.4mA} = 12k\Omega$ (3.2)

Trong đó, R_n là điện trở của nút nhấn, I_n là cường độ dòng điện qua nút nhấn.

Cách kết nối giữa khối nút nhấn và Arduino UNO R3, được thể hiện trong hình 3.7 với các chân được mô tả sau đây :

- Chân GND nối với GND của Arduino
- Chân số 1 nối với nguồn 5V.
- Các chân ngõ ra của nút nhấn để bật/tắt 3 chế độ chống trộm, xe, còi được nối lần lượt vào các chân 10,9,6.



Hình 3.7. Sơ đồ nguyên lý kết nối của khối nút nhấn

d. Khối nguồn

Theo thông số kỹ thuật, Arduino hoạt động ở điện áp 5V, module hoạt động ở điện áp 5-18V. Vì vậy, nhóm sinh viên sử dụng nguồn adapter 12V 2A có tích hợp ổn áp trên module sim 808A đảm bảo cho arduino và module sim 808A hoạt động ổn định.

e. Khối cảnh báo

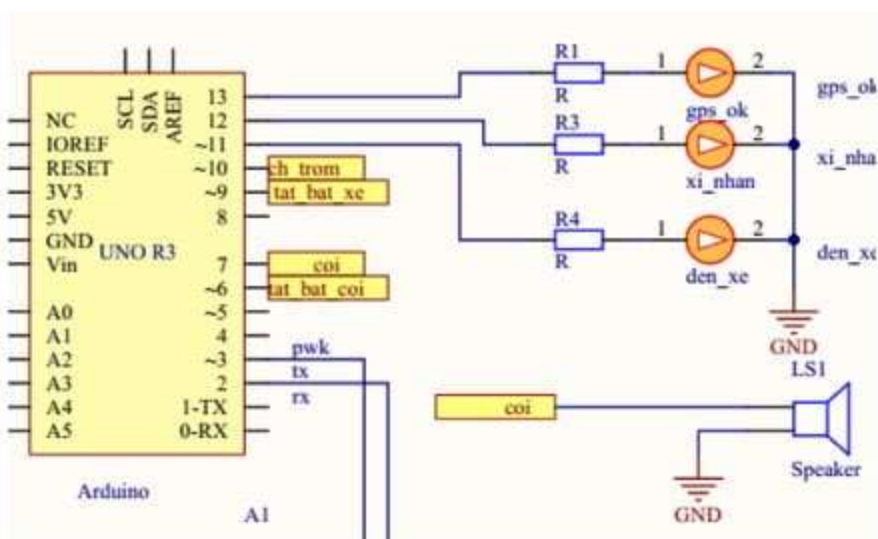
Khối cảnh báo sử dụng led 5mm, hoạt động ở mức 2.7V, dòng qua led hoạt động là 10mA. Điện áp đầu vào là 5V. Giá trị điện trở hạn dòng cho led được tính bằng công thức :

$$R_{led} = (V - V_{led})/I_{led} = (5V - 2.7V)/10mA = 0,23 \text{ k}\Omega = 230 \Omega. \quad (3.3)$$

Chọn giá trị điện trở là 330 Ω - nhằm giảm dòng điện thực tế xuống, giúp giảm độ chói của led và hạn chế việc led bị cháy. Dòng qua led thực tế tính bằng công thức (3.3) :

$I_t = (V - V_{led})/R_t = (5V - 2.7V)/330 \Omega = 7mA$ với I_t , R_t lần lượt là cường độ và điện trở thực tế của led

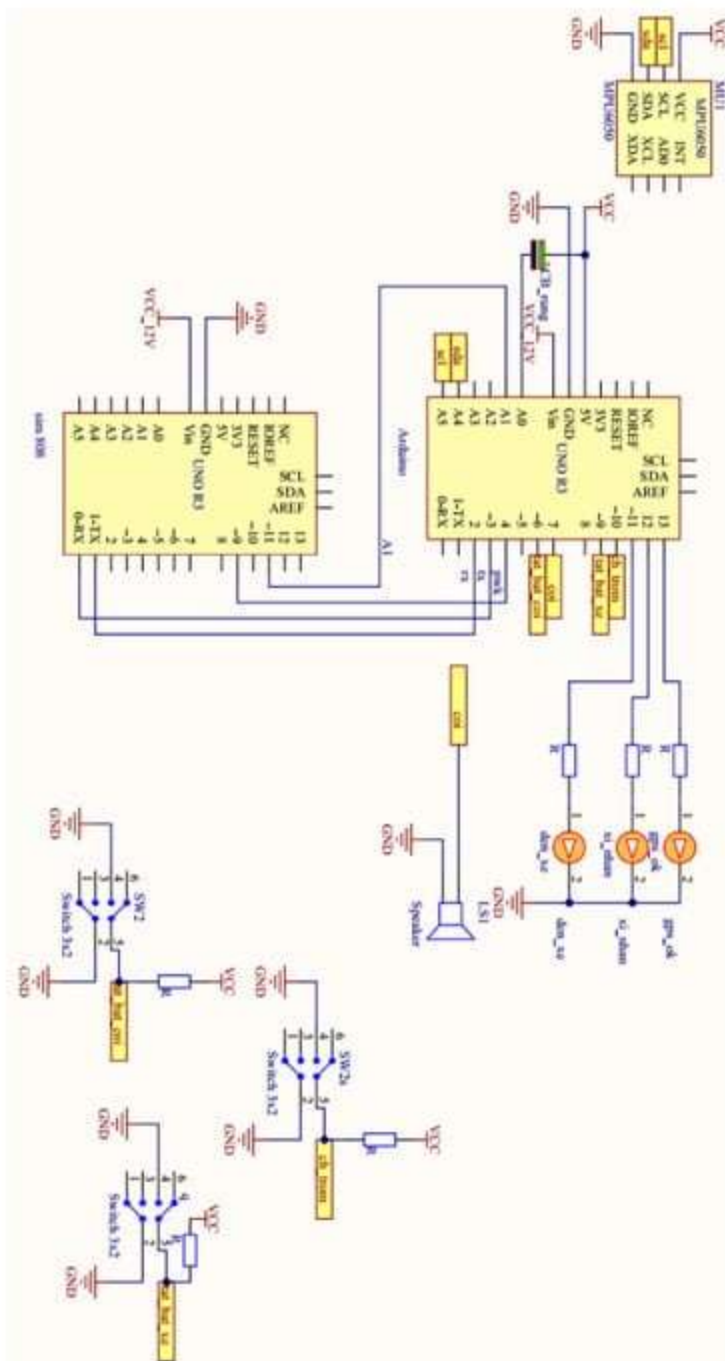
Cách kết nối giữa khối cảnh báo bao gồm led và buzzer với Arduino UNO R3, được thể hiện trong hình 3.8 như sau:



Hình 3.8. Sơ đồ nguyên lý kết nối của khối cảnh báo

3.2.3 Sơ đồ nguyên lý của toàn mạch

Dưới đây là hình 3.9 là sơ đồ nguyên lý toàn mạch thể hiện tất cả các khối và kết nối các thiết bị lại với nhau rồi cắm vào Arduino UNO R3.



