

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH**



**ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ
TIỀM NGỪA CHÓ MÈO**

**GVHD: NGUYỄN VĂN HIỆP
SVTH: TRẦN NGỌC HÙNG
MSSV: 15141174
SVTH: ĐẶNG MINH SANG
MSSV: 15141262**



Tp. Hồ Chí Minh, tháng 07/2020

BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP - Y SINH



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG
ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG QUẢN LÝ TIÊM NGỪA CHÓ MÈO

GVHD: ThS Nguyễn Văn Hiệp

SVTH:

MSSV:

Trần Ngọc Hùng

15141174

Đặng Minh Sang

15141262

TP Hồ Chí Minh - 07/2020

**BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP - Y SINH**

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG
ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG QUẢN LÝ TIÊM NGỪA CHÓ MÈO

GVHD: ThS Nguyễn Văn Hiệp

SVTH:

MSSV:

Trần Ngọc Hùng

15141174

Đặng Minh Sang

15141262

TP Hồ Chí Minh - 07/2020

Tp. HCM, ngày 20 tháng 07 năm 2020

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Trần Ngọc Hùng MSSV: 15141174
Đặng Minh Sang MSSV: 15141262
Chuyên ngành: Sư phạm kỹ thuật điện tử- truyền thông
Mã ngành: 510302
Hệ đào tạo: Đại học chính quy Mã hệ: D
Khóa: 2015 Lớp: 15941DT

I. TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ TIÊM NGỪA CHÓ MÈO

II. NHIỆM VỤ

1. Các số liệu ban đầu:

- Tình hình tiêm ngừa chó mèo
- Số người tử vong vì bệnh dại.
- Vi điều khiển: Arduino uno R3, ESP8266.
- Các loại module: RFID RC522, Bluetooth HC05.
- Màn hình hiển thị: LCD 1602
- Nguồn: Adapter, mạch sạc và tăng áp, Pin.

2. Nội dung thực hiện:

- Tìm hiểu nguyên lý hoạt động của module ESP8266
- Tìm hiểu các chuẩn truyền thông như USART, SPI.
- Thiết kế và thi công hộp đựng sản phẩm.
- Thiết kế phần mềm thông báo trên điện thoại Android.
- Viết chương trình điều khiển cho Arduino và ESP8266, nạp code và chạy thử nghiệm sản phẩm.
- Chỉnh sửa và hoàn thiện hệ thống.
- Thực hiện viết luận văn báo cáo.
- Tiến hành báo cáo đề tài tốt nghiệp.

III. NGÀY GIAO NHIỆM VỤ:

IV. NGÀY HOÀN THÀNH NHIỆM VỤ:

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: ThS Nguyễn Văn Hiệp

CÁN BỘ HƯỚNG DẪN

BM. ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

Tp. HCM, ngày 20 tháng 07 năm 2020

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên 1: Trần Ngọc Hùng

Lớp: 151494DT

MSSV: 15141174

Họ tên sinh viên 2: Đặng Minh Sang

Lớp: 151494DT

MSSV: 1514126

Tên đề tài: **XÂY DỰNG HỆ THỐNG QUẢN LÝ TIÊM NGỪA CHÓ MÈO**

<i>Tuần/ngày</i>	<i>Nội dung</i>	<i>Xác nhận GVHD</i>
Tuần 1 (09/03/2020)	_ Báo cáo GVHD _ Tìm đề tài	
Tuần 2 (16/03/2020)	_ Báo cáo GVHD _ Tìm hiểu về đề tài, các công thức tính toán, các phương pháp thực hiện	
Tuần 3 (23/03/2020)	_ Báo cáo GVHD _ Tổng hợp các linh kiện cần dùng cho đề tài	
Tuần 4 (30/03/2020)	_ Báo cáo GVHD _ Tìm hiểu hoạt động NodeMCU, Arduino, các module, LCD	
Tuần 5 (06/04/2020)	_ Báo cáo GVHD _ Tìm hiểu về google sheet, Android studio	
Tuần 6 (13/04/2020)	_ Báo cáo GVHD _ Tìm hiểu về google sheet, Android studio	
Tuần 7 (20/04/2020)	_ Báo cáo GVHD _ Kết nối, kiểm tra hoạt động của các module với NodeMCU	
Tuần 8 (27/04/2020)	_ Báo cáo GVHD _ Kết nối, kiểm tra hoạt động của các module, LCD với Arduino uno r3	
Tuần 9 (04/05/2020)	_ Báo cáo GVHD _ Viết chương trình cho NodeMCU và Arduino	

Tuần 10 (11/05/2020)	_ Báo cáo GVHD _ Viết chương trình nhận và gửi dữ liệu trên google sheet	
Tuần 11 (18/05/2020)	_ Báo cáo GVHD _ Viết chương trình cho ứng dụng điều khiển trên điện thoại	
Tuần 12 (25/05/2020)	_ Báo cáo GVHD _ Thi công mạch điều khiển và chỉnh sửa	
Tuần 13 (01/06/2020)	_ Báo cáo GVHD _ Thiết kế thi công vỏ hộp, đóng gói mô hình	
Tuần 14 (08/06/2020)	_ Báo cáo GVHD _ Viết báo cáo và chỉnh sửa	
Tuần 15 (15/06/2020)	_ Báo cáo GVHD _ Chỉnh sửa, in luận văn	

GV HƯỚNG DẪN

(Ký và ghi rõ họ và tên)

LỜI CAM ĐOAN

Đề tài này là do nhóm tự thực hiện dưới sự hướng dẫn của giảng viên ThS Nguyễn Văn Hiệp. Đề tài dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Nhóm thực hiện đề tài

Trần Ngọc Hùng

Đặng Minh Sang

LỜI CẢM ƠN

Nhóm em xin gửi lời cảm ơn sâu sắc đến Thầy Nguyễn Văn Hiệp, người đã trực tiếp hướng dẫn và giúp đỡ tạo điều kiện để nhóm hoàn thành tốt đề tài này. Mặc dù kiến thức của nhóm còn hạn chế nhưng thầy đã nhiệt tình giúp đỡ, vạch ra hướng đi sao cho phù hợp.

Xin chân thành cảm ơn quý thầy cô giảng viên khoa Điện - Điện Tử, trường Đại Học Sư Phạm Kỹ Thuật Tp.HCM đã giúp đỡ nhóm trong quá trình thực hiện đồ án.

Ngoài ra nhóm còn nhận sự quan tâm, giúp đỡ của gia đình, bạn bè, các anh chị trong trường để hoàn thành đồ án này. Do kiến thức còn hạn chế nên trong quá trình thực hiện không tránh khỏi sai sót mong thầy cô và các bạn góp ý để nhóm có thể hoàn thành đồ án tốt nhất.

Xin chân thành cảm ơn!

Nhóm thực hiện đề tài

Trần Ngọc Hùng Đặng Minh Sang

MỤC LỤC

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	i
LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP	ii
LỜI CAM ĐOAN	iv
LỜI CẢM ƠN	v
MỤC LỤC	vi
LIỆT KÊ HÌNH	ix
LIỆT KÊ BẢNG	xii
TÓM TẮT	xiii
CHƯƠNG 1: TỔNG QUAN	1
1.1 ĐẶT VẤN ĐỀ	1
1.2 MỤC TIÊU	2
1.3 NỘI DUNG NGHIÊN CỨU	2
1.4 GIỚI HẠN	2
1.5 BỐ CỤC	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	4
2.1 TỔNG QUAN VỀ HỆ THỐNG QUẢN LÝ TIÊM NGỪA CHÓ MÈO	4
2.1.1 Giới thiệu hệ thống quản lý tiêm ngừa chó mèo	4
2.1.2 Ứng dụng của hệ thống quản lý tiêm phòng chó mèo	4
2.2 GIỚI THIỆU VỀ HỆ THỐNG RFID	5
2.2.1 Giới thiệu Công Nghệ RFID	5
2.2.2 Ưu và nhược điểm của hệ thống RFID	7
2.3 TỔNG QUAN VỀ HỆ ĐIỀU HÀNH ANDROID	8
2.3.1 Khái niệm hệ điều hành android.....	8
2.3.2 Giao diện.....	8
2.3.3 Ứng dụng	10
2.3.4 Ưu và nhược điểm của hệ điều hành Android.....	11
2.4 TỔNG QUAN VỀ TRANG TÍNH GOOGLE SHEET	12
2.4.1 khái niệm về trang tính google sheet	12
2.4.2 Ưu điểm của trang tính google sheet	12
2.5 ỨNG DỤNG GOOGLE SHEET CHO ỨNG DỤNG IOTS	13
2.5.1 Các phương thức sử dụng để truyền tải dữ liệu	13
2.5.2 Cách thức truyền dữ liệu giữa google sheets và các thiết bị IoTs	15

2.6 CÁC CHUẨN TRUYỀN DỮ LIỆU	17
2.6.1 Chẩn giao tiếp I2C.....	17
2.6.2 Chuẩn truyền thông UART	19
2.6.3 Chuẩn truyền thông SPI	20
CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ	22
3.1 GIỚI THIỆU	22
3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG	22
3.2.1 Thiết kế sơ đồ khối hệ thống.....	22
3.3 TÍNH TOÁN VÀ THIẾT KẾ MẠCH	24
3.3.1 Khối module RFID	24
3.3.2. Khối module wifi.....	26
3.3.3 Vi điều khiển	27
3.3.5 Module I2C	32
3.3.6 Module Bluetooth	32
3.3.7 Khối nguồn	33
3.4 SƠ ĐỒ NGUYÊN LÝ TOÀN MẠCH.....	34
CHƯƠNG 4: THI CÔNG HỆ THỐNG	36
4.1 GIỚI THIỆU	36
4.2 THI CÔNG HỆ THỐNG	36
4.2.1 Thi công Board mạch.....	36
4.2.2 Lắp ráp và kiểm tra.....	38
4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH.....	39
4.3.1 Đóng gói board dành cho trung tâm y tế.....	39
4.3.2 Đóng gói board dành cho đội bắt chó mèo thả rông	41
4.4 LẬP TRÌNH HỆ THỐNG	42
4.4.1 Lưu đồ giải thuật	42
4.4.2 Phần mềm lập trình cho vi điều khiển.....	44
4.4.3 Phần mềm lập trình cho điện thoại, máy tính	50
4.4.4 Hướng dẫn xây dựng code đối với google sheet.....	53
4.5 VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC.....	55
CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ.....	58
5.1 GIỚI THIỆU	58
5.2 KẾT QUẢ ĐẠT ĐƯỢC	58
5.3 KẾT QUẢ THỰC NGHIỆM.....	59

5.3.1 Thiết kế thi công phần cứng	59
5.4 NHẬN XÉT, ĐÁNH GIÁ.....	62
5.4.1 Nhận xét.....	62
5.4.2 Đánh giá.....	63
6.1 KẾT LUẬN	64
6.2 HƯỚNG PHÁT TRIỂN.....	64
TÀI LIỆU THAM KHẢO.....	65
PHỤ LỤC	66

LIỆT KÊ HÌNH

Hình 2.2. Giao tiếp giữa thẻ Tag và đầu đọc.....	6
Hình 2.3. Logo hệ điều hành Android.....	8
Hình 2.4. Giao diện của thiết bị chạy hệ điều hành Android.....	9
Hình 2.5. Thanh trạng thái của thiết bị chạy hệ điều hành Android.....	9
Hình 2.6. Kho ứng dụng Google Play Store.....	10
Hình 2.7. Ứng dụng trang tính google sheet.....	12
Hình 2.8. Chuỗi dữ liệu ghi dưới dạng JSON.....	16
Hình 2.9. Lập trình code cho Google Sheets.....	16
Hình 2.10. Khai báo thư viện và các lệnh cài đặt để truyền dữ liệu.....	17
Hình 2.11. Xử lý các chuỗi JSON.....	18
Hình 2.12. Bus I2C và các thiết bị ngoại vi.....	18
Hình 2.13. Trình tự truyền bit trên đường truyền.....	19
Hình 2.14. Truyền thông UART.....	20
Hình 2.15. Giao tiếp SPI.....	22
Hình 3.1. Sơ đồ khối hệ thống đặt ở trạm y tế.....	23
Hình 3.2. Sơ đồ khối hệ thống dành cho đội bắt chó mèo thả rong.....	23
Hình 3.3. Module RFID RC522.....	24
Hình 3.4. Sơ đồ kết nối module RFID RC522 với module Wifi.....	25
Hình 3.5. Module Wifi NodeMCU ESP8266.....	26
Hình 3.6. Board arduino uno R3.....	27
Hình 3.7. Sơ đồ và kí hiệu chân trên arduino uno R3.....	29
Hình 3.8. Hình ảnh mặt trước của LCD 1602.....	30
Hình 3.9. Module chuyên đổi I2C cho LCD 1602.....	32
Hình 3.10. Module Bluetooth HC05.....	32
Hình 3.11. Pin Cell 18650 4200mAh 3.7V.....	31
Hình 3.12. Mạch sạc bảo vệ pin.....	31
Hình 3.13. Sơ đồ nguyên lý mạch trong trung tâm y tế.....	34
Hình 3.14. Sơ đồ nguyên lý toàn mạch của đội bắt chó mèo thả rong.....	35
Hình 4.1. PCB của board tại trung tâm y tế.....	36
Hình 4.2. PCB của board nhân viên đô thị.....	37
Hình 4.3. Hình mạch sau khi ủi rửa của board tại trung tâm y tế.....	37