Hình 3.9. Sơ đồ nguyên lý toàn mạch

CHƯƠNG 4. THI CÔNG HỆ THỐNG

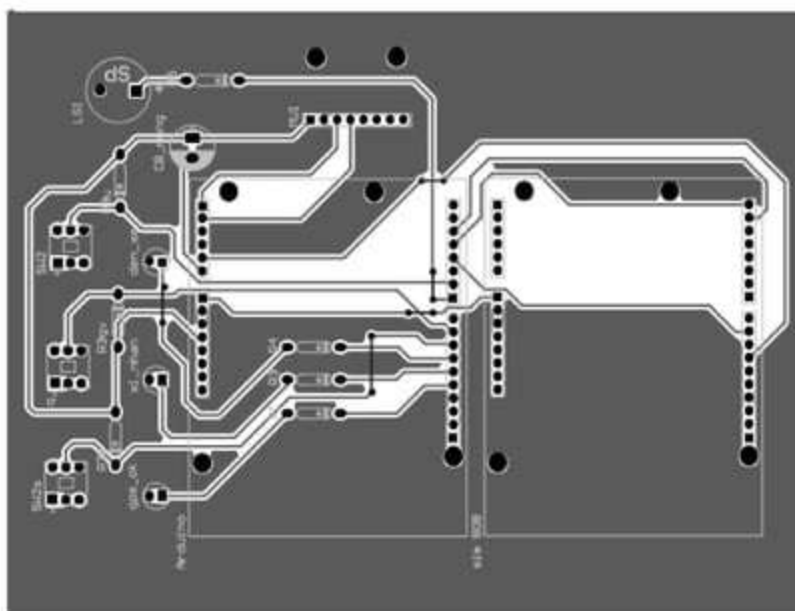
4.1 GIỚI THIỆU

Trong chương này nhóm sẽ trình bày về sơ đồ mạch in lớp trên, lớp dưới, sơ đồ bố trí linh kiện, lắp ráp và kiểm tra, lập trình hệ thống, viết tài liệu hướng dẫn sử dụng.

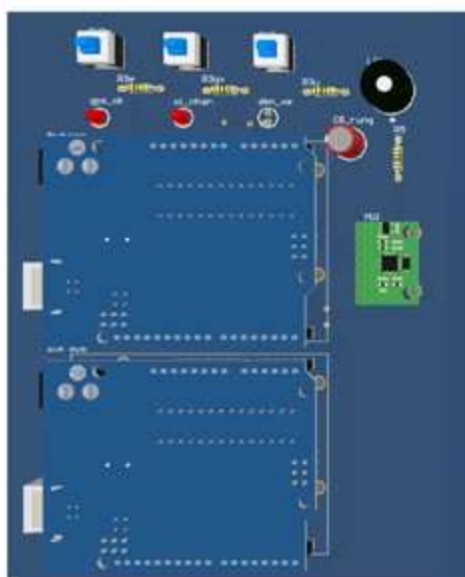
4.2 THI CÔNG HỆ THỐNG

4.2.1 Thi công bo mạch hệ thống

Sau khi thiết kế xong sơ đồ nguyên lý và tiến hành vẽ mạch PCB 1 lớp thủ công. Với kích thước board là 20 x 7 cm.. Hình 4.1 thể hiện sơ đồ đi dây khi đã phù hợp. Hình 4.2 thể hiện sơ đồ hình dạng 3D.



Hình 4.1. Sơ đồ đi dây đã phù hợp đồng bộ mạch hệ thống



Hình 4.2. Hình dạng 3D lớp bo mạch hệ thống

4.2.2 Lắp ráp và kiểm tra bo mạch hệ thống

Quy trình lắp ráp – kiểm tra mạch :

Bước 1: Rửa board đồng sạch sẽ bằng nước rửa mạch sau khi ủi mạch và tiến hành khoan lỗ.

Bước 2: Hàn tất cả các hàng rào vào board đồng. Đo kiểm tra các hàng rào có kết nối với nhau như PCB không.

Bước 3: Gắn board arduino vào mạch vừa hàn xong. Đo kiểm tra từng chân từ arduino ra port đã kết nối hết chưa.

Bước 4: Gắn đầu bus của các module sim 808A, module gia tốc MPU-6050, cảm biến rung vào mạch, kết nối với Arduino. Kết nối còi, nút nhấn, các điện trở. Đo kiểm tra từng chân của các thiết bị đã kết nối hết chưa.

Bước 5: Cấp nguồn 5V vào Arduino và module sim 808A. Dùng đồng hồ đo áp ở ngõ vào và ngõ ra gắn với các module và các cảm biến.

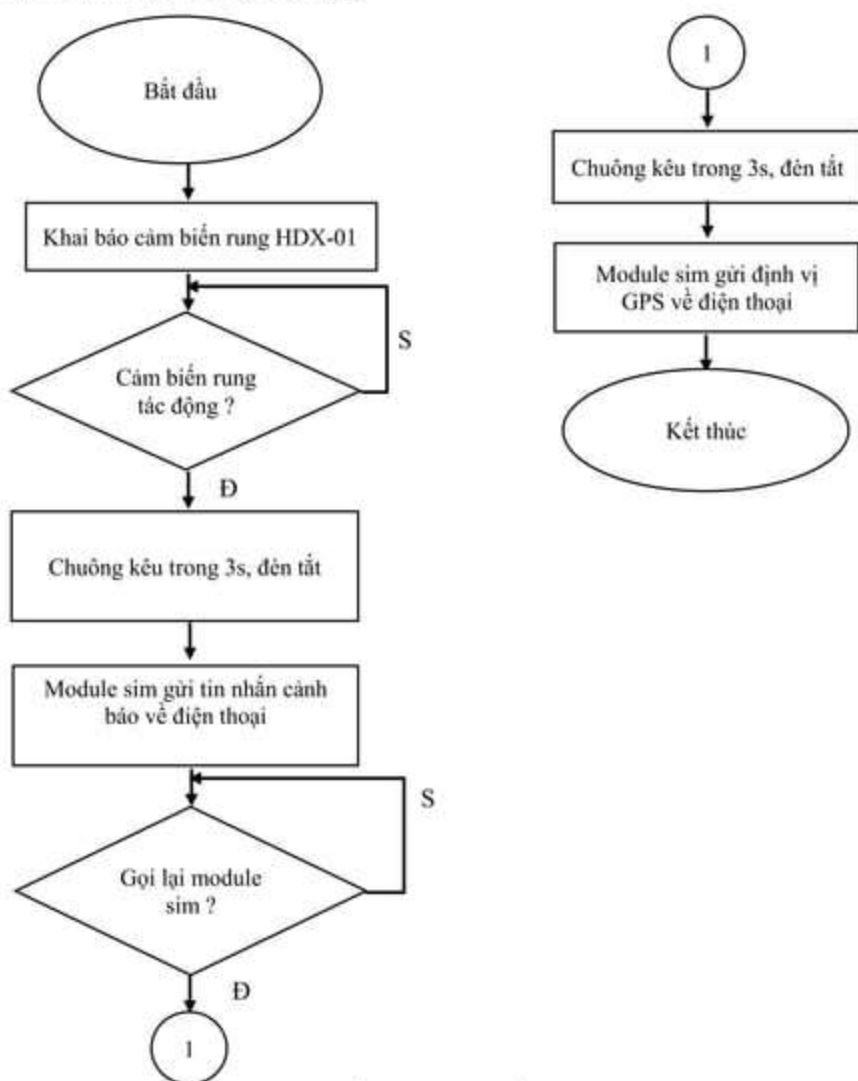
Bước 6: Cuối cùng nạp chương trình và test chương trình có đạt như yêu cầu ban đầu không.

4.3 LẬP TRÌNH HỆ THỐNG

4.3.1 Lưu đồ

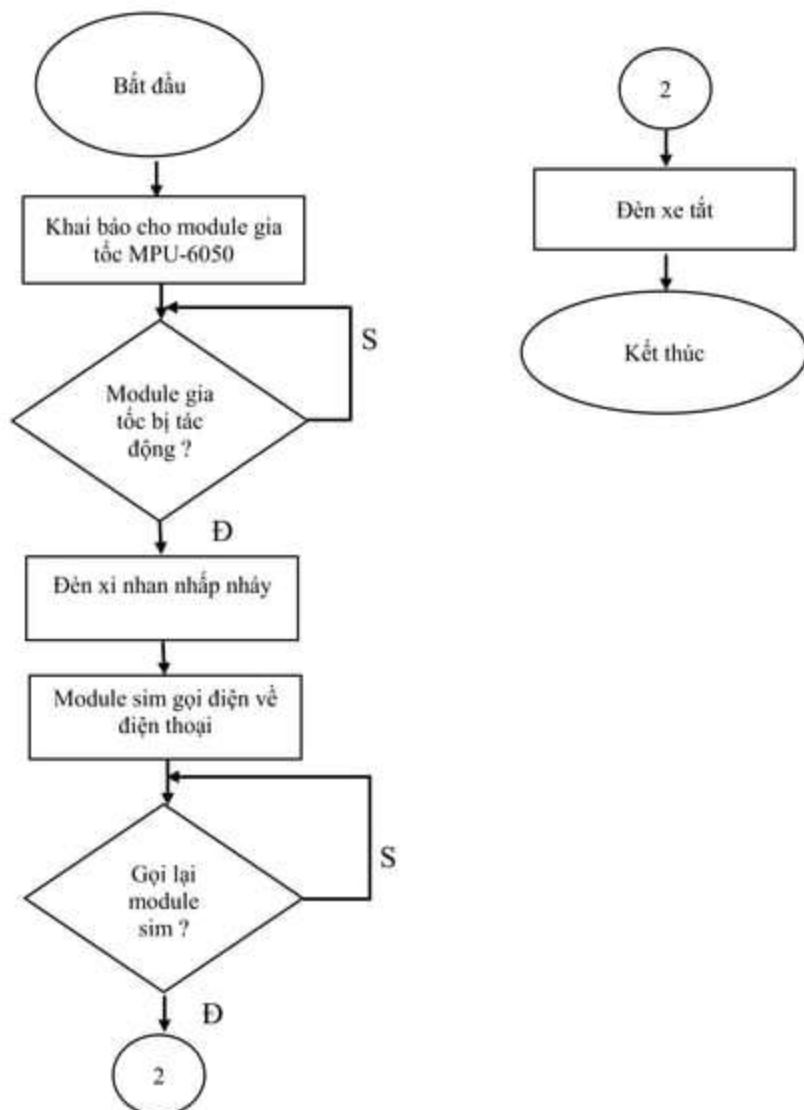
Việc xây dựng lưu đồ là một bước quan trọng trong việc lập trình hệ thống. “Hệ thống thông minh cảnh báo cho xe máy” sẽ gồm 2 lưu đồ con về cảm biến rung và module sim gia tốc và một lưu đồ chính cho toàn mạch.

a. Lưu đồ khi cảm biến rung tác động



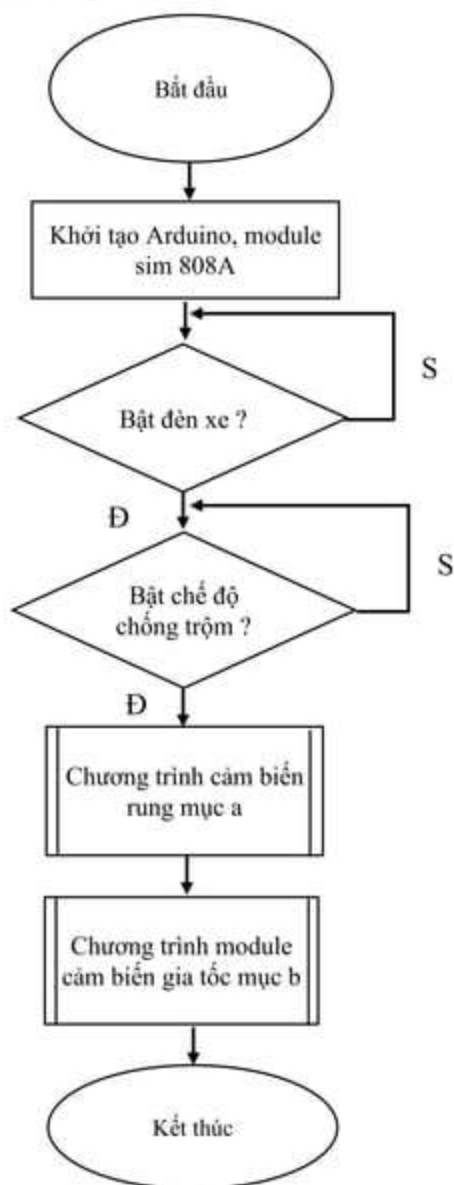
Hình 4.3. Lưu đồ khi cảm biến rung tác động

Khi tác động cảm biến rung HDX-01, chuông sẽ kêu lên trong 3s và đèn sẽ tắt, module sim sẽ gửi tin nhắn cảnh báo về điện thoại, trong trường hợp không tác động thì sẽ không có hiện tượng gì xảy ra. Sau đó, khi gọi lại module sim, chuông sẽ kêu lên trong 3s, đèn tắt, điện thoại sẽ nhận được tin nhắn định vị GPS.

b.Lưu đồ khi module gia tốc tác động**Hình 4.4.** Lưu đồ khi module gia tốc tác động

Khi module gia tốc bị tác động, đèn xi nhan nhấp nháy, module sim gọi điện về điện thoại, nếu không tác động, sẽ không có hiện tượng nào xảy ra. Sau khi module sim gọi điện, nếu gọi lại module sim, đèn xe sẽ tắt.

c. Lưu đồ chính cho chương trình



Hình 4.5. Lưu đồ chính cho chương trình

Khi bật công tắc xe, đồng thời bật công tắc chế độ chống trượt (nếu 1 trong 2 công tắc không được bật thì mạch sẽ không hoạt động), lưu đồ chính sẽ hoạt động theo 2 lưu đồ con khi cảm biến rung và module gia tốc tác động.

4.3.2 Phần mềm lập trình cho vi điều khiển

- Giới thiệu phần mềm lập trình Arduino IDE

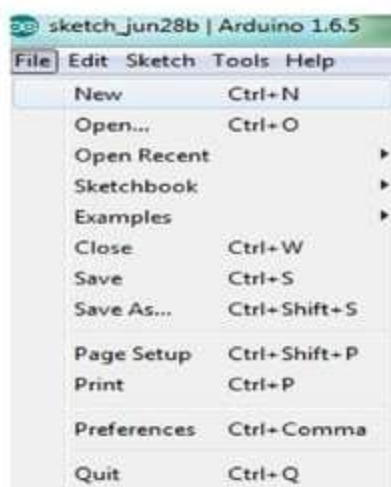
Môi trường phát triển tích hợp Arduino IDE là một ứng dụng đa nền tảng được viết bằng Java, và được dẫn xuất từ IDE cho ngôn ngữ lập trình xử lý và các dự án lắp ráp. Do có tính chất mã nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

Người sử dụng chỉ cần định nghĩa hai hàm để thực hiện một chương trình hoạt động theo chu trình:

setup(): hàm chạy một lần duy nhất vào lúc bắt đầu của một chương trình dùng để khởi tạo các thiết lập.