Hình 4.4. Hình mạch sau khi ủi rửa của board nhân viên đô thị.....	38
Hình 4.5. Mặt trên board của trạm y tế khi lắp linh kiện	38
Hình 4.6. Mặt trên board của nhân viên đô thị khi lắp linh kiện.....	39
Hình 4.7. Hình ảnh mặt trước khi đóng hộp mica.....	40
Hình 4.8. Hình ảnh mặt sau khi đóng hộp mica	40
Hình 4.9. Hình ảnh mặt sau khi đóng hộp mica	41
Hình 4.10. Hình ảnh mặt trước khi đóng hộp mica.....	41
Hình 4.11. Lưu đồ chương trình board sử dụng trong trạm y tế	42
Hình 4.12. Lưu đồ chương trình sử dụng cho đội bắt chó mèo thả rông	43
Hình 4.13 Giao diện tải Arduino IDE	45
Hình 4.14 Ủng hộ nhà phát triển Arduino IDE	45
Hình 4.15 Giao diện chính của Arduino IDE.....	46
Hình 4.16 Cài đặt Driver cho NodeMCU 1.....	47
Hình 4.17 Cài đặt Driver cho NodeMCU 2.....	47
Hình 4.18 Cài đặt Driver cho NodeMCU3.....	48
Hình 4.19 Cài đặt Driver cho NodeMCU4.....	48
Hình 4.20 Chọn phần cứng để lập trình	49
Hình 4.21 Chọn Port kết nối.....	49
Hình 4.22 Cài đặt thư viện cho Arduino IDE.....	50
Hình 4.23 Giao diện phần mềm Android Studio.....	50
Hình 4.24 Giao diện phần thiết kế giao diện cho ứng dụng	51
Hình 4.25 Ánh xạ các đối tượng trong giao diện với các câu lệnh	52
Hình 4.26 Giao diện phân lập trình Java	52
Hình 4.27 Giao diện trang google sheet	53
Hình 4.28 Các bước mở trang code lập trình	53
Hình 4.29 Giao diện trang code lập trình	54
Hình 4.30 Các bước cho phép google script liên kết	54
Hình 4.31 giao diện app sử dụng.....	55
Hình 4.32 Giao diện hiển thị của app dành cho nhân viên.....	56
Hình 4.33 Giao diện hiển thị thông tin	57
Hình 5.1. Hình ảnh thực tế mặt trước sau khi hoàn thiện.....	59
Hình 5.2. Giao diện chính.....	60
Hình 5.3 Giao diện của mục thêm dữ liệu.....	61

Hình 5.4 Giao diện của danh sách dữ liệu và tìm kiếm.....	61
Hình 5.5 Giao diện của hệ thống đội bắt chó mèo thả rông	62

LIỆT KÊ BẢNG

Bảng 3.1. Thông số Ardiuno Uno R3.....	29
Bảng 3.2 Thông số các chân của LCD 1602	30
Bảng 3.3. Thông số, giá trị các linh kiện sử dụng.....	31
Bảng 5.1. Số liệu thực nghiệm	63

TÓM TẮT

Đề tài “ Xây dựng hệ thống quản lý tiêm ngừa chó mèo” sẽ là một công cụ hỗ trợ đắc lực nhất trong công tác quản lý dữ liệu, bên cạnh đó trong lĩnh vực y tế thú y cũng được áp dụng trong việc quản lý các thông tin liên quan đến các bệnh nhân ở đây là các con thú cưng như: các thông tin cá nhân, hồ sơ bệnh án, lịch sử khám bệnh. Nhưng sau khi nhóm đã đi tìm hiểu tại các bệnh viện thú y hiện nay thì đa số việc áp dụng còn rất nhiều hạn chế vì vậy nhóm đã làm đề tài này với mong muốn để giải quyết vấn đề về quản lý dữ liệu bệnh nhân để giúp giảm bớt thời gian cũng như chi phí cho công tác lưu trữ dữ liệu của các con thú cưng. Vừa giúp mọi người có thể phòng tránh được bệnh đại từ các con thú cưng và kể cả các con chó mèo hoang, vừa đảm bảo được sức khỏe của chúng.

Nội dung chính của đề tài quản lý dữ liệu bệnh nhân trong bệnh viện sử dụng công nghệ

RFID, bao gồm:

- Sử dụng board Arduino UNO làm vi điều khiển của khối điều khiển trung tâm.
- Ứng dụng công nghệ RFID trong việc quản lý chó mèo.
- Thêm, tra cứu thông tin chó mèo qua phần mềm máy tính.
- Quản lý và lưu trữ thông tin bệnh nhân qua database.

CHƯƠNG 1: TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Để chủ động ngăn ngừa bệnh Đại lầy truyền từ động vật sang người, hạn chế tối đa số trường hợp tử vong do bệnh Đại. Theo đó, việc yêu cầu tổ chức thực hiện phải có sự chỉ đạo và phối hợp chặt chẽ, đồng bộ giữa các cấp, các ngành và thực hiện theo hướng dẫn của Chi cục Chăn nuôi - Thú y, các chủ vật thực hiện kế hoạch hành động tiêm phòng vắc xin Đại cho chó, mèo; tiêm phòng đúng chủng loại vắc xin, đúng đối tượng, an toàn cho người, động vật và không ảnh hưởng đến môi trường; tiêm phòng triệt để số lượng chó, mèo chưa được tiêm phòng Đại.

Hiện nay việc tiêm phòng cho chó mèo chủ yếu là dựa vào chủ vật nuôi tự động mang đến các trung tâm thú y để tiêm, và họ cũng không thể nhớ chính xác được thời gian tiêm, loại vacxin... Điều đó cũng dẫn tới rất khó kiểm soát được số lần tiêm, thời gian, thuốc đã sử dụng... Phía bên trung tâm y tế cũng không thể liên hệ thông báo cho từng chủ của vật nuôi với những cách thông thường được.

Nhận thấy hiện tại việc kiểm soát tiêm phòng cho chó mèo còn chưa được ổn định nên nhóm chúng em quyết định lựa chọn đề tài “*Xây dựng hệ thống quản lý tiêm ngừa chó mèo*” sử dụng công nghệ RFID và đồng bộ dữ liệu thông qua Google Sheet có thể giúp việc quản lý tiêm phòng chó mèo được chặt chẽ và chính xác hơn. Hệ thống có thể tự động thông báo cho người chủ vật nuôi biết được các thông tin tiêm phòng cho chủ vật nuôi một cách chính xác tránh tình trạng chậm trễ.

1.2 MỤC TIÊU

Thiết kế và thi công hệ thống giám sát và tiêm ngừa chó mèo. Được theo dõi trực tuyến trên laptop thông qua Google sheet và trên điện thoại Android thông qua giao diện được lập trình.

1.3 NỘI DUNG NGHIÊN CỨU

- **Nội dung 1:** Nghiên cứu mạch giao tiếp, đọc ghi dữ liệu lên thẻ RFID
- **Nội dung 2:** Nghiên cứu xây dựng việc cập nhật dữ liệu lên mạng, quản lý thông tin của bác sĩ thú y, của người dùng.
- **Nội dung 3:** Xây dựng bảng thông tin trên google sheet giao tiếp với hệ thống thông qua Wifi và dữ liệu di động.
- **Nội dung 4:** Xây dựng một ứng dụng Android giao tiếp với hệ thống thông qua Wifi và dữ liệu di động.
- **Nội dung 5:** Thiết kế mô hình hệ thống.
- **Nội dung 6:** Thi công phần cứng, chạy thử nghiệm và hiệu chỉnh hệ thống.
- **Nội dung 7:** Viết báo cáo thực hiện.
- **Nội dung 8:** Bảo vệ luận văn.

1.4 GIỚI HẠN

- Ứng dụng điện thoại chạy trên hệ điều hành Android.
- Giao tiếp với hệ thống thông qua Wifi và dữ liệu di động.
- Google sheet hiển thị các thông tin về những lần tiêm phòng trước đó.
- Chỉ áp dụng trên chó, mèo và với việc tiêm phòng bệnh dại.

1.5 BỐ CỤC

- **Chương 1:** Tổng quan
Chương này trình bày đặt vấn đề, 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
Chương này trình bày lý thuyết về các module, cảm biến và linh kiện sử dụng trong hệ thống, các chuẩn truyền thông, giao thức.
- **Chương 3:** Thiết kế và tính toán
Chương này thiết kế sơ đồ khối, sơ đồ nguyên lí của các khối trong hệ thống và thực hiện tính toán thiết kế.

CHƯƠNG 1: TỔNG QUAN

- **Chương 4:** Thi công hệ thống

Chương này trình bày lưu đồ giải thuật, thiết kế app android, viết chương trình hệ thống, thiết kế sơ đồ mạch in PCB.

- **Chương 5:** Kết quả, nhận xét và đánh giá

Chương này trình bày kết quả thi công phần cứng và kết quả hình ảnh thực tế của tủ, nhận xét đánh giá chung về sản phẩm.

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

Trong chương này sẽ đưa ra kết quả đạt được, phân tích những ưu nhược điểm và đề xuất hướng phát triển đề tài.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 TỔNG QUAN VỀ HỆ THỐNG QUẢN LÝ TIÊM NGỪA CHÓ MÈO

2.1.1 Giới thiệu hệ thống quản lý tiêm ngừa chó mèo

Tiêm chủng cho chó mèo là biện pháp giúp bạn bảo vệ sức khỏe cho vật nuôi của mình, tránh các tác nhân gây bệnh cho vật nuôi và bảo vệ sức khỏe cho con người trước khả năng lây bệnh từ vật nuôi sang người.

Tiêm chủng cho vật nuôi, đặc biệt là chó mèo là việc làm cần thiết và bắt buộc, trước hết là phòng đại, tiếp là bảo vệ sức khỏe cho cả con người và vật nuôi đó.

Có rất nhiều nguyên nhân gây bệnh cho vật nuôi, khi có dấu hiệu bệnh tật ở vật nuôi, bản thân người chủ rất khó phát hiện vì con vật không có khả năng biểu hiện ra. Do đó, việc tiêm chủng cho mèo, chó là tăng sức đề kháng và hệ thống miễn dịch với các loại bệnh truyền nhiễm, bệnh nguy hiểm của thú cưng, nâng cao khả năng chống chọi lại bệnh tật.

Nhận định tình hình thực tế như trên, cùng với các kiến thức đã được trang bị, nhóm bọn em kiến nghị thực hiện đề tài “*Xây dựng hệ thống quản lý tiêm ngừa chó mèo*” để giải quyết vấn đề tiêm ngừa cho chó, mèo, cũng như bảo vệ an toàn sức khỏe cho con người.

2.1.2 Ứng dụng của hệ thống quản lý tiêm phòng chó mèo

- Hệ thống có chức năng thêm, xóa, chỉnh sửa các thông tin của chó, mèo thông qua trang dữ liệu Google sheet.
- Lưu trữ thông tin chó, mèo với số lượng lớn
- Đáp ứng và dễ dàng tìm kiếm tất cả các thông tin cơ bản của chó, mèo.
- Cho phép người dùng tìm kiếm thông tin vật nuôi với mã thẻ đọc được của mỗi vật nuôi. Quản lý các thông tin cơ bản của chó, mèo như: họ và tên, địa chỉ của chủ chó, mèo, số điện thoại, ngày vào, ngày ra, và hồ sơ khám bệnh.
- Quản lý và xuất hồ sơ của vật nuôi theo mã thẻ, ngày vào hoặc cho từng đơn thuốc khác nhau của chó, mèo.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.2 GIỚI THIỆU VỀ HỆ THỐNG RFID

2.2.1 Giới thiệu Công Nghệ RFID

Công nghệ RFID (Radio Frequency Identification) là công nghệ nhận dạng đối tượng bằng sóng vô tuyến, cho phép một thiết bị đọc có thể đọc thông tin chứa trong một thiết bị khác ở một khoảng cách gần mà không cần phải có một sự tiếp xúc vật lý nào. Một hệ thống RFID thường bao gồm 2 phần chính là thẻ tag (chip RFID chứa thông tin) và bộ đọc (reader) đọc các thông tin trên chip.

Kỹ thuật RFID sử dụng truyền thông không dây trong tải tần sóng vô tuyến để truyền dữ liệu từ các thẻ tag đến bộ đọc. Bộ đọc dữ liệu của tag và gửi thông tin để hệ thống để xử lý trên cơ sở dữ liệu.

Dạng đơn giản và phổ biến nhất được sử dụng hiện nay là hệ thống RFID bị động. Trong đó bộ đọc truyền một tín hiệu tần số vô tuyến thông qua anten đến một con chip, sau đó bộ đọc sẽ nhận lại thông tin phản hồi từ chip và gửi đến máy tính để xử lý thông tin. Các con chip từ các thẻ tag này không cần nguồn nuôi, chúng sử dụng năng lượng phát ra từ tín hiệu được gửi bởi bộ đọc.

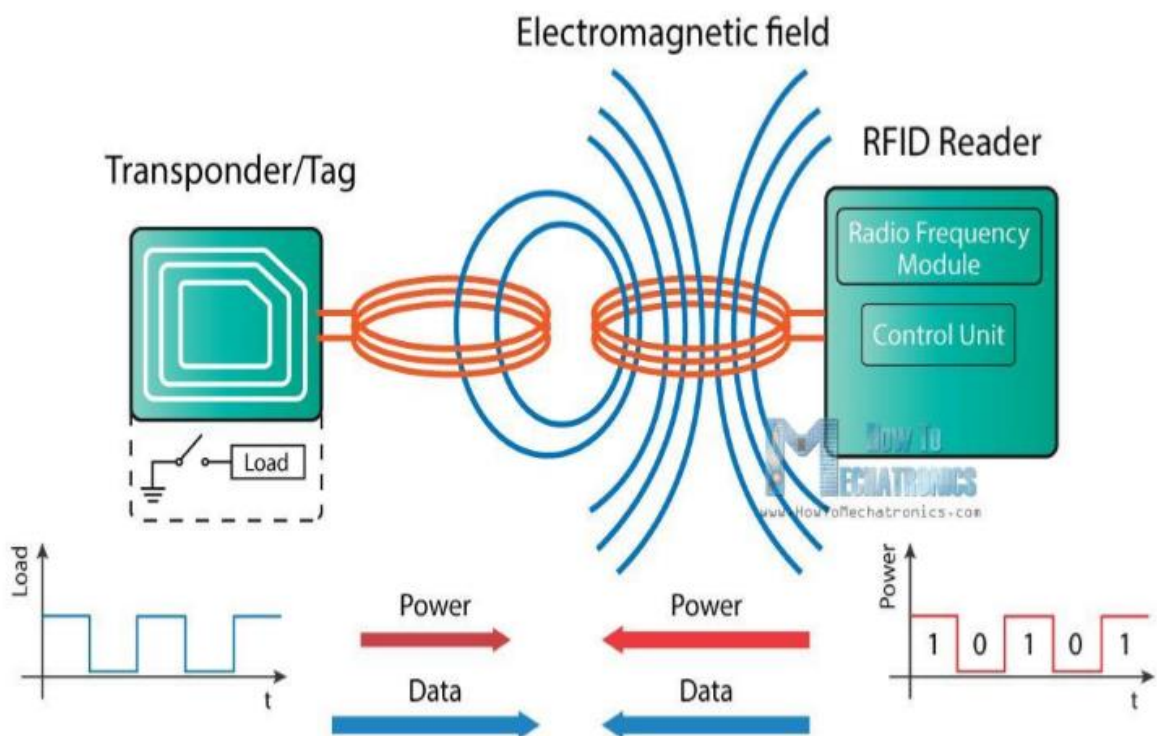
Thẻ RFID là thiết bị có thể lưu trữ và truyền dữ liệu về bộ đọc bằng sóng vô tuyến. Trong đó các thẻ thường lưu trữ thông tin về các sản phẩm nào đó. Dữ liệu có thể là một số nhận dạng đơn giản được lưu trữ trong một thẻ chỉ đọc hoặc dữ liệu phức tạp hơn. Các thẻ phức tạp hơn này có thể chứa được các dữ liệu về ngày sản xuất số serial, hoặc thậm chí một số loại đặc biệt còn chứa các cảm biến để theo dõi nhiệt độ trung bình hoặc các loại dữ liệu khác. Thẻ RFID gồm chip bán dẫn nhỏ (bộ nhớ của chip có thể chứa từ 96 đến 512 bit dữ liệu, nhiều gấp 64 lần so với mã vạch) và anten được thu nhỏ trong một số hình thức đóng gói. Vài thẻ RFID giống như những nhãn giấy và được ứng dụng để bỏ vào hộp và đóng gói. Một số khác được sản xuất thành các miếng da bao cổ tay. Mỗi thẻ được lập trình với một nhận dạng duy nhất cho phép theo dõi không dây đối tượng hoặc con người đang gắn thẻ đó. Khi thẻ đi vào vùng sóng điện từ, nó sẽ phát hiện ra tín hiệu kích hoạt từ đầu đọc và sẽ phát thông tin nhận

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

dạng đến đầu đọc. Đầu đọc giải mã dữ liệu được mã hóa trong chip (sóng vô tuyến phản xạ từ thẻ) và gửi vào hệ thống để xử lý.