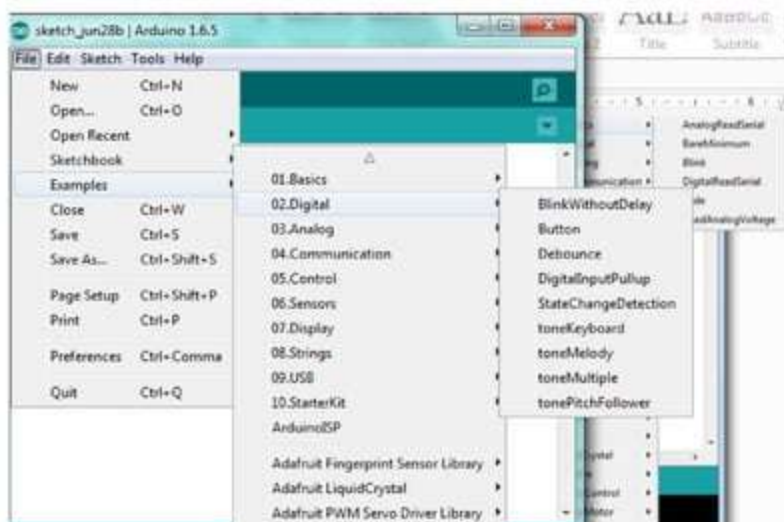
loop(): hàm được gọi lặp lại liên tục cho đến khi bo mạch được tắt.

- File menu:



Hình 4.6. Giao diện file menu arduino IDE

Trong file menu cần quan tâm tới mục Examples, đây là nơi chứa các chương trình mẫu đơn giản như: cách sử dụng các chân digital, analog, sensor.



Hình 4.7. Giao diện Examples menu arduino IDE

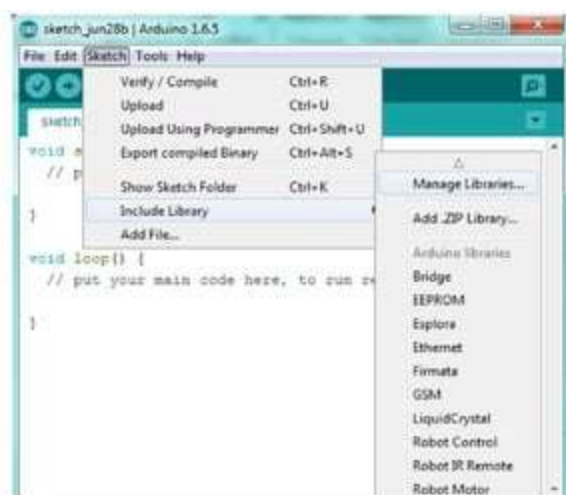
- Sketch menu:

Verify/ Compile: chức năng kiểm tra lỗi code.

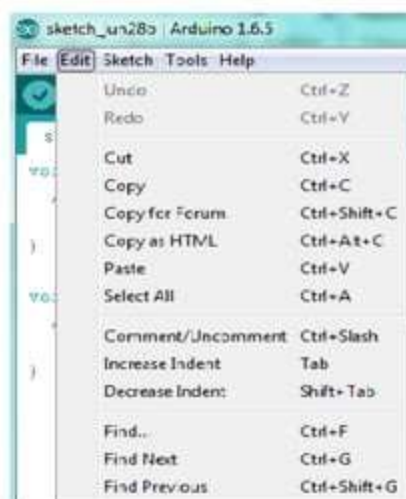
Show Sketch Folder: hiển thị nơi code được lưu.

Add File: thêm vào một Tập code mới.

Import Library: thêm thư viện từ bên ngoài cho IDE.



Hình 4.8. Giao diện Sketch Menu Arduino IDE



Hình 4.9. Giao diện edit menu arduino IDE

- Tool menu:



Hình 4.10. Giao diện Tool Menu Arduino IDE

Trong Tool menu ta quan tâm các mục Board và Serial Port. Trong mục Board, cần phải lựa chọn board mạch cho phù hợp với loại board sử dụng. Nếu sử dụng loại board khác thì phải chọn đúng loại board mà mình đang có, nếu sai thì khi Upload chương trình vào chip sẽ bị lỗi. Nếu là Arduino UNO R3 thì phải chọn như hình 4.11:



Hình 4.11. Board Arduino sử dụng

Serial Port: đây là nơi lựa chọn cổng Com của Arduino. Khi chúng ta cài đặt driver thì máy tính sẽ hiện thông báo tên cổng Com của Arduino là bao nhiêu, ta chỉ việc vào Serial Port chọn đúng cổng Com để nạp code, nếu chọn sai thì không thể nạp code cho Arduino được.

Arduino Toolbar có một số button và chức năng của chúng như sau:



Hình 4.12. Arduino Toolbar

- Verify (1): kiểm tra code có lỗi hay không.
- Upload (2): nạp code đang soạn thảo vào Arduino.
- New, Open, Save (3): Tạo mới, mở và lưu sketch.
- Serial Monitor (4): Đây là màn hình hiển thị dữ liệu từ Arduino gửi lên máy tính.

4.4 VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC

Bước 1: Cấp nguồn cho hệ thống, hệ thống sử dụng nguồn là 220V DC, đã có sẵn nguồn adapter 12V 2A. Đợi hệ thống khởi động xong, led màu vàng sẽ sáng nhấp nháy cho biết đã có tín hiệu GPS.

Bước 2: Nhấn nút nhấn thứ 2, đèn màu xanh sáng lên báo hiệu bật đèn xe.

Bước 3: Ta test hệ thống báo tai nạn. Khi nghiêng mạch đi 1 góc nghiêng, mạch sẽ có 1 cuộc gọi tới điện thoại, khi ta gọi lại thì đèn xe và hệ thống xe máy sẽ tắt và đèn xe nhấp nháy.

Bước 4: Tiếp theo, ta test chế độ chống trộm. Nhấn tắt/bật đèn xe, nhấn nút nhất thứ nhất, đèn màu vàng sáng lên báo hiệu đã bật hệ thống chống trộm.

Bước 5: Khi tác động cảm biến rung, còi sẽ kêu lên, đồng thời đèn xi nhan màu xanh sẽ sáng nhấp nháy, gửi tin nhắn “Xe bạn đang bị rung” về số điện thoại đã cài đặt trước. Khi điện thoại nhận được tin nhắn, nhấn gọi lại số điện thoại cài đặt cho module sim thì chuông sẽ kêu và gửi tọa độ GPS về tin nhắn điện thoại.

- Lưu ý: Các nút nhấn đã được chú thích ở chương sau.

CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT - ĐÁNH GIÁ

5.1 KẾT QUẢ

Sau thời gian tiến hành tìm hiểu, nghiên cứu các tài liệu chuyên ngành, tìm hiểu thêm qua mạng Internet, tổng hợp lại các kiến thức đã được học trong suốt 4 năm cũng như được sự hướng dẫn của thầy GVHD Ts.Nguyễn Thanh Hải. Nhóm chúng em cũng đã hoàn thành được đồ án tốt nghiệp với đề tài “THIẾT KẾ HỆ THỐNG THÔNG MINH CẢNH BÁO CHO XE MÁY”.

Sau đề tài đồ án này, nhóm em cũng đã nghiên cứu và tích lũy được thêm nhiều hiểu biết, kiến thức mới như:

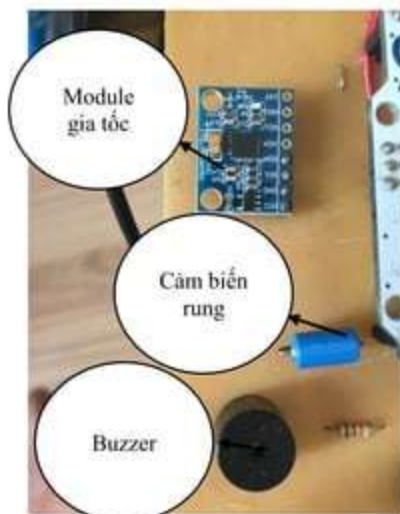
- Hiểu biết sâu hơn về sử dụng và các tính năng của Arduino như giao tiếp giữa Arduino với các module mở rộng như: module cảm biến rung HDX-01, module Sim 808A.
- Nghiên cứu và biết cách kết nối giữa Arduino với các module cảm biến rung HDX-01, module Sim 808A.
- Nghiên cứu biết được cách sử dụng của cảm biến rung HDX-01, nguyên lý hoạt động của cảm biến, các thông số kỹ thuật, các tính năng của cảm biến.
- Nghiên cứu biết được cách sử dụng module Sim 808A, nguyên lý hoạt động, các thông số kỹ thuật, tính năng của Sim 808A. Biết được cách thiết lập cho module sim, sử dụng tính năng nhắn tin SMS đến điện thoại và gửi dữ liệu GPS trên module Sim để điều khiển thiết bị.
- Biết cách sử dụng thêm phần mềm vẽ mạch Altium để thiết kế mạch in, làm mạch kết nối giữa kit Arduino với các module mở rộng để giảm sử dụng các dây cắm và các linh kiện gắn rời nhằm tăng tính nhỏ gọn cho mạch điều khiển.

Mô tả:

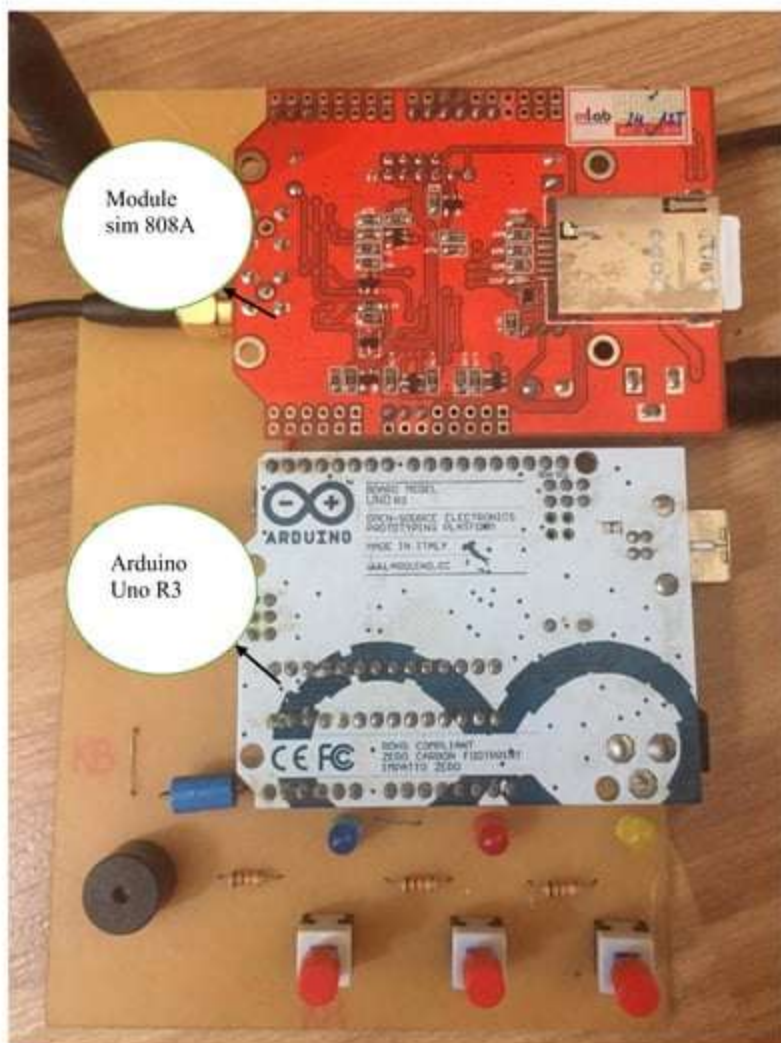
- Sau đây là hình ảnh thực tế của mạch.



Hình 5.1. Hình thực tế khối nút nhấn và cảnh báo



Hình 5.2. Hình thực tế khối cảm biến



Hình 5.3. Hình thực tế gắn linh kiện cho mặt trên

Trong hình 5.3 hình ảnh thực tế lắp trên có các hàng rào dùng để kết nối với các thiết bị và module như: cảm biến rung HDX-01, module gia tốc MPU-6050, module Sim 808A, Arduino, nút nhấn.

Khi cấp nguồn xong, nhấn nút nhấn bật đèn xe và bật chế độ chống trộm, led màu xanh (đèn xe) và led màu vàng (chế độ chống trộm) sẽ sáng lên như hình 5.4



Hình 5.4. Trạng thái led cho biết trạng thái của mạch

Hình 5.5 thể hiện thao tác tác động lên cảm biến rung, bằng cách gõ vào cảm biến nhiều lần.



Hình 5.5. Thao tác với cảm biến rung

Hình 5.6 thể hiện thao tác tác động lên module gia tốc, chỉ cần nghiêng mạch 1 góc 45° và chế độ cảnh báo tai nạn sẽ được kích hoạt.



Hình 5.6. Thao tác với module gia tốc

Khi nhận được tín hiệu GPS, điện thoại sẽ nhận được tin nhắn như hình 5.7. Việc bị trùng lặp địa chỉ dẫn tới việc lỗi dư phần chữ CARRIER. Cách khắc phục là sao chép tọa độ 10.850817,106.771730 và nhập vào phần địa chỉ trên Google Map như hình 5.8 và có được địa chỉ GPS của xe.

**Xe Ban Hien Tai O : [https://maps.google.com/maps?q=](https://maps.google.com/maps?q=10.850817,106.771730)
10.850817,106.771730CARRIE
R**

Hình 5.7. Tọa độ GPS mà module sim gửi về điện thoại



Hình 5.8. Tra cứu địa chỉ với tọa độ nhận được

Việc kiểm tra, chạy thử nghiệm trong phòng kín, cao tầng cũng có thể gây nhiễu tín hiệu GPS, dẫn tới việc không thể xác định vị trí chính xác nhất như hình 5.9

```
,161,"",0  
> G  
  
+CLIP: "0968987377",161,"",0  
>
```

Hình 5.9. Lỗi thường gặp khi anten không bắt được tín hiệu

Khi cảm biến rung được tác động, điện thoại sẽ nhận được tin nhắn như hình 5.6.