Hình 2.1. Một số thẻ RFID thông dụng hiện nay



Hình 2.2. Giao tiếp giữa thẻ Tag và đầu đọc

Một hệ thống RFID bao gồm các thành phần sau:

- Reader: là thành phần bắt buộc, thường được tích hợp sẵn cả anten.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Thẻ tag: là thành phần bắt buộc với mọi hệ thống RFID.
- Thiết bị xử lý: bao gồm các vi xử lý có khả năng nhận được mã tag được gửi về từ reader, sau đó gửi lên hệ thống.
- Ngoài ra các hệ thống lớn còn được kết nối với các máy tính, hạ tầng mạng để truyền nhận thông tin của thẻ tag, thực hiện các tác vụ như liên kết tài khoản, thông tin, tiền phí... Cũng như có các cơ cấu chấp hành để thực thi các yêu cầu đặt ra với hệ thống.

2.2.2 Ưu và nhược điểm của hệ thống RFID

➤ Ưu điểm

- Đọc với tốc độ cao mà không cần tiếp xúc vật lý: nhiều đối tượng có thể được quét tại cùng một thời điểm, có thể lên đến 40 thẻ trong 3 giây làm giảm thời gian hoạt động tăng năng suất của hệ thống.
- Khả năng đọc và ghi dữ liệu nhiều lần: một số thẻ cho phép đọc và ghi dữ liệu nhiều lần, từ đó làm giảm chi phí hoạt động của hệ thống, cũng như của người sử dụng.
- Nhỏ gọn, bền: các thẻ RFID hoạt động khá tốt trong môi trường không thuận lợi (nóng ẩm, bụi bẩn...).
- Một số thẻ RFID, đặc biệt là các thẻ thụ động không cần phải cung cấp nguồn để có thể hoạt động, từ đó nâng cao tính tiện lợi của hệ thống.
- Việc áp dụng công nghệ RFID vào các lĩnh vực của đời sống làm tăng năng suất lao động, đồng thời tự động hoá các quy trình, sản xuất, thay thế các hoạt động đòi hỏi việc phải lặp đi lặp lại với tần suất cao của con người, từ đó giảm thiểu, triệt tiêu những sai sót có thể xảy ra.
- Kiểm kê với tốc độ cao mà không cần tiếp xúc: Nhiều đối tượng có thể được quét tại cùng một thời điểm, có thể lên đến 40 thẻ trong 1 giây. Kết quả là thời gian để đếm các đối tượng đã giảm thực sự.

➤ Nhược điểm

- Khả năng kiểm soát các thiết bị còn hạn chế: thẻ dễ bị nhiễu sóng trong môi trường nước và kim loại.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Các đầu đọc có thể đọc chồng lên nhau: vì nhiệm vụ của các đầu đọc thẻ là gửi tín hiệu đến các thẻ tag, sau đó nhận tín hiệu gửi về, vì thế trong một số trường hợp có thể xảy ra việc đọc chồng chéo lên nhau.
- Giá thành của hệ thống RFID hiện nay vẫn còn khá cao, xét đến tính thực tế ở Việt Nam thì vẫn chưa thể ứng dụng rộng rãi trong mọi lĩnh vực.

2.3 TỔNG QUAN VỀ HỆ ĐIỀU HÀNH ANDROID

2.3.1 Khái niệm hệ điều hành android

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Android, Inc. Với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào năm 2008.

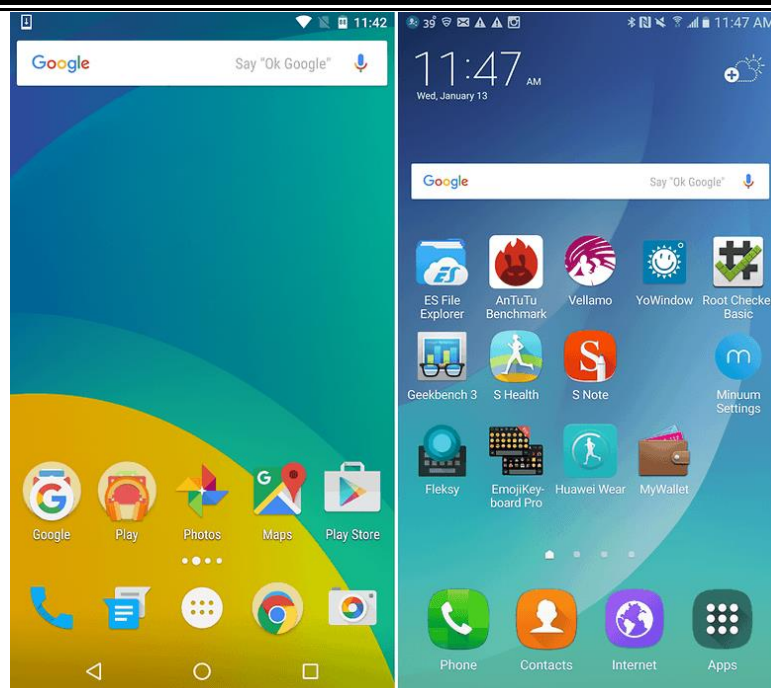


Hình 2.3. Logo hệ điều hành Android

2.3.2 Giao diện

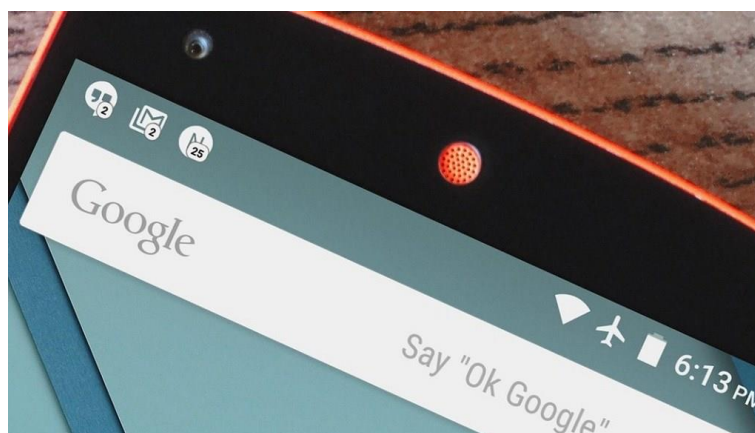
Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.4. Giao diện của thiết bị chạy hệ điều hành Android

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng (icon) và tiện ích (widget); biểu tượng ứng dụng sẽ mở ứng dụng tương ứng, còn tiện ích hiển thị những nội dung sống động, cập nhật tự động như dự báo thời tiết, hộp thư của người dùng, hoặc những mẩu tin thời sự ngay trên màn hình chính. Màn hình chính có thể gồm nhiều trang xem được bằng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích.



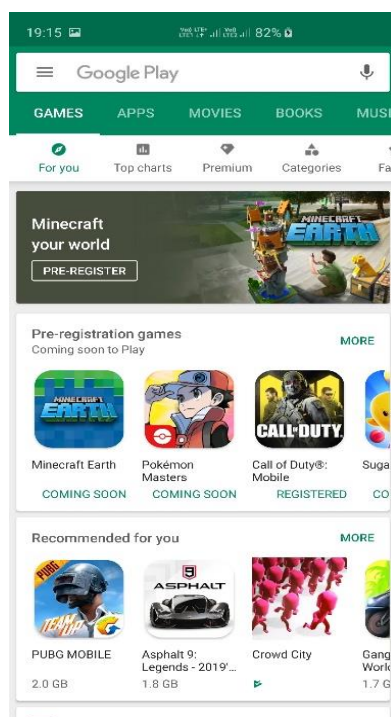
Hình 2.5. Thanh trạng thái của thiết bị chạy hệ điều hành Android

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể "kéo" xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện. Trong các phiên bản đời đầu, người dùng có thể nhấn vào thông báo để mở ra ứng dụng tương ứng, về sau này các thông tin cập nhật được bổ sung thêm tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi nhỡ mà không cần phải mở ứng dụng gọi điện ra. Thông báo sẽ luôn nằm đó cho đến khi người dùng đã đọc hoặc xóa nó đi.

2.3.3 Ứng dụng

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng Bộ phát triển phần mềm Android (SDK). SDK bao gồm một bộ đầy đủ các công cụ dùng để phát triển gồm có công cụ gỡ lỗi, thư viện phần mềm, bộ giả lập điện thoại dựa trên tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp (IDE) được hỗ trợ chính thức là Eclipse sử dụng phần bổ sung Android Development Tools (ADT). Các công cụ phát triển khác cũng có sẵn, gồm có Bộ phát triển gốc dành cho các ứng dụng hoặc phần mở rộng viết bằng C hoặc C++, Google App Inventor, một môi trường đồ họa cho những nhà lập trình mới bắt đầu, và nhiều nền tảng ứng dụng web di động đa nền tảng phong phú.



Hình 2.6. Kho ứng dụng Google Play Store

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Android có lượng ứng dụng của bên thứ ba ngày càng nhiều, được chọn lọc và đặt trên một cửa hàng ứng dụng như Google Play hay Amazon Appstore để người dùng lấy về, hoặc bằng cách tải xuống rồi cài đặt tập tin APK từ trang web khác. Các ứng dụng trên Play Store cho phép người dùng duyệt, tải về và cập nhật các ứng dụng do Google và các nhà phát triển thứ ba phát hành. Play Store được cài đặt sẵn trên các thiết bị thỏa mãn điều kiện tương thích của Google. Ứng dụng sẽ tự động lọc ra một danh sách các ứng dụng tương thích với thiết bị của người dùng, và nhà phát triển có thể giới hạn ứng dụng của họ chỉ dành cho những nhà mạng cố định hoặc những quốc gia cố định vì lý do kinh doanh. Nếu người dùng mua một ứng dụng mà họ cảm thấy không thích, họ được hoàn trả tiền sau 15 phút kể từ lúc tải về và một vài nhà mạng còn có khả năng mua giúp các ứng dụng trên Google Play, sau đó tính tiền vào trong hóa đơn sử dụng hàng tháng của người dùng. Đến tháng 9 năm 2012, có hơn 675.000 ứng dụng dành cho Android, và số lượng ứng dụng tải về từ Play Store ước tính đạt 25 tỷ.

2.3.4 Ưu và nhược điểm của hệ điều hành Android

Ưu điểm:

- Là hệ điều hành có mã nguồn mở nên khả năng tùy biến cao, có thể tùy ý chỉnh sửa.
- Kho ứng dụng Google Play Store đồ sộ.
- Thân thiện và dễ sử dụng.
- Khả năng đa nhiệm, chạy cùng lúc nhiều ứng dụng cao.

Nhược điểm:

- Màn hình cơ bản hệ điều hành Android cung cấp một giao diện người dùng đẹp và trực quan.
- Dễ nhiễm phần mềm độc hại và virus. Do tính chất mã nguồn mở, nhiều phần mềm không được kiểm soát có chất lượng không tốt dễ gây hại cho thiết bị.
- Khả năng bảo mật không cao qua đó người dùng có thể bị đánh cắp thông tin qua các ứng dụng.
- Kho ứng dụng quá nhiều dẫn đến khó kiểm soát chất lượng các ứng dụng.
- Hiện tượng giật lag trong quá trình sử dụng diễn ra thường xuyên do đặc trưng hệ điều hành sản sinh ra nhiều file rác.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.4 TỔNG QUAN VỀ TRANG TÍNH GOOGLE SHEET

2.4.1 khái niệm về trang tính google sheet

Google Sheets (Google SpreadSheet) là một chương trình bảng tính được bao gồm như một phần của bộ phần mềm văn phòng dựa trên web miễn phí được cung cấp bởi Google trong dịch vụ Google Drive của mình. Hay nói một cách ngắn gọn thì Google Sheets là ứng dụng về bảng tính của Google Docs.



Hình 2.7. Ứng dụng trang tính google sheet

Với tính năng AI (trí tuệ nhân tạo), bạn có thể tận dụng thông tin chi tiết phù hợp để đưa ra những quyết định có ý nghĩa trong công việc. Kiến trúc dựa trên đám mây giúp bạn cộng tác với bất kỳ ai, mọi lúc, mọi nơi. Khả năng tương thích với các hệ thống bên ngoài, trong đó có Microsoft Office, loại bỏ sự phiền hà khi bạn làm việc với nhiều nguồn dữ liệu. Ngoài ra, nhờ xây dựng trên cơ sở hạ tầng của Google, Trang tính giúp mọi người thỏa sức sáng tạo mà vẫn giúp bảo mật thông tin của bạn.

2.4.2 Ưu điểm của trang tính google sheet

So với Excel, có rất nhiều tính năng của Google Sheets thực hiện tốt và thậm chí còn vượt trội hơn, các tính năng khiến Google Sheets trở nên vượt trội:

- Form dễ dàng nhập dữ liệu

Sheets cung cấp cho bạn một tính năng để dễ dàng nhập dữ liệu của khách hàng khi truy cập vào trang web của bạn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Với một form nhập liệu, bạn có thể thiết lập các câu hỏi để khách hàng truy cập để điền vào. Khi người dùng điền vào các câu trả lời, dữ liệu đó sẽ tự động được đăng nhập và phân loại trên Google Sheets của bạn.