XE BAN DANG BI RUNG

Hình 5.10. Tin nhắn cảnh báo chống trộm

5.2 NHẬN XÉT – ĐÁNH GIÁ

Sau quá trình vận hành thử hệ thống, nhóm thực hiện đã thu về những số liệu như bảng sau:

Bảng 5.1. Số liệu thực nghiệm

| | Số lần thử nghiệm | Số lần thành công | Hiệu suất (%) | Đánh giá | Ghi chú |
|-----------------------------|-------------------|-------------------|--------------------------------------|-----------|---------|
| Định vị GPS | 30 | 6 | 20% | Không đạt | |
| Hoạt động của các cảm biến | 30 | 30 | 100% | Đạt | |
| Hệ thống cảnh báo còi, led. | 30 | 30 | 100% | Đạt | |
| Tin nhắn báo về điện thoại | 30 | 30 | 100% | Đạt | |
| Đánh giá chung | | | Hoàn thành được 75% mục tiêu ban đầu | | |

***Chú thích: Trên 70% được xem đạt.**

Qua những số liệu bảng trên, mô hình đã đáp ứng được 75% mục tiêu ban đầu. Hệ thống dễ dàng kiểm tra, và gửi tin nhắn về điện thoại đã kết nối, các cảm biến hoạt động tốt, hệ thống cảnh báo còi, led hoạt động ổn định. Người dùng thao tác một cách đơn giản, dễ sử dụng. Hệ thống đảm bảo an toàn. Hệ thống không đạt 25% mục tiêu ban đầu do anten nhận được tín hiệu yếu, không đủ để định vị chính xác vị trí kinh độ, vĩ độ, dẫn tới việc xác định vị trí GPS không thành công.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 KẾT LUẬN

Sau thời gian tìm hiểu, nghiên cứu và thực hiện mô hình, nhiệm vụ đề tài cơ bản được hoàn thành. Bằng sự cố gắng của bản thân, học hỏi từ Thầy Cô, bạn bè, nhóm đã hoàn thành đề tài “Thiết kế hệ thống thông minh cho cảnh báo xe máy”.

Đề tài đã hoàn thành cụ thể:

- Kết quả thi công hoàn thiện.
- Hệ thống chạy ổn định, hoàn thành 75% mục tiêu ban đầu.
- Độ phản hồi của hệ thống cảnh báo nhanh.
- Tương tác giữa điện thoại và module sim ổn định

Quá trình thực hiện đề tài đã đạt được những kết quả như mục tiêu ban đầu đề ra, tuy nhiên còn gặp nhiều khó khăn và hạn chế. Tín hiệu GPS vẫn chưa được định vị ổn định và chính xác được.

6.2 HƯỚNG PHÁT TRIỂN

Từ đề tài “**Hệ thống thông minh cảnh báo cho xe máy**” ta có thể phát triển theo các hướng nhằm đảm bảo sự ổn định, độ bền và tính thực tế cho sản phẩm sau:

- Hệ thống xe máy chống trộm và cảnh báo tai nạn này có thể mở rộng khả năng điều khiển bằng Internet.
- Vị trí GPS có thể theo dõi trực tiếp chứ không phải chỉ gửi được 1 lần.

TÀI LIỆU THAM KHẢO

Sách tham khảo

- [1] Phạm Quang Huy, Lê Cảnh Trung, “Lập trình điều khiển với Arduino”, NXB Khoa học và kỹ thuật, 2014.
- [2] Vũ Văn Quý, “Mạch chống trộm xe máy và thông báo cho người thân qua điện thoại định vị GPS module sim 808”, Đồ án tốt nghiệp, trường ĐH Công nghệ GTVT, 2017.
- [3] Trần Thu Hà, “Điện tử cơ bản”, NXB ĐH Quốc Gia Tp.HCM, 2013.

Trang web tham khảo

- [1] Minh Quang, “Việt Nam tiêu thụ xe máy đứng thứ 4 thế giới”, 2017.
“<https://baomoi.com/viet-nam-tieu-thu-xe-may-dung-thu-4-the-gioi/c/21318042.epi>”
- [2] Wikipedia, “Hệ thống định vị toàn cầu”, 2018.
“https://vi.wikipedia.org/wiki/Hệ_thống_định_vị_toàn_cầu”
- [3] Thanh Sơn, “Hướng dẫn sử dụng Module sim 808 GSM/GPRS/GPS”, 2017.
“<http://mlab.vn/huong-dan-su-dung-module-Sim808.html>”
- [4] Vũ Quang Huy, “Thông số kỹ thuật Arduino UNO R3 – Các biến thể và lưu ý”, 2015.
“<https://www.stdio.vn/articles/thong-so-ki-thuat-arduino-uno-r3-va-cac-luu-y-400>”.
- [5] Hack Não, “Cách sử dụng cảm biến rung với Arduino”, 2015.
“<http://arduino.vn/bai-viet/557-cach-su-dung-cam-bien-rung-voi-arduino>”
- [6] Arvind Sanjeev, “A review of basic IMU sensors that work with Arduino, and how to interface Arduino with the best sensor available”, 2018.
“<https://maker.pro/arduino/tutorial/how-to-interface-arduino-and-the-mpu-6050-sensor>”.

DANH SÁCH TỪ VIẾT TẮT

| Tên từ viết tắt | Tên đầy đủ |
|-----------------|---|
| DATN | Đồ Án Tốt Nghiệp |
| GPS | Global Positioning System |
| GPRS | General Packet Radio Service |
| UART | Universal Asynchronous Receiver – Transmitter |
| GSM | Global System for Mobile Communications |
| PCB | Printed circuit board |
| VĐK | Vi Điều Khiển |