- Kết nối các bảng tính với nhau

Mọi người thường ngần ngại sử dụng Sheets vì nghĩ rằng Sheets không đủ các tính năng như Excel. Tuy nhiên hầu hết các tính năng đó đều có trên Google Sheets nếu bạn biết chỗ để tìm kiếm chúng.

Ví dụ tính năng lấy dữ liệu từ bảng tính khác. Bằng cách sử dụng các hàm như =IMPORTRANGE là có thể thực hiện và nhận được dữ liệu từ bảng tính hoàn toàn riêng biệt.

- Thích hợp cho làm việc nhóm

Với Google Sheets, bạn có thể mời nhiều người vào cùng làm việc với mình. Đây là một tính năng mà Sheets đã tiên phong cho mọi ứng dụng khác. Bạn có thể tạo URL công cộng để cho bất cứ ai xem bảng tính, hoặc mời cộng tác viên cụ thể bằng cách nhấp vào nút Share (Chia sẻ) ở góc trên bên phải và lựa chọn các chế độ: Có thể xem, Có thể nhận xét, Có thể sửa.

2.5 ỨNG DỤNG GOOGLE SHEET CHO ỨNG DỤNG IOTS

2.5.1 Các phương thức sử dụng để truyền tải dữ liệu

Trong thời điểm mà mọi người đều phải làm việc trực tuyến thì Google Sheets (Google Trang tính) sẽ rất hữu ích, giúp một nhóm có thể làm việc tương tác trên cùng một file bảng tính. Hơn thế Google Sheets cũng có ưu thế đa nền tảng. Một người có thể tham gia trên Google Sheets bằng máy tính, cũng có thể bằng điện thoại với các app Google Sheets trong kho ứng dụng của Apple và Google.

Google Sheets là một ứng dụng bảng dữ liệu tương tự Microsoft Excel nhưng hoạt động online. Bình thường Google Sheets tự động giúp chúng ta lưu dữ liệu lên đám mây, tránh mất dữ liệu, đồng thời hỗ trợ làm một file dữ liệu ở nhiều nơi khác nhau.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Khi muốn liên kết giữa Google sheet với các ứng dụng IoTs thì trước hết chúng ta cần phải có internet, thời buổi 4.0 thì internet chạy với tốc độ cực nhanh giúp chúng ta xử lý thông tin nhanh, gọn, lẹ và chính xác.

Để làm được điều đó thì đối với google sheet cần phải có tài khoản google. Nó sẽ giúp ta tạo ra được một trang tính online, các dữ liệu sẽ được đồng bộ và truyền nhận thông tin dễ dàng. Các thông tin sẽ được chuyển qua giao thức HTTP. Trong đó có các hàm như là hàm `GET` là phương thức yêu cầu dữ liệu đơn giản và thường sử dụng nhất của HTTP. Phương thức `GET` yêu cầu server chỉ trả về dữ liệu bằng việc cung cấp các thông tin truy vấn trên URL, thông thường Server căn cứ vào thông tin truy vấn đó trả về dữ liệu mà không thay đổi nó. Hàm `POST` tương tự như `GET`, nhưng `POST` có thể gửi dữ liệu về Server.

Khi cần truyền tải một lượng thông tin lớn thì người ta sẽ sử dụng chuỗi JSON (JavaScript Object Notation), nó là 1 định dạng trao đổi dữ liệu để giúp việc đọc và viết dữ liệu trở nên dễ dàng hơn, máy tính cũng sẽ dễ phân tích và tạo ra JSON.

JSON được xây dựng trên 2 cấu trúc:

Là tập hợp của các cặp tên và giá trị name-value. Trong những ngôn ngữ khác nhau, đây có thể là 1 object, record, struct, dictionary, hash table, keyed list hay associative array.

Là 1 tập hợp các giá trị đã được sắp xếp. Trong hầu hết các ngôn ngữ, dữ liệu này được xem như array, véc tơ, list hay sequence. Đây là 1 cấu trúc dữ liệu phổ dụng. Hầu như tất cả các ngôn ngữ lập trình hiện đại đều hỗ trợ. Chúng tạo nên ý nghĩa của 1 định dạng hoán vị dữ liệu với các ngôn ngữ lập trình cũng đã được cơ sở hoá trên cấu trúc này.

Cú pháp

- Dữ liệu nằm trong các cặp name/value
- Các dữ liệu được ngăn cách bởi dấu phẩy ,
- Các đối tượng (name/value) nằm giữa hai dấu ngoặc kép " "

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Tất cả các đối tượng nằm bên trong hai dấu ngoặc nhọn { }
- Dữ liệu của JSON được viết theo từng cặp name/value. Một cặp name/value bao gồm trường name (nằm trong hai dấu ngoặc kép " ", theo sau là dấu hai chấm : và sau cùng là trường value (cũng được nằm trong hai dấu ngoặc kép " ")

Ví dụ: "name": "John"

```
{
  "username" : "your-user-name",
  "email" : "your-email@email.com",
  "website" : "iota.edu.vn",
  "title" : "IoT Stater Course"
}
```

Hình 2.8 Chuỗi dữ liệu ghi dưới dạng JSON

2.5.2 Cách thức truyền dữ liệu giữa google sheets và các thiết bị IoTs

Đầu tiên mọi người cần tạo ra một trang tính sử dụng tài khoản google của chính mình. Chúng ta cần viết code điều khiển nhận và gửi dữ liệu của trang tính thông qua các hàm POST, GET... các dữ liệu này sẽ được gửi theo mã Current web app URL.

```
var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1hU08JBGLrwmKrVWfrjIqFrtzELhi3kNkpUZzHye42v8/edit#gid=0");
var sheet = ss.getSheetByName('Items');

//----- GOOGLE SHEET ----> ANDROID -----

function doPost(e)
{
  var action = e.parameter.action;

  if(action == 'addItem'){
    return addItem(e);
  }

  if(action == 'addItem1'){
    return addItem1(e);
  }

  if(action == 'getItems1')
  {
    return getItems1(e);
  }
}
```

Hình 2.9. Lập trình code cho google sheets

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Đối với module wifi Esp8266, sau khi khai báo các thư viện cần thiết chúng ta tiến hành lập trình đường dẫn để truyền tải các dữ liệu lên google sheet thông qua đường link URL đã lấy ở trên.

```
#include <MFRC522.h>
#define SS_PIN D4
#define RST_PIN D2

MFRC522 mfrc522(SS_PIN, RST_PIN);

int readsuccess;
byte readcard[4];
char str[32] = "";
String StrUID;

//Khai báo thông tin về Wifi
const char* ssid = "ZZZ";
const char* password = "tamchinmuoi";

const char *GScriptID = "AKfycbzaHeqVu_FVma6B_vrsoLoCM1R5MJDELLJ7x4MB1xNpKdiwKAXL";
const char* host = "script.google.com";
const char *googleRedirectHost = "script.googleusercontent.com";
const int httpsPort = 443;

HTTPSRedirect client(httpsPort);

//Khai báo URL chứa thông tin gửi dữ liệu
String url = String("/macros/s/") + GScriptID + "/exec?";
```

Hình 2.10. Khai báo thư viện và các lệnh cài đặt để truyền dữ liệu

Khi hoạt động các dữ liệu sẽ được truyền tải qua lại với google sheet dưới dạng chuỗi string thông qua link URL. Đường link đó sẽ chứa lệnh action để lựa chọn các chương trình con, ví dụ như nếu action là “additem” thì function additem sẽ chạy và cho phép ghi các dữ liệu lên trang tính, ngược lại thì lấy dữ liệu gửi đi cũng tương tự.

Còn đối với các app trên nền tảng android thì cũng tương tự. khi lập trình chúng ta cũng sẽ xử dụng đường link chứa các lệnh action để lựa chọn các function trong trang tính google sheet. Nếu dữ liệu quá nhiều thì trang tính sẽ ghi các dữ liệu dưới dạng chuỗi JSON và bắt đầu truyền qua app, chương trình android sẽ tiến hành xử lý chuỗi JSON đó và xuất dữ liệu ra màn hình.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

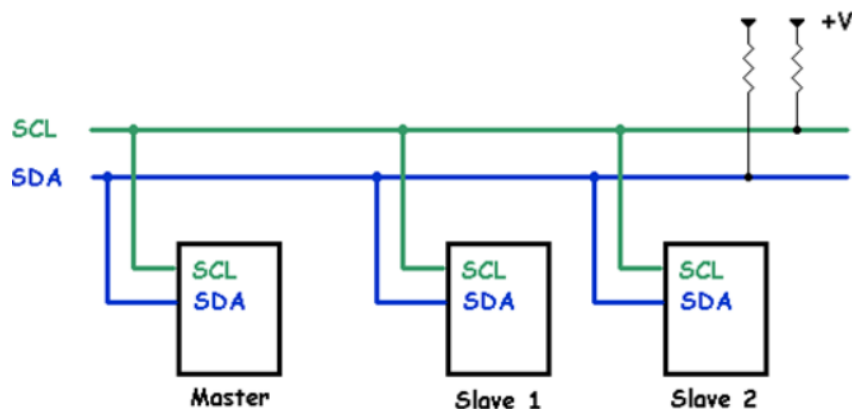
```
private void parseItems(String jsonResposnce) {  
  
    ArrayList<HashMap<String, String>> list = new ArrayList<>();  
  
    try {  
        JSONObject jsonObj = new JSONObject(jsonResposnce);  
        JSONArray jarray = jsonObj.getJSONArray( name: "items");  
  
        for (int i = 0; i < jarray.length(); i++) {  
  
            JSONObject jo = jarray.getJSONObject(i);  
  
            String id = jo.getString( name: "a");  
            String name = jo.getString( name: "d");  
            String sdt = jo.getString( name: "e");  
            String diachi = jo.getString( name: "f");  
            String chungloai = jo.getString( name: "g");  
            String solantiem = jo.getString( name: "h");  
            String ngaytiem = jo.getString( name: "j");  
  
        }  
    }  
}
```

Hình 2.11. Xử lý các chuỗi JSON

2.6 CÁC CHUẨN TRUYỀN DỮ LIỆU

2.6.1 Chẩn giao tiếp I2C

I2C là tên viết tắt của cụm từ Inter-Integrated Circuit. Đây là đường Bus giao tiếp giữa các IC với nhau. Bus I2C được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, ARM



Hình 2.12. Bus I2C và các thiết bị ngoại vi

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Đặc điểm giao tiếp I2C:

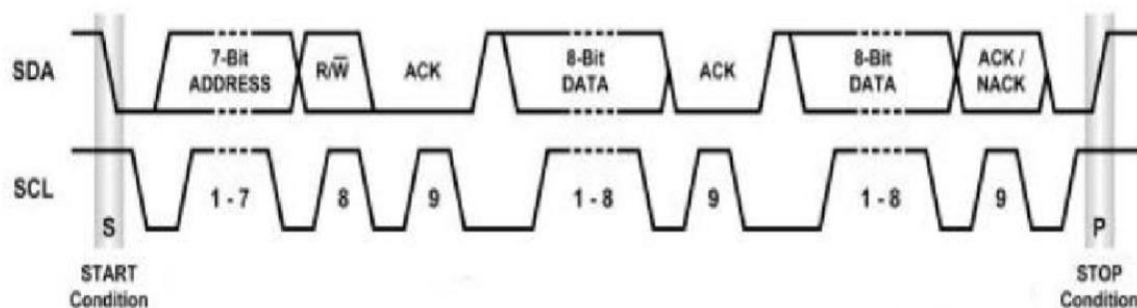
Một giao tiếp I2C gồm có 2 dây: Serial Data (SDA) và Serial Clock (SCL). SDA là đường truyền dữ liệu 2 hướng, còn SCL là đường truyền xung đồng hồ để đồng bộ và chỉ theo một hướng. Khi một thiết bị ngoại vi kết nối vào đường bus I2C thì chân SDA của nó sẽ nối với dây SDA của bus, chân SCL sẽ nối với dây SCL.

Mỗi dây SDA hay SCL đều được nối với điện áp dương của nguồn cấp thông qua một điện trở kéo lên (pullup resistor). Sự cần thiết của các điện trở kéo này là vì chân giao tiếp I2C của các thiết bị ngoại vi thường là dạng cực máng hở (opendrain hay opencollector). Giá trị của các điện trở này khác nhau tùy vào từng thiết bị và chuẩn giao tiếp, thường dao động trong khoảng 1 K Ω đến 4.7K Ω .

❖ Chế độ hoạt động (tốc độ truyền):

Các bus I2C có thể hoạt động ở ba chế độ, hay nói cách khác các dữ liệu trên bus I2C có thể được truyền trong ba chế độ khác nhau.

- Chế độ tiêu chuẩn (Standard mode)
- Chế độ nhanh (Fast mode)
- Chế độ cao tốc High-Speed (Hs) mode



Hình 2.13. Trình tự truyền bit trên đường truyền

Trình tự truyền bit trên đường truyền:

- Thiết bị chủ tạo một điều kiện start. Điều kiện này thông báo cho tất cả các thiết bị tớ lắng nghe dữ liệu trên đường truyền.
- Thiết bị chủ gửi địa chỉ của thiết bị tớ mà thiết bị chủ muốn giao tiếp và cờ đọc/ghi dữ liệu (nếu cờ thiết lập lên 1 thì byte tiếp theo được truyền từ thiết bị tớ đến thiết bị chủ, nếu cờ thiết lập xuống 0 thì byte tiếp theo truyền từ thiết bị chủ đến thiết bị tớ).

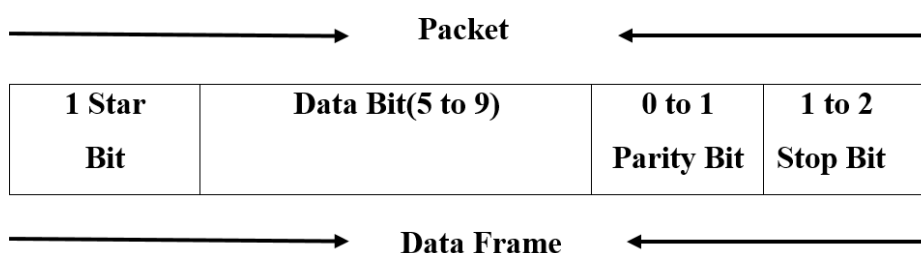
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Khi thiết bị tớ trên bus I2C có địa chỉ đúng với địa chỉ mà thiết bị chủ gửi sẽ phản hồi lại bằng một xung ACK.
- Giao tiếp giữa thiết bị chủ và tớ trên bus dữ liệu bắt đầu. Cả chủ và tớ đều có thể nhận hoặc truyền dữ liệu tùy thuộc vào việc truyền thông là đọc hay viết. Bộ truyền gửi 8bit dữ liệu tới bộ nhận, bộ nhận trả lời với một bit ACK.
- Để kết thúc quá trình giao tiếp, thiết bị chủ tạo ra một điều kiện stop.