PHỤ LỤC

Code chương trình cho board Arduino:

```
#include <SoftwareSerial.h>
#include<Wire.h>
const int MPU_addr = 0x68; // I2C address of the MPU-6050
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
//////////
#define bat_xe 9
#define coi 7
#define bat_coi 6
#define chong_trom 10
#define led_xe 11
#define led_xinhan 12
#define led_gps 13
int button = 0;
int button1 = 0;
//////////
// Buttons
#define btnRIGHT 0
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
#define btnSELECT 4
#define btnNONE 5

// RMC
#define GPS_RSTS 0
```

```
#define GPS_FSTS      1
#define GPS_TIME      2
#define GPS_LAT       3
#define GPS_LON       4
#define GPS_MSL       5
#define GPS_SPEED     6
#define GPS_ORG       7
```

```
typedef union
{
    struct
    {
        unsigned int year;
        unsigned char month;
        unsigned char date;
        unsigned char hour;
        unsigned char minute;
        unsigned char sec;
    } time;
} SysTime;
```

```
typedef enum
{
    DISPLAY_GPS = 0,
    DISPLAY_RTC,
    DISPLAY_SPEED_ORG,
    DISPLAY_MAX
} DISPLAY_E;
```

```
typedef struct {  
    bool State_gps;    /*!< trạng thái gps 1 = có tín hiệu, 0 = mất tín hiệu */  
    SysTime time_gps; /*!< time gps */  
    float Lat;    /*!< vĩ độ */  
    float Lng;    /*!< kinh độ */  
    float SpeedGPS; /*!< Tốc độ GPS */  
    float OrGPS;    /*!< Hướng GPS 360 độ */  
} GPS_T; /*!< cấu trúc dữ liệu hành trình */  
  
int phone_number_to_send = "0968987377";  
int PWK_PIN = 4;    // the number of the PWK pin  
int STATUS_PIN = A1; // select the input pin for detect module power-on  
int ab = 0, test = 0;  
  
SoftwareSerial modemSerial(3, 2); // RX, TX  
  
GPS_T g_gps_data = {false, 0, 0, 0, 0};  
  
int button_pressed, tmp_button, table_display = 0;  
void setup() {    // chương trình thiết lập chân input hoặc output  
    ///////////  
  
    pinMode(bat_xe, INPUT);  
    pinMode(coi, OUTPUT);  
    pinMode(bat_coi, INPUT);
```

```
pinMode (chong_trom, INPUT);
pinMode (led_xe, OUTPUT);
pinMode (led_xinhan, OUTPUT);
pinMode (led_gps, OUTPUT);
/////////

Serial.begin (9600);
modemSerial.begin(9600);
// initialize serial communications
// Serial.begin(115200);

// set up the LCD's number of columns and rows:
// lcd.begin(16, 2);
// Print a message to the LCD.
Serial.write("SIM808 DEMO");
Serial.println("ok");
/////////

Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0);    // set to zero (wakes up the MPU-6050)
Wire.endTransmission(true);
/////////
}

char ch;
char respond[300];
int new_gps = 0;
int dk1 = 0;
int kt_ok = 0;
int gps_ok = 0;
```

```
int rung_ok = 0, i = 0;
int chongtrom = 0, nga = 0;
int xe_ok = 0;
void loop() // chương trình chính bắt đầu tại đây
{

    if (kt_ok == 0) // kiểm tra kết nối gps
    {
        ModemInit();
    }
    /////

    update_gps(); // update địa chỉ gps
    display_gps(); // hiển thị gps
    //////////

    if (xe_ok == 0) // kiểm tra xem xe khởi động chưa
    {
        khai_dong();
    }
    ////////////

    if (xe_ok == 1)
    {
        ////////////

        if (nga == 0) // kiểm tra xe ngã
        {

            mpu(); // khai báo chương trình mpu6050
            kt_nut();
```

```
khởi_dong();
if (AcX > 10000 || AcX < 0) // nếu xe nghiêng toạ độ trục x > 10000
{
    digitalWrite(led_xe, LOW);
    gọi();
    nga = 1;

}
////
if (chongtrom == 0)
{

    dkien1(); // kiểm tra có cuộc gọi đến
    rung(); // kiểm tra xe bị rung
}
}
//////////

if (nga == 1) // nếu xe bị ngã
{
    digitalWrite(led_xe, LOW);
    delay(100);
    digitalWrite(led_xinhan, HIGH);
    delay(1000);
    digitalWrite(led_xinhan, LOW);
    delay(1000);
    byte button = digitalRead(bat_xe);
    if (button == HIGH)
    {
```

```
    xe_ok = 0;
    nga = 0;
    dk1 = 0;
}

}

//////////

}

}

void kt_nut() // chương trình kt các nút nhấn
{

    byte button3 = digitalRead(chong_trom);
    if (button3 == LOW)
    {
        chongtrom = 0;
        digitalWrite(led_gps, HIGH);
        rung_ok = 0;

    }
    else {
        chongtrom = 1;
        digitalWrite(led_gps, LOW);
    }
}

void rung() // chương trình đọc cảm biến rung
```



```
{

    int value = analogRead(A0);
    Serial.println(value);
    delay(20);
    if (value < 900)
    {
        digitalWrite(coi, HIGH);
        delay(4000);
        digitalWrite(coi, LOW);
        char data_to_send_sms[160];
        sprintf(data_to_send_sms, "XE BAN DANG BI RUNG");
        SendSMS(phone_number_to_send, data_to_send_sms);
        Serial.println("send_ok_ok");
        delay(3000);

    }

}

void mpu() // chương trình lấy dữ liệu từ cảm biến mpu6050
{
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr, 14, true); // request a total of 14 registers
    AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C
    (ACCEL_XOUT_L)
```

```
AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E
(ACCEL_YOUT_L)
AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40
(ACCEL_ZOUT_L)
Tmp = Wire.read() << 8 | Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42
(TEMP_OUT_L)
GyX = Wire.read() << 8 | Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44
(GYRO_XOUT_L)
GyY = Wire.read() << 8 | Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46
(GYRO_YOUT_L)
GyZ = Wire.read() << 8 | Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48
(GYRO_ZOUT_L)
Serial.println("AcX = "); Serial.print(AcX);
Serial.print(" | AcY = "); Serial.print(AcY);
Serial.print(" | AcZ = "); Serial.print(AcZ);
//Serial.print(" | Tmp = "); Serial.print(Tmp/340.00+36.53); //equation for temperature
in degrees C from datasheet
//Serial.print(" | GyX = "); Serial.print(GyX);
//Serial.print(" | GyY = "); Serial.print(GyY);
//Serial.print(" | GyZ = "); Serial.println(GyZ);
delay(333);
}

void khoi_dong() // chương trình iem tra khoi dong xe
{
    byte button = digitalRead(bat_xe);
    if (button == LOW)
    {
        digitalWrite(led_xe, HIGH);
        xe_ok = 1;
    }
}
```

```
}  
else {  
    digitalWrite(led_xe, LOW);  
    xe_ok = 0;  
}  
/////////  
  
byte button1 = digitalRead(bat_coi);  
if (button1 == LOW)  
{  
    digitalWrite(coi, HIGH);  
}  
else {  
    digitalWrite(coi, LOW);  
}  
}  
  
void dkien1() // chng trnh kiem tra cuoc goi den va gui dia xh xe ve  
{  
    if (dk1 == 0)  
    {  
  
        while (modemSerial.available() > 0)  
        {  
            ch = modemSerial.read();  
            Serial.print(ch);  
  
            if (ch == '9')  
            {
```

```
digitalWrite (led_xe, LOW);
digitalWrite (coi, HIGH);
delay(5000);
digitalWrite (coi, LOW);
delay(5000);
gui_tin();
dk1 = 1;
nga = 1;
}
delay(3);
}
}
}

void display_gps()
{
char float_str[10];
if (g_gps_data.State_gps == true)
{

dtostrf(g_gps_data.Lat, 10, 6, float_str);
if (gps_ok == 0)
{
Serial.println(float_str);

dtostrf(g_gps_data.Lng, 10, 6, float_str);
Serial.println( float_str);

digitalWrite(led_gps, HIGH);
```

```
    delay(1000);
    digitalWrite(led_gps, LOW);
    delay(700);
    digitalWrite(led_gps, HIGH);
    delay(1000);
    digitalWrite(led_gps, LOW);
    delay(700);
    digitalWrite(led_gps, HIGH);
    delay(1000);
    digitalWrite(led_gps, LOW);
    delay(700);
    gps_ok = 1;
}
}
}

void gui_tin() // chương trình gui tin nhan di
{
    char data_to_send_sms[160];
    char tmp_str1[20], tmp_str2[20];
    dtostrf(g_gps_data.Lat, 10, 6, tmp_str1);
    dtostrf(g_gps_data.Lng, 10, 6, tmp_str2);
    sprintf(data_to_send_sms, "Xe Ban Hien Tai O :
https://maps.google.com/maps?q=%s,%s", tmp\_str1, tmp\_str2\);
    SendSMS\(phone\_number\_to\_send, data\_to\_send\_sms\);