2.6.2 Chuẩn truyền thông UART

UART viết tắt của “Universal Asynchronous Receiver / Transmitter” là kiểu truyền thông tin nối tiếp không đồng bộ thường là một mạch tích hợp. Chức năng chính của UART là truyền dữ liệu nối tiếp. Trong UART, giao tiếp giữa hai thiết bị có thể được thực hiện theo hai cách là giao tiếp dữ liệu nối tiếp và giao tiếp dữ liệu song song.

Việc truyền dữ liệu của UART có thể được thực hiện bằng cách sử dụng bus dữ liệu ở dạng song song bởi các thiết bị khác như vi điều khiển, bộ nhớ, CPU,.. Sau khi nhận được dữ liệu song song từ bus, nó tạo thành gói dữ liệu bằng cách thêm ba bit như bắt đầu, dừng lại và trung bình. Nó đọc từng bit gói dữ liệu và chuyển đổi dữ liệu nhận được thành dạng song song để loại bỏ ba bit của gói dữ liệu. Tóm lại, gói dữ liệu nhận được bởi UART chuyển song song về phía bus dữ liệu ở đầu nhận.



Hình 2.14. Truyền thông UART

- Start-bit

Start-bit còn được gọi là bit đồng bộ hóa được đặt trước dữ liệu thực tế. Nói chung, một đường truyền dữ liệu không hoạt động được điều khiển ở mức điện áp cao. Để bắt đầu truyền dữ liệu, truyền UART kéo đường dữ liệu từ mức điện áp cao (1) xuống mức điện áp thấp (0). UART thu được thông báo sự chuyển đổi này từ mức cao sang mức thấp qua đường dữ liệu cũng như bắt đầu hiểu dữ liệu thực. Chỉ có một start-bit.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Bit dừng

Bit dừng được đặt ở phần cuối của gói dữ liệu. Thông thường, bit này dài 2 bit nhưng thường chỉ sử dụng 1 bit. Để dừng sóng, UART giữ đường dữ liệu ở mức điện áp cao.

- Bit chẵn lẻ

Bit chẵn lẻ cho phép người nhận đảm bảo liệu dữ liệu được thu thập có đúng hay không. Đây là một hệ thống kiểm tra lỗi cấp thấp và bit chẵn lẻ có sẵn trong hai phạm vi như Chẵn lẻ – chẵn lẻ cũng như Chẵn lẻ – lẻ.

- Dữ liệu bit hoặc khung dữ liệu

Các bit dữ liệu bao gồm dữ liệu thực được truyền từ người gửi đến người nhận. Độ dài khung dữ liệu có thể nằm trong khoảng 5 và 8. Nếu bit chẵn lẻ không được sử dụng thì chiều dài khung dữ liệu có thể dài 9 bit. Nói chung, LSB của dữ liệu được truyền trước tiên sau đó nó rất hữu ích cho việc truyền.

2.6.3 Chuẩn truyền thông SPI

SPI (Serial Peripheral Interface, SPI bus) là một chuẩn đồng bộ nối tiếp để truyền dữ liệu ở chế độ song công toàn phần.

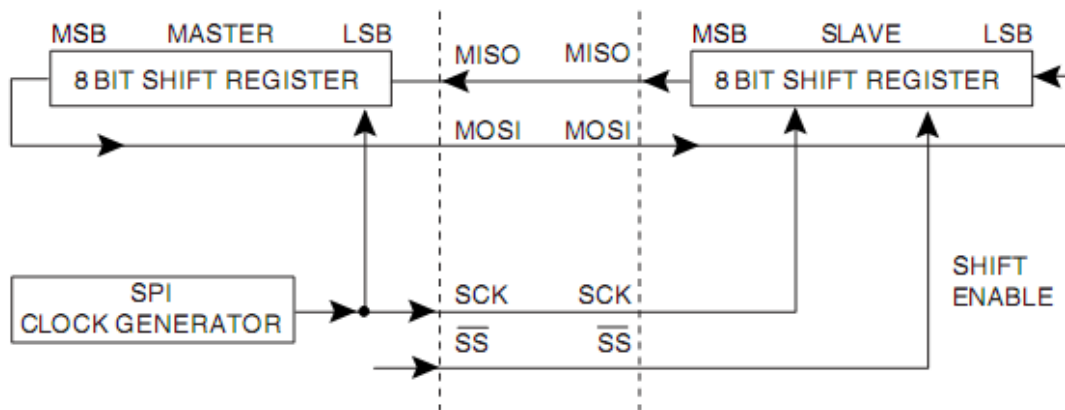
Khác với cổng nối tiếp chuẩn (standard serial port), SPI là giao diện đồng bộ, trong đó bất cứ quá trình truyền nào cũng được đồng bộ hóa với tín hiệu xung clock, tín hiệu này sinh ra bởi thiết bị master (thiết bị chủ động). Thiết bị ngoại vi bên phía nhận (bị động) làm đồng bộ quá trình nhận chuỗi bit với tín hiệu xung clock. Có thể kết nối một số vi mạch vào mỗi giao diện ngoại vi nối tiếp của vi mạch-thiết bị master. Thiết bị master chọn thiết bị động để truyền dữ liệu bằng cách kích hoạt tín hiệu "chọn chip" (chip select) trên vi mạch bị động. Thiết bị ngoại vi nếu không được chọn bởi bộ vi xử lý sẽ không tham gia vào quá trình truyền theo giao diện SPI.

- Trong giao diện SPI có sử dụng bốn tín hiệu số:

- **MOSI** (Master Out Slave In) hay SI - cổng ra của bên master, cổng vào của bên bị động, dành cho việc truyền dữ liệu từ thiết bị master đến thiết bị slave.
- **MISO** (Master In Slave Out) hay SO - cổng vào của bên master, cổng ra của bên bị động, dành cho việc truyền dữ liệu từ thiết bị slave đến thiết bị master.
- **SCLK** (Serial Clock) hay SCK - tín hiệu xung clock nối tiếp, dành cho việc truyền tín hiệu dành cho thiết bị slave.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- CS hay SS (Chip Select, Slave Select): chọn vi mạch, chọn thiết bị slave.



Hình 2.15. Giao tiếp SPI

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

3.1 GIỚI THIỆU

Đối với đề tài “*Xây dựng hệ thống quản lý tiêm ngừa chó mèo*” gồm có các yêu cầu đặt ra như sau:

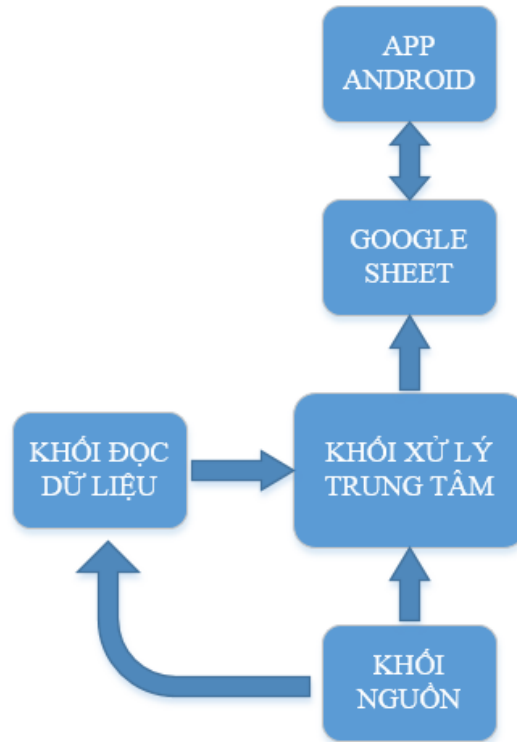
- Thiết kế khối xử lý trung tâm giao tiếp với module RFID giúp đọc và ghi các thông tin cần thiết vào thẻ.
- Thiết kế khối đọc và ghi thẻ RFID.
- Thiết kế bảng dữ liệu sử dụng google sheet để hiển thị trên màn hình máy tính.
- Thiết kế 2 app android: app sử dụng cho bác sĩ có chức năng để kiểm soát và ghi các thông tin cần thiết vào, app sử dụng cho chủ của vật nuôi chỉ sử dụng để xem các thông tin cần thiết.

Qua những yêu cầu đã đề ra nhóm sẽ tiến hành tính toán và thiết kế cho hệ thống.

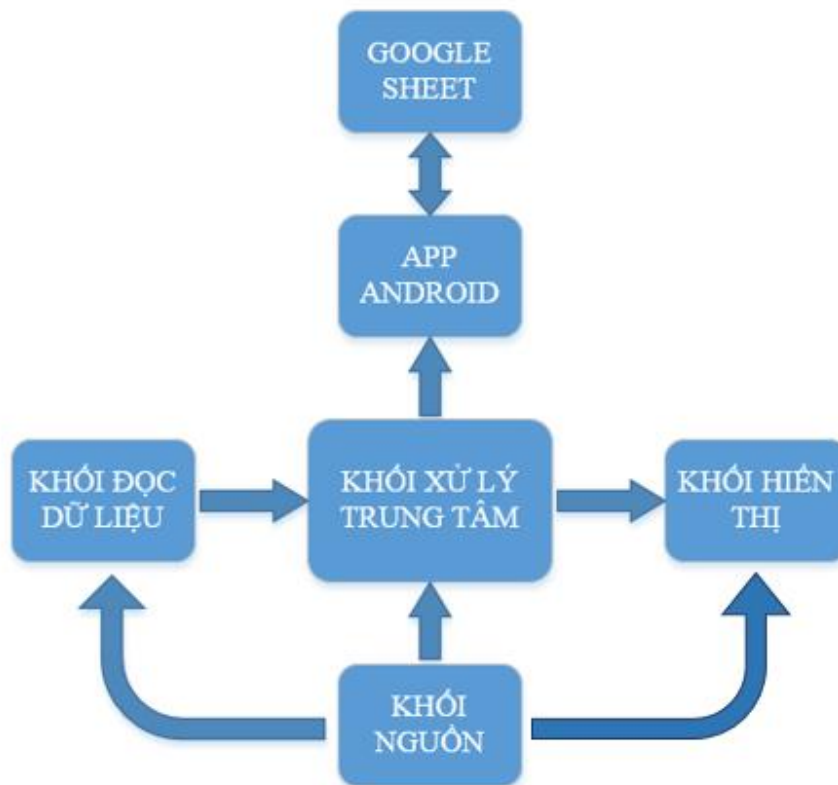
3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

3.2.1 Thiết kế sơ đồ khối hệ thống

Với mỗi đề tài, để tạo nên một sản phẩm hoàn thiện chúng ta cần khảo sát, tìm hiểu theo từng bước để biết được các thành phần, cấu tạo cũng như hoạt động của chúng. Sơ đồ khối giúp ta có cái nhìn tổng quan hơn về đề tài. Từ đó hình dung được những công việc cần làm và phân chia thời gian cụ thể, hợp lý trong quá trình thực hiện. Chính vì vậy, để làm được đề tài theo yêu cầu đã đặt ra, nhóm thiết kế sơ đồ khối như hình 3.1, và hình 3.2.



Hình 3.1. Sơ đồ khối hệ thống đặt ở trạm y tế



Hình 3.2. Sơ đồ khối hệ thống dành cho đội bắt chó mèo thả rong

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

❖ Chức năng từng khối:

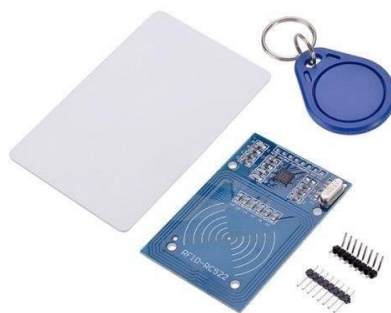
- **Khối xử lý trung tâm:** Khối này có chức năng kết nối mạng wifi để truyền dữ liệu lên google sheet và giao tiếp với module RFID,
- **Khối đọc dữ liệu:** Khối này có nhiệm vụ là đọc thông tin được lưu trong thẻ hoặc ghi thông tin vào thẻ mới để truyền thông tin lên module wifi để xử lý thông tin.
- **Khối nguồn:** Cung cấp nguồn liên tục cho toàn bộ hệ thống,
- **App Android:** Là một ứng dụng android có chức năng hiển thị các thông tin liên quan đến vật nuôi.
- **Google sheet:** Là nơi lưu trữ thông tin trực tuyến, tự động đồng bộ thông tin ngay lập tức.
- **Khối hiển thị:** là một màn hình LCD dùng để hiển thị số UID sau mỗi lần quét thẻ.

3.3 TÍNH TOÁN VÀ THIẾT KẾ MẠCH

Đối với đề tài này nhóm chỉ dừng lại ở việc tính toán các yêu cầu và lựa chọn các Module có sẵn trên thị trường phù hợp với các yêu cầu. Và nhóm sẽ không có phần thiết kế, làm mạch cho từng khối riêng.

3.3.1 Khối module RFID

Trên thị trường có rất nhiều loại đầu đọc thẻ RFID. Nó đa dạng về kiểu, màu sắc, loại kết nối như đầu đọc RFID HF, đầu đọc RFID UHF... Nhưng vì đầu đọc RFID HF là loại có giá thành thấp nhất trên thị trường, cho phép ghi/ đọc dữ liệu lên thẻ cũng như tương thích kết nối với đa số vi xử lý trên thị trường. Vì những thuận lợi mà thiết bị này mang đến nên nhóm nghiên cứu đã chọn module này làm thiết bị đọc RFID chính.



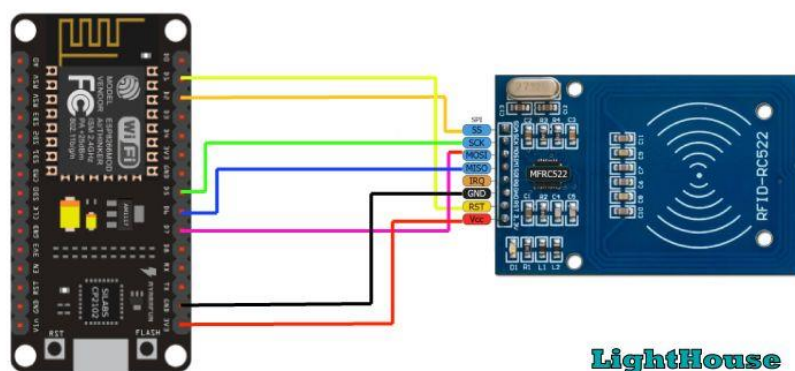
Hình 3.3. Module RFID RC522

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Thông số kỹ thuật:

- Nguồn: 3.3Vdc, 13 - 26mA
- Dòng ở chế độ chờ: 10 - 13mA
- Dòng ở chế độ nghỉ: <80uA
- Tần số sóng mang: 13.56MHz
- Khoảng cách hoạt động: 0~60mm (mifare1 card)
- Giao tiếp: SPI
- Tốc độ truyền dữ liệu: tối đa 10Mbit/s
- Các loại card RFID hỗ trợ: mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire
- Kích thước: 40mm × 60mm

Module wifi Node MCU ESP8266 kết nối với vi điều khiển có sơ đồ như sau:



Hình 3.4. Sơ đồ kết nối module RFID RC522 với module Wifi

- Giao tiếp giữa module RFID và module Wifi ESP8266 theo chuẩn SPI.
- Điện áp sử dụng 3.3V.
- Dòng điện 13 - 26mA
- Mô tả chi tiết các chân của module RFID RC522 kết nối với module Wifi:

Chân RST/Reset: kết nối với chân D1.

Chân SDA: kết nối với chân D2.

Chân MOSI: kết nối với chân D7.

Chân MISO: kết nối với chân D6.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Chân SCK: kết nối với chân D5.

Chân 3.3V: kết nối với chân 3.3V.

Chân GND: kết nối với chân GND.

3.3.2. Khối module wifi

Để đơn giản cho người sử dụng ở mọi lứa tuổi, ngành nghề, dự án trên thị trường xuất hiện rất nhiều Module mạch phát triển toàn diện phù hợp với các yêu cầu trên. Một số Module ESP8266 điển hình như: ESP-01, ESP-02, ESP-03, ESP-07, ESP-12F, ... và một số Module mạch phát triển như: NodeMCU, Wemos, ... Trong đó, có ba loại được sử dụng nhiều và cũng khá là tốt như ESP-01, ESP-12E Node MCU và Wemos D1 Mini. Với giá thành hợp lý, có thể đáp ứng nhu cầu các dự án tiên tiến và một điều đặc biệt là các Module mạch trên còn tương thích với Arduino IDE rất tiện lợi cho việc lập trình để phát triển.

Sau khi tìm hiểu so sánh một số loại Module mạch phát triển nhóm chúng em lựa chọn Module ESP-12E Node MCU. Với các đặc điểm đó là nhiều chân I/O, giá cả hợp lý, rất dễ tìm và mua ngoài thị trường, nạp chương trình điều khiển cho Module dễ dàng thông qua Arduino IDE và cổng COM của máy tính.



Hình 3.5. Module Wifi NodeMCU ESP8266

Thông số kỹ thuật:

- Chip: ESP8266EX
- WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
- Điện áp hoạt động: 3.3V

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Điện áp vào: 5V thông qua cổng USB
- Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
- Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
- Bộ nhớ Flash: 4MB
- Giao tiếp: Cable Micro USB
- Hỗ trợ bảo mật: WPA/WPA2
- Tích hợp giao thức TCP/IP
- Lập trình trên các ngôn ngữ: C/C++, Micropython, NodeMCU – Lua
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Hỗ trợ nhiều loại anten.
- 16 chân GPIO.
- Hỗ trợ SDIO 2.0, UART, SPI, I2C, PWM, I2S với DMA.
- 1 ADC 10-bit.
- Dải nhiệt độ hoạt động rộng: - 40 ~ 125°C.

3.3.3 Vi điều khiển

- Hiện nay trên thị trường có rất nhiều phiên bản Arduino như Arduino Uno R3, Arduino Uno R3 CH340, Arduino Mega256... Với những đặc tính của hệ thống, nhóm lựa chọn vi điều khiển trung tâm là Arduino Uno R3
- Hình ảnh thực tế của board arduino uno R3:



Hình 3.6. Board arduino uno R3

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Thông số kỹ thuật:

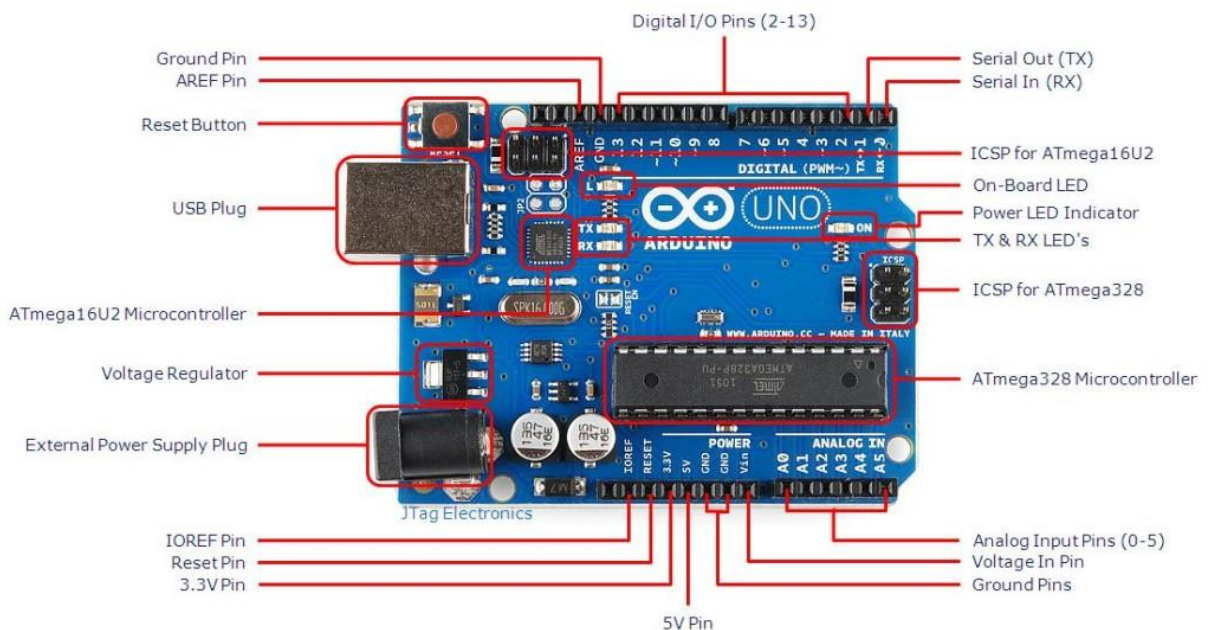
- Chip điều khiển chính: ATmega328P
- Chip nạp và giao tiếp UART: ATmega16U2
- Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ giắc tròn DC (nếu sử dụng nguồn ngoài từ giắc tròn DC Hshop.vn khuyên bạn nên cấp nguồn từ 6~9VDC để đảm bảo mạch hoạt động tốt, nếu bạn cắm 12VDC thì IC ổn áp rất nóng, dễ cháy và gây hư hỏng mạch).
- Số chân Digital I/O: 14 (trong đó 6 chân có khả năng xuất xung PWM).
- Số chân PWM Digital I/O: 6
- Số chân Analog Input: 6
- Dòng điện DC Current trên mỗi chân I/O: 20 Ma
- Dòng điện DC Current chân 3.3V: 50 Ma
- Flash Memory: 32 KB (ATmega328P), 0.5 KB dùng cho bootloader.
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Clock Speed: 16 MHz
- Kích thước: 68.6 x 53.4 mm

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Bảng 3.1. Thông số Arduino Uno R3

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

- Sơ đồ chân và kí hiệu chân trên board arduino uno R3:



Hình 3.7. Sơ đồ và kí hiệu chân trên arduino uno R3

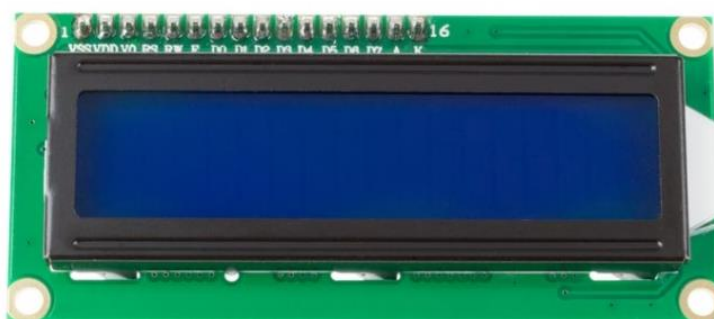
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

3.3.4 Khối LCD

Đối với khối hiển thị thì ta có 2 sự lựa chọn chính phổ biến nhất hiện nay, đó là màn hình LCD và led 7 đoạn. Nhưng với LCD ta có thể hiển thị nhiều nội dung, giao tiếp đơn giản và ít tốn chân của vi điều khiển với sự hỗ trợ của mạch chuyển đổi giao tiếp I2C cho LCD, còn với led 7 đoạn thì ta hiển thị được ít nội dung hơn.

Nội dung cần hiển thị là các kí tự mang tên “bắt đầu”, “ mã uid” và các đoạn mã vừa có kí tự vừa có khoảng 3, 4 chữ số. Như vậy ta cần 1 màn hình có khả năng hiển thị ít nhất 15 kí tự mỗi hàng và cần tới 2 hàng đối để hiển thị hết tất cả nội dung cần thiết. Dựa vào bảng so sánh bên trên và nội dung cần hiển thị nhóm em quyết định chọn module LCD 1602

LCD có rất nhiều loại khác nhau được phân biệt theo kích thước ví dụ: LCD 1602 nghĩa là có 2 hàng mỗi hàng gồm 16 kí tự, LCD 2004 nghĩa là có 4 hàng mỗi hàng gồm 20 kí tự.



Hình 3.8. Hình ảnh mặt trước của LCD 1602

Chi tiết chức năng các chân của LCD 1602 trong bảng 2.1:

Bảng 3.2 Thông số các chân của LCD 1602

Chân số	Tên chân	Input/Output	Chức năng tín hiệu
1	Vss	Power	GND

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2	Vdd	Power	+3V đến +5V
3	Vo	Analog	Điều khiển độ tương phản
4	RS	Input	Thanh ghi lựa chọn tín hiệu H: data signal, L: instruction signal
5	R/W	Input	Đọc/Ghi. H: read mode, L: write mode
6	E	Input	Tín hiệu cho phép H → L
7	DB0	I/O	Dữ liệu (LSB)
8	DB1	I/O	Dữ liệu
9	DB2	I/O	Dữ liệu
10	DB3	I/O	Dữ liệu
11	DB4	I/O	Dữ liệu
12	DB5	I/O	Dữ liệu
13	DB6	I/O	Dữ liệu
14	DB7	I/O	Dữ liệu (MSB)
15	A/V _{ee}	Input	4,2V cho đầu ra điện áp LED/tiêu cực
16	K	Input	Nguồn cho B/L (0V)

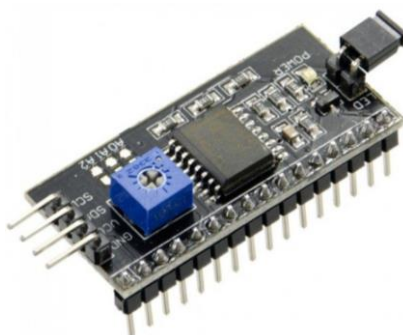
Chân của LCD được chia làm 3 dạng tín hiệu:

- Các chân cấp nguồn: Chân số 1 là chân nối GND, chân thứ 2 nối với nguồn 5VDC. Chân thứ 3 dùng để chỉnh độ sáng màn hình LCD thường nối với biến trở để thuận tiện cho việc thay đổi độ sáng.
- Các chân điều khiển: Chân số 4 là chân RS dùng điều khiển lựa chọn thanh ghi. Chân R/W dùng để điều khiển quá trình đọc và ghi. Chân E là chân cho phép dạng xung chốt.
- Các chân dữ liệu D7-D0: Chân số 7 đến chân số 14 là tám chân dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

3.3.5 Module I2C

Việc sử dụng LCD để hiển thị rất tiện lợi nhưng có một điều bất tiện là LCD có quá nhiều chân gây khó khăn trong quá trình đi dây và chiếm nhiều chân của Module ESP-12E Node MCU. Để hệ thống đơn giản và gọn gàng hơn nhóm chọn Module I2C giao tiếp với LCD, thay vì dùng tối thiểu 6 chân của Module ESP-12E Node MCU để kết nối với LCD (RS, EN, D7, D6, D5 và D4) thì với Module I2C chỉ cần dùng hai chân SCL và SDA. Trong đó, chân SCL có tác dụng đồng bộ hóa giữa các thiết bị khi truyền dữ liệu, còn chân SDA là chân cho dữ liệu truyền qua.

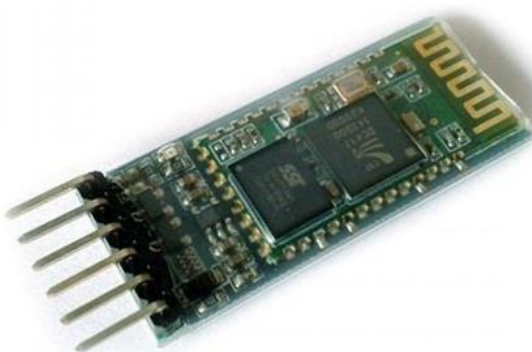


Hình 3.9. Module chuyển đổi I2C cho LCD 1602

Một vài thông số kỹ thuật:

- Điện áp hoạt động: 2,5VDC – 6VDC
- Hỗ trợ màn hình: LDC1602, 1604, 2004 (driver HD44780)
- Giao tiếp: I2C
- Địa chỉ mặc định: 0X27
- Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

3.3.6 Module Bluetooth



Hình 3.10. Module Bluetooth HC05

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Thông số kỹ thuật:

- Điện áp hoạt động: 3.3 - 5V.
- Dòng điện khi hoạt động: khi Pairing 30 mA, sau khi pairing hoạt động truyền nhận bình thường 8 mA
- Baudrate UART có thể chọn được: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
- Kích thước của module chính: 28 mm x 15 mm x 2.35 mm

3.3.7 Khối nguồn

Các mức điện áp của các linh kiện sử dụng trong mạch là:

- Module RC522 hoạt động ở mức điện áp 3.3VDC, dòng tiêu thụ từ (13-26) mA.
- Module ESP8266 sử dụng ở mức điện áp 3.3VDC và 5V thông qua cổng USB, làm việc từ 12-200 mA.