    Serial.println\("send\_ok\_ok"\);
}

void goi \(\) // chương trình thực hiện cuộc gọi
{
    modemSerial.write\("AT+CLIP=1\r"\);
```

```
modemSerial.write("ATD0968987377;\r");
Serial.write("ATD0968987377x'\r");
delay(15000);
modemSerial.write("ATH\r");
Serial.write("ATH\r");
}
void update_gps()
{
    static unsigned long previousMillis;
    unsigned long currentMillis;

    // Update GPS data
    currentMillis = millis();
    if (currentMillis - previousMillis >= 2000) {
        modemSerial.write("AT+CGNSINF\r");
        if (readSerialFrame(respond) > 0)
        {
            if (strcmp(respond, "\r\n+CGNSINF: ", 12) == 0)
            {
                NMEA_GPRMC_Decoder(&g_gps_data, respond + 12);
            }
        }
        previousMillis = currentMillis;
    }
    // End Update GPS data
}
/*
    Read input serial
*/
```

```
int readSerial(char result[], int time_out) {
    int i = 0;
    unsigned long currentMillis, previousMillis;
    previousMillis = millis();
    currentMillis = previousMillis;
    while (1) {
        if (currentMillis - previousMillis >= time_out)
        {
            return 0;
        }
        else
        {
            while (modemSerial.available() > 0) {
                char inChar = modemSerial.read();
                if (inChar == '\r') {
                    result[i] = '\0';
                    modemSerial.flush();
                    return i;
                }
                if (inChar != '*') {
                    result[i] = inChar;
                    i++;
                }
            }
        }
        currentMillis = millis();
    }
}
```

```
bool SendSMS(char *phone_num, char *data)
{
    unsigned long currentMillis, previousMillis;
    char sms_cmd[50];
    char respond[30];
    char command_en[2] = {0x1A, 0};

    sprintf(sms_cmd, "AT+CMGS=\"%s\"", phone_num);
    modemSerial.write(sms_cmd);
    if (readSerialFrame(respond) > 0)
    {
        Serial.println(respond);
        if (strcmp(respond, "\r\n>") != 0)
        {
            modemSerial.write(data);
        }
    }
    modemSerial.write(command_en);
    modemSerial.write(0x1A);
    Serial.println("send ok");
    previousMillis = millis();
    currentMillis = previousMillis;
    while (currentMillis - previousMillis < 12000) // timeout 5s
    {
        currentMillis = millis();
        if (readSerialFrame(respond) > 0)
        {
            Serial.println(respond);
        }
    }
}
```



```
if (strcmp(respond, "\r\nOK\r\n") != 0)
{

    digitalWrite (led_xinhan, HIGH);
    delay (3000);
    digitalWrite (led_xinhan, LOW);

    return false;
}
}
}

delay (2000);
return false;
}

void ModemInit(void) // chương trình thiết lập giao tiếp giữa arduino với sim808
{
    static bool is_modem_init_ok = false;
    int status_pin_level = 0; // to detect modem has been turn-on
    // Turnon modem:
    try_on:
    status_pin_level = analogRead(STATUS_PIN);
    if (status_pin_level < 500) // Check module on ? if not turn-off
    {

        is_modem_init_ok = false;
        Serial.write("No Sim808");
        digitalWrite(PWK_PIN, HIGH);
    }
}
```

```
delay(2000);
digitalWrite(PWK_PIN, LOW);
delay(2000);
goto try_on;          // We try turn-on until module SIM ready too use
}

if (is_modem_init_ok == false)
{

    Serial.write("Initting...");
    modemSerial.write("ATE0\r");
    if (readSerialFrame(respond) > 0)
    {
        if (strcmp(respond, "\r\nOK\r\n") != 0)
        {
            goto try_on;
        }
    }
    else
    {
        goto try_on;
    }

    modemSerial.write("AT+CMGF=1\r");
    if (readSerialFrame(respond) > 0)
    {
        if (strcmp(respond, "\r\nOK\r\n") != 0)
        {
            goto try_on;
        }
    }
}
```

```
    }  
    }  
    else  
    {  
        goto try_on;  
    }  
  
    modemSerial.write("AT+CGNSPWR=1\r");  
    if (readSerialFrame(respond) > 0)  
    {  
        if (strcmp(respond, "\r\nOK\r\n") != 0)  
        {  
            goto try_on;  
        }  
    }  
    else  
    {  
        goto try_on;  
    }  
  
    modemSerial.write("AT+CGNSSEQ=\"RMC\"\r");  
    if (readSerialFrame(respond) > 0)  
    {  
        if (strcmp(respond, "\r\nOK\r\n") != 0)  
        {  
            goto try_on;  
        }  
    }  
    else
```

```
{
    goto try_on;
}

modemSerial.write("AT+CMGF=1\r");
if (readSerialFrame(respond) > 0)
{
    if (strcmp(respond, "\r\nOK\r\n") != 0)
    {
        goto try_on;
    }
}
else
{
    goto try_on;
}

modemSerial.write("AT+CNMI=2,2,0,0,0\r");
if (readSerialFrame(respond) > 0)
{
    if (strcmp(respond, "\r\nOK\r\n") != 0)
    {
        goto try_on;
    }
}
else
{
    goto try_on;
}
```

```
modemSerial.write("AT+CMGDA=DEL ALL\r");
if (readSerialFrame(respond) > 0)
{
    if (strcmp(respond, "\r\nOK\r\n") != 0)
    {
        goto try_on;
    }
}
else
{
    goto try_on;
}

is_modem_init_ok = true;

Serial.write("Initted");
delay(2000);
kt_ok = 1;
}
```

```
int readSerialFrame(char result[]) {
    int i = 0;
    unsigned long currentMillis, previousMillis;
    previousMillis = millis();
    currentMillis = previousMillis;
    while (1) {
        if (currentMillis - previousMillis >= 100)
```

```
{
    return i;
}
else
{
    if (modemSerial.available() > 0) {
        previousMillis = currentMillis;
        char inChar = modemSerial.read();
        result[i] = inChar;
        i++;
        result[i] = 0;
    }
}
currentMillis = millis();
}
}
```

```
static int GetMessageFeilds(unsigned char **message_feilds, unsigned char *msg,
unsigned char separate_char, int max_feild_get)
```

```
{
    int count_feild = 0;
    *message_feilds = msg;
    message_feilds++;
    count_feild++;
    while (*msg != '\0')
    {
        if (*msg == separate_char)
        {
```

```
*msg = '\0';
*message_feilds = msg + 1;
message_feilds++;
count_feild++;
if (count_feild >= max_feild_get)
    return count_feild;
}
msg++;
}
return count_feild;
}

static void NMEA_GPRMC_Decoder(GPS_T *gps_data, char *data_bytes)
{
    //Get data
    char *message_field[13];
    if (GetMessageFeilds(message_field, data_bytes, ',', 13) == 13)
    {
        Serial.write(message_field[GPS_FSTS]);
        Serial.write(message_field[GPS_SPEED]);
        Serial.write(message_field[GPS_ORG]);
        Serial.write(message_field[GPS_LON]);
        Serial.write(message_field[GPS_TIME]);

        if (strcmp(message_field[GPS_FSTS], "1") == 0)
        {
            gps_data->State_gps = true;
        }
    }
}
```

```
gps_data->State_gps = true;

gps_data->SpeedGPS = atof(message_field[GPS_SPEED]);
gps_data->OrGPS = atof(message_field[GPS_ORG]);
gps_data->Lat = atof(message_field[GPS_LAT]);
gps_data->Lng = atof(message_field[GPS_LON]);
}
else
{
gps_data->State_gps = false;

gps_data->Lat = 0;
gps_data->Lng = 0;
gps_data->SpeedGPS = 0;
}

gps_data->time_gps.time.sec = message_field[GPS_TIME][13] - '0' + 10 *
(message_field[GPS_TIME][12] - '0');
gps_data->time_gps.time.minute = message_field[GPS_TIME][11] - '0' + 10 *
(message_field[GPS_TIME][10] - '0');
gps_data->time_gps.time.hour = message_field[GPS_TIME][9] - '0' + 10 *
(message_field[GPS_TIME][8] - '0');

gps_data->time_gps.time.date = message_field[GPS_TIME][7] - '0' + 10 *
(message_field[GPS_TIME][6] - '0');
gps_data->time_gps.time.month = message_field[GPS_TIME][5] - '0' + 10 *
(message_field[GPS_TIME][4] - '0');
```



```
gps_data->time_gps.time.year = message_field[GPS_TIME][3] - '0' + 10 *  
(message_field[GPS_TIME][2] - '0') + 100 * (message_field[GPS_TIME][1] - '0') + 1000  
* (message_field[GPS_TIME][0] - '0');  
  
}  
}
```