Kết luận: Để đảm bảo thời gian để thiết kế app hiển thị nhóm tiến hành lựa chọn sử dụng nguồn 5V-1A từ adapter để cung cấp cho module ESP8266.

Đối với module sử dụng cho các nhân viên đô thị thì thông số của các IC sẽ hiển thị ở bảng sau:

Bảng 3.3: Thông số, giá trị các linh kiện sử dụng

STT	Tên linh kiện	Số lượng	Điện áp (VDC)	Dòng điện (mA)	Tổng dòng tiêu thụ (mA)
1	Arduino Uno R3	1	5-12	30	200
2	Module bluetooth	1	5	30	100
3	LCD 1602	1	5	600uA	150
4	Module RFID	1	3.3	26	150
Tổng dòng tiêu thụ của hệ thống (mA)				86.6	

Với các thông số tính toán ở trên cùng với đặc tính hệ thống, nhóm lựa chọn Pin Cell 18650 4200mAh 3.7V kết hợp với mạch sạc và tăng áp để cấp nguồn dự phòng cho hệ thống.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

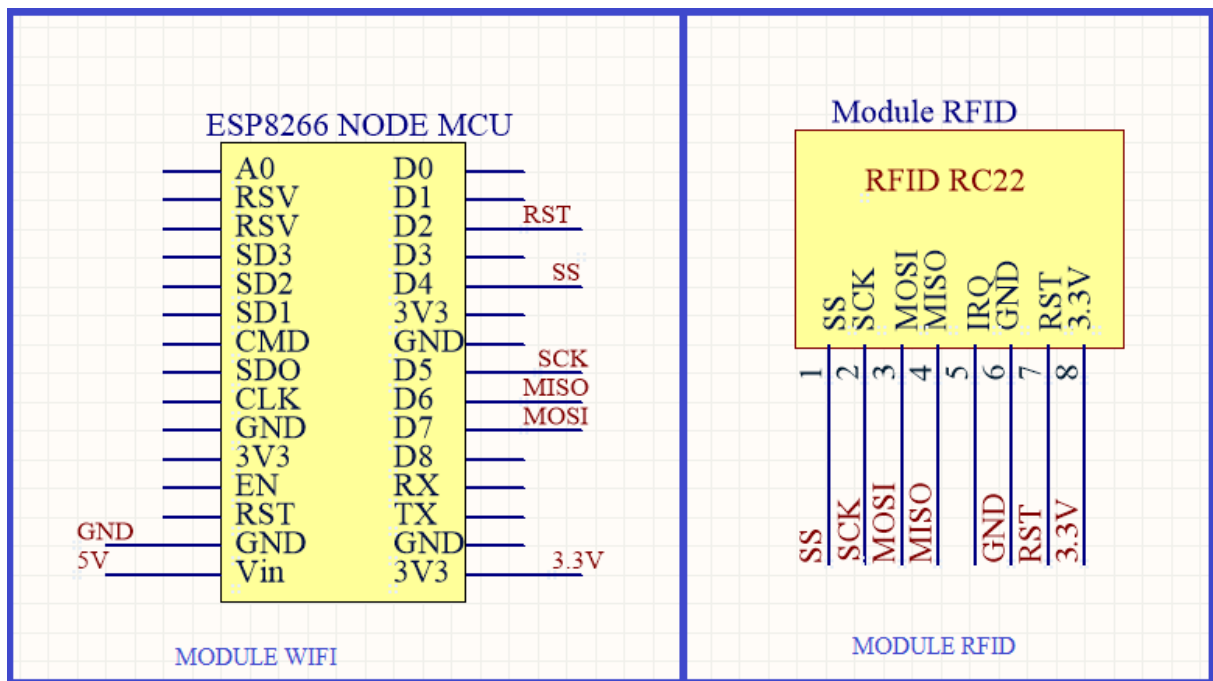


Hình 3.11. Pin Cell 18650 4200mAh 3.7V



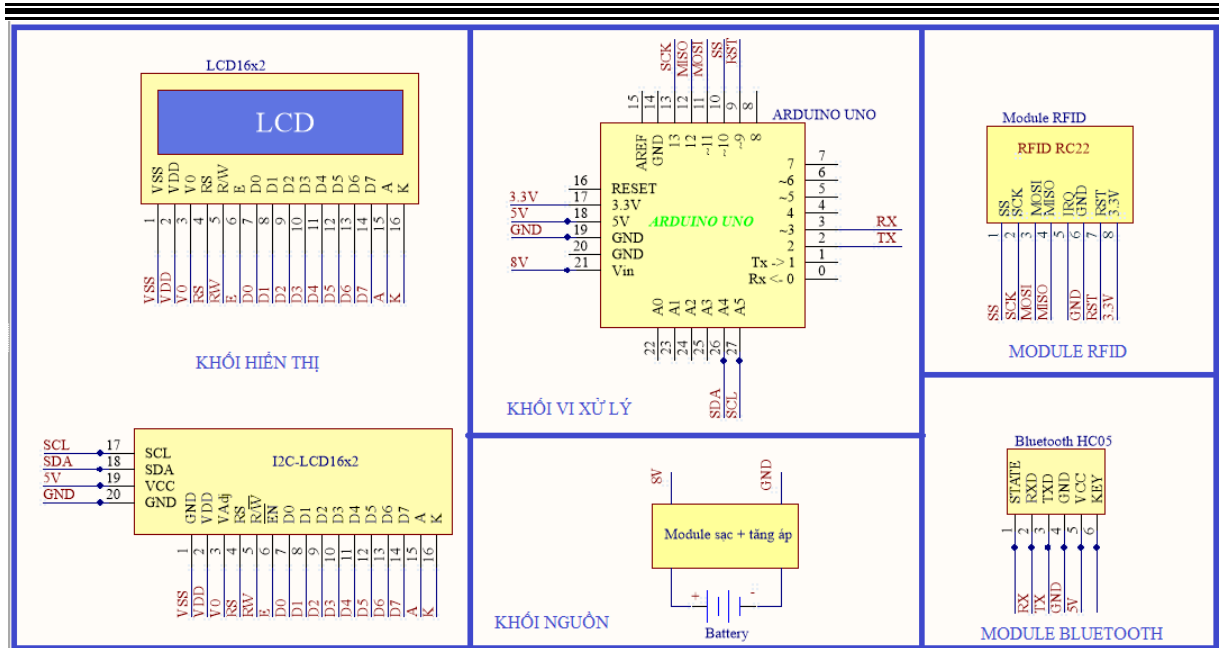
Hình 3.12. Mạch sạc bảo vệ pin.

3.4 SƠ ĐỒ NGUYÊN LÝ TOÀN MẠCH



Hình 3.13. Sơ đồ nguyên lý mạch trong trung tâm y tế

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 3.14. Sơ đồ nguyên lý toàn mạch của nhân viên đô thị

CHƯƠNG 4: THI CÔNG HỆ THỐNG

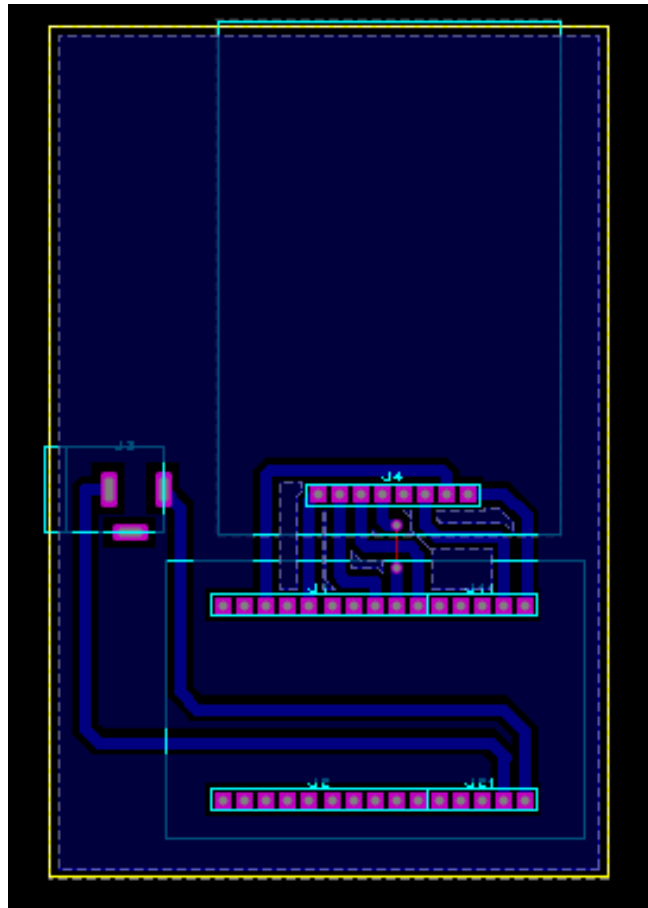
4.1 GIỚI THIỆU

Trong chương này là quá trình thi công PCB, lập trình, lắp ráp phần cứng và kiểm tra mạch. Bên cạnh đó là hình vẽ được chụp từ mô hình thực tế của hệ thống, hình chụp các kết quả chạy của hệ thống.

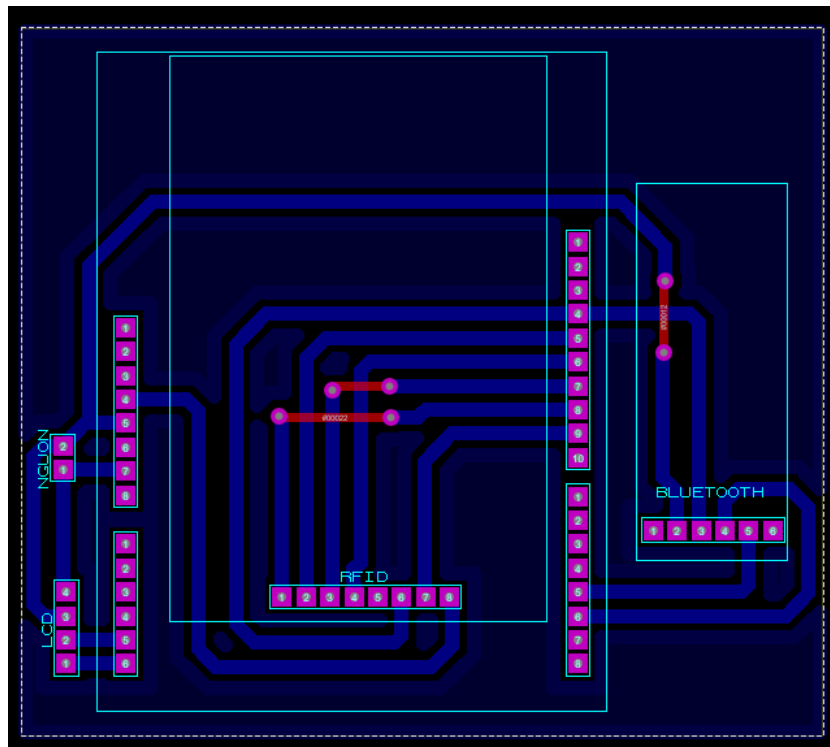
4.2 THI CÔNG HỆ THỐNG

4.2.1 Thi công Board mạch

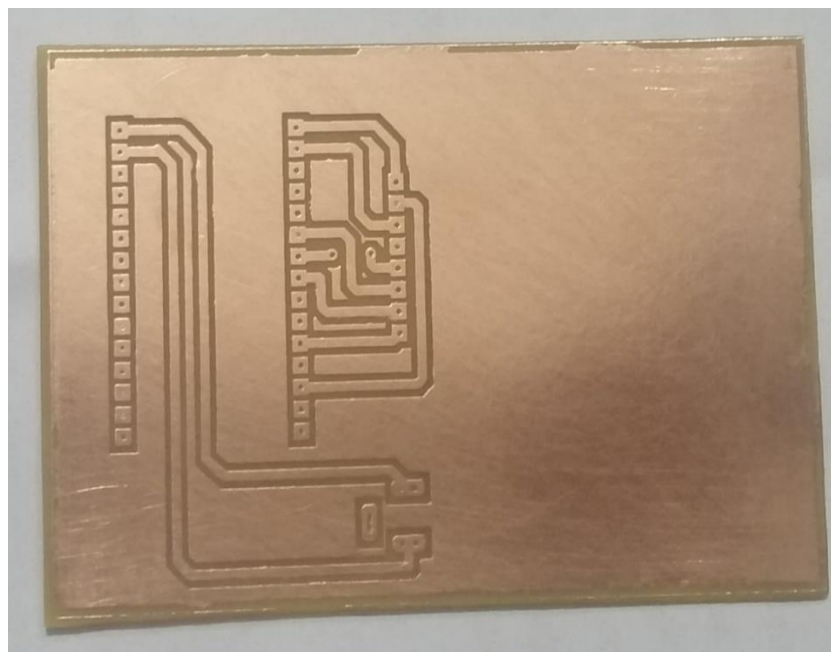
Do chỉ cần thiết kế mạch đơn giản một lớp nên nhóm quyết định thiết kế board mạch in bằng phần mềm Proteus. Cụ thể hình ảnh về Board mạch được thiết kế như sau:



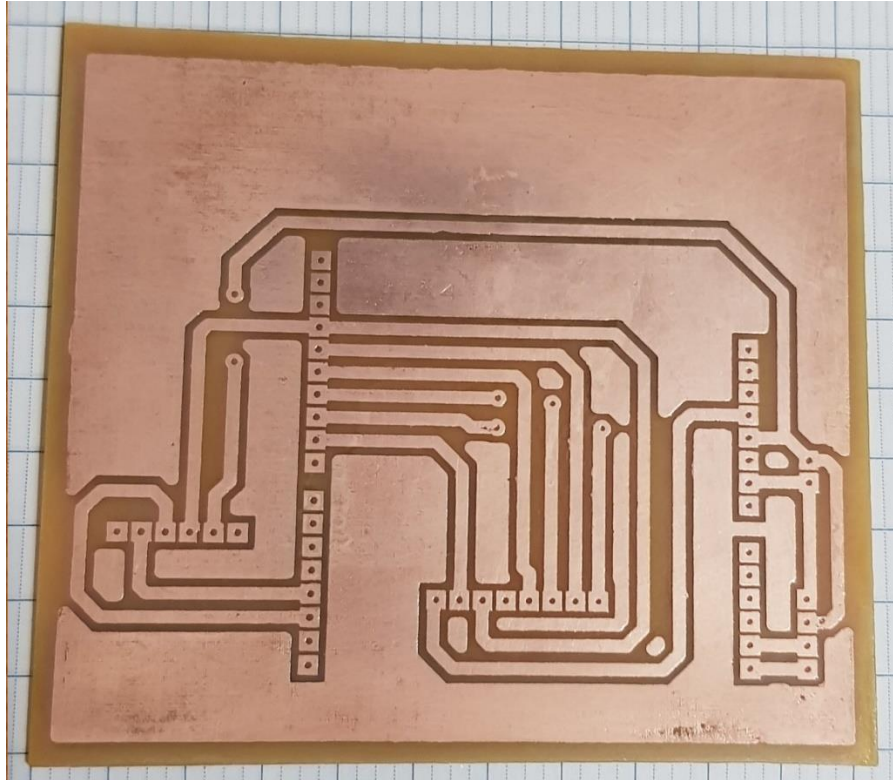
Hình 4.1. PCB của board tại trung tâm y tế



Hình 4.2. PCB của board đội bắt chó mèo thả rông



Hình 4.3. Hình mạch sau khi ủi rửa của board tại trung tâm y tế



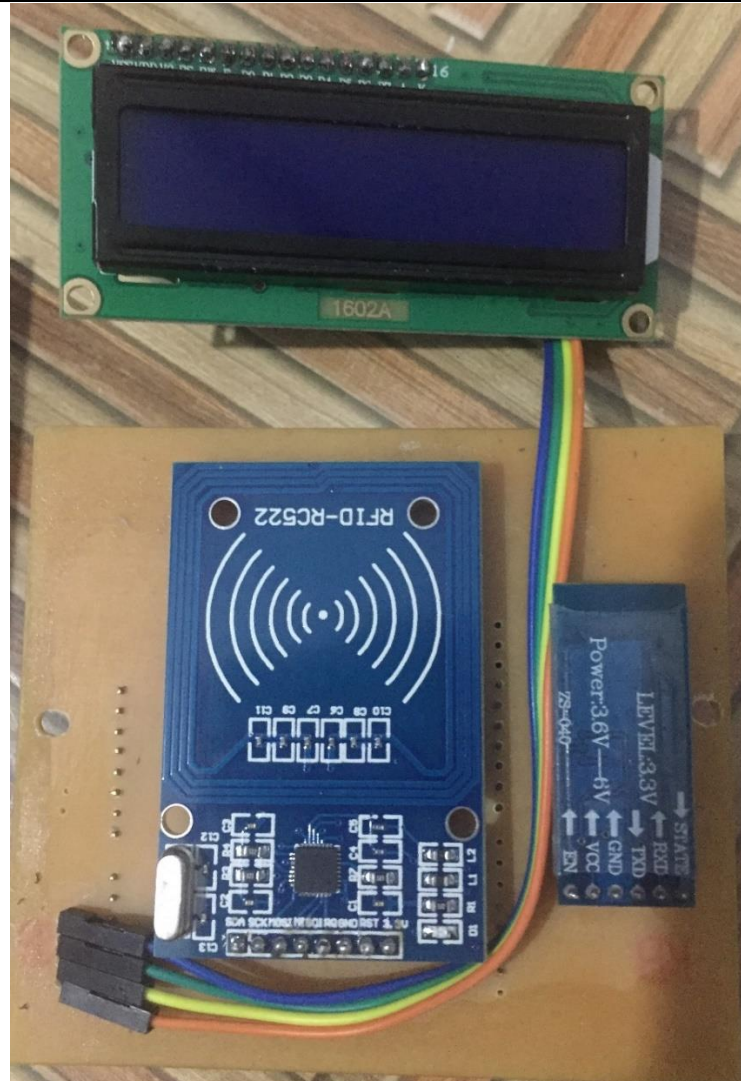
Hình 4.4. Hình mạch sau khi ủ rửa của board đội bắt chó mèo thả rộng

4.2.2 Lắp ráp và kiểm tra

Sau khi thiết kế xong PCB nhóm tiến hành đặt gia công mạch in. Nhóm tiến hành hàn các linh kiện lên bo và chạy thử để xem mạch có chạy có gì lỗi hay cần khắc phục nữa không.



Hình 4.5. Mặt trên board của trạm y tế khi lắp linh kiện



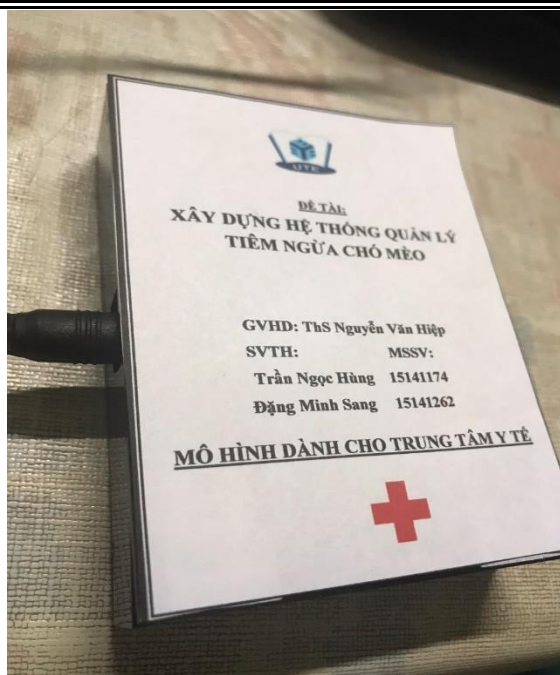
Hình 4.6. Mặt trên board của nhân viên đô thị khi lắp linh kiện

4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH

4.3.1 Đóng gói board dành cho trung tâm y tế

Nhóm sử dụng vật liệu Mica đen để tạo một cái hộp để chứa vừa board mạch, các hộp được cắt vuông cạnh và nhỏ gọn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 4.7. Hình ảnh mặt trước khi đóng hộp mica



Hình 4.8. Hình ảnh mặt sau khi đóng hộp mica

Mặt sau của hộp được cố định các đinh ốc, giúp cho board mạch ở phía trong hộp không bị dịch chuyển.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

4.3.2 Đóng gói board dành cho đội bắt chó mèo thả rông



Hình 4.9. Hình ảnh mặt sau khi đóng hộp mica



Hình 4.10. Hình ảnh mặt trước khi đóng hộp mica

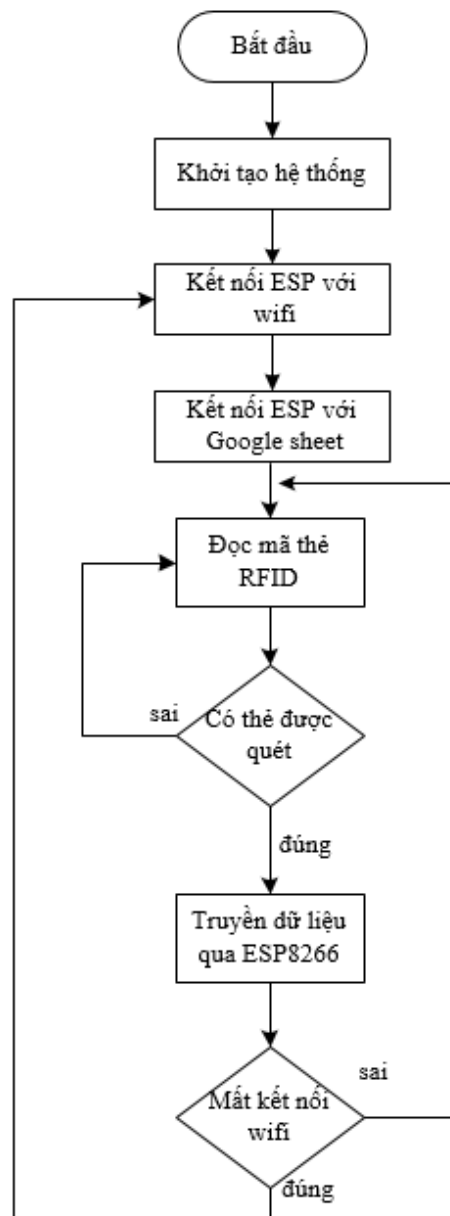
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Hộp được thiết kế có kích thước 10x15x3.5cm. Có màn hình lcd hiển thị, một công tắc để bật tắt thiết bị, ngoài ra hộp cũng có đầu cắm micro USB để sạc cho mạch khi nào hết pin.

4.4 LẬP TRÌNH HỆ THỐNG

4.4.1 Lưu đồ giải thuật

Chương trình board sử dụng trong trạm y tế



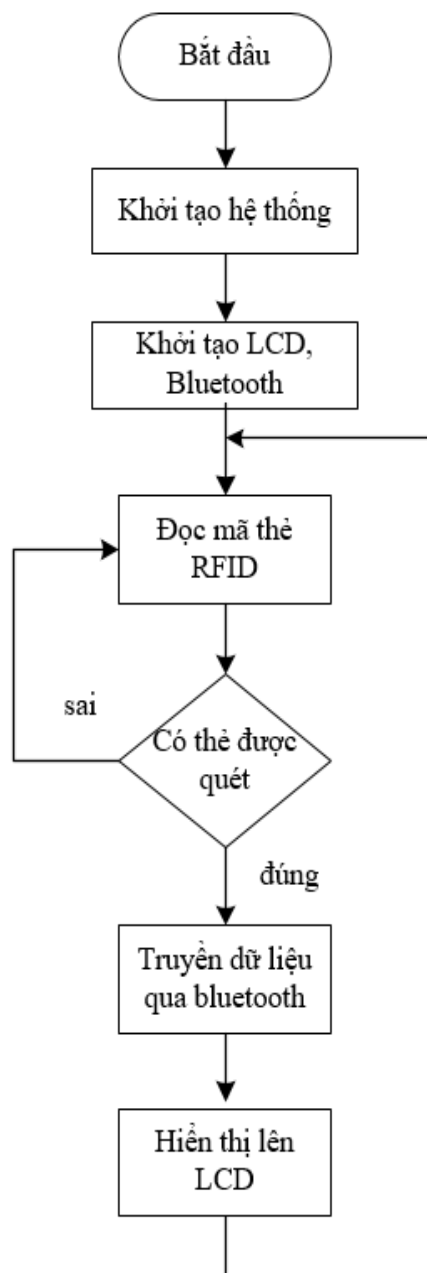
Hình 4.11. Lưu đồ chương trình board sử dụng trong trạm y tế

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Giải thích lưu đồ:

Khi cấp nguồn chương trình sẽ bắt đầu chạy, hệ thống sẽ được khởi tạo, sau đó ESP8266 sẽ bắt đầu kết nối với wifi thông qua ID và password được viết trong chương trình. Tiếp đến nó sẽ kết nối với google sheet thông qua địa chỉ dưới dạng đường link web. Một chương trình đọc mã thẻ rfid sẽ được chạy, khi có thẻ rfid được quét thì số ID đó sẽ được chuyển qua đường link theo dạng một biến thông qua mạng wifi.

Chương trình board sử dụng cho đội bắt chó mèo thả rông



Hình 4.12. Lưu đồ chương trình sử dụng cho đội bắt chó mèo thả rông

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Giải thích lưu đồ:

Khi cấp nguồn thì hệ thống bắt đầu khởi tạo, khi có thẻ rfid được quét qua thì mà đó sẽ hiển thị giá trị lên màn hình LCD, đồng thời nó cũng sẽ được chuyển qua điện thoại có cài đặt app thông qua kết nối Bluetooth.

4.4.2 Phần mềm lập trình cho vi điều khiển

Đề tài sử dụng phần mềm ArduinoIDE (Arduino Intergrated Development Environment) để lập trình cho NodeMCU; Ngôn ngữ được sử dụng ở ArduinoIDE là C/C++. Tất cả đều là mã nguồn mở, được đóng góp và hỗ trợ rất nhiều từ cộng đồng, rất thích hợp cho những ai mới bắt đầu tìm hiểu hoặc không chuyên để dễ dàng tiếp cận, nắm bắt và triển khai nhanh chóng. ArduinoIDE hoạt động được trên cả 3 nền tảng: Windows, MAC OS và Linux.

Bên dưới là hướng dẫn về cách cài đặt, cách tạo project, viết code và biên dịch chương trình trên hệ điều hành Windows. Cài đặt:

Cài đặt Java Runtime Environment (JRE)

Vì Arduino IDE được viết trên Java, nên ta cần phải cài đặt JRE trước; nếu không, Arduino IDE sẽ không hoạt động được. JRE có 2 bản phổ biến nhất hiện nay dành cho Windows 32 bit (x86) và Windows 64 bit (x64).

Trang web tải JRE:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Cài đặt Arduino IDE

Trang web tải Arduino:

<https://www.arduino.cc/en/Main/Software/>

Bước 1: Ta click vào đường dẫn ở trên, chọn “Windows ZIP file for non admin install” như hình sau:

Download the Arduino Software



Hình 4.13 Giao diện tải Arduino IDE

Tiếp tục bấm vào “JUST DOWNLOAD” để tải phần mềm Arduino IDE, bạn cũng có thể đóng góp ở ngay phím bên cạnh.

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.

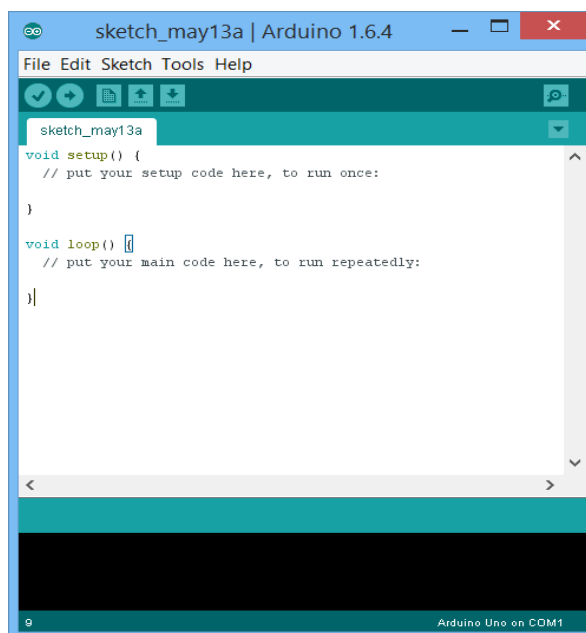


Hình 4.14 Ủng hộ nhà phát triển Arduino IDE

Bước 2: Khi đã tải xong, giải nén file vừa tải. Sau đó copy thư mục đó đến nơi lưu trữ mong muốn.

Bước 3: Chạy file arduino.exe để khởi động Arduino IDE

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 4.15 Giao diện chính của Arduino IDE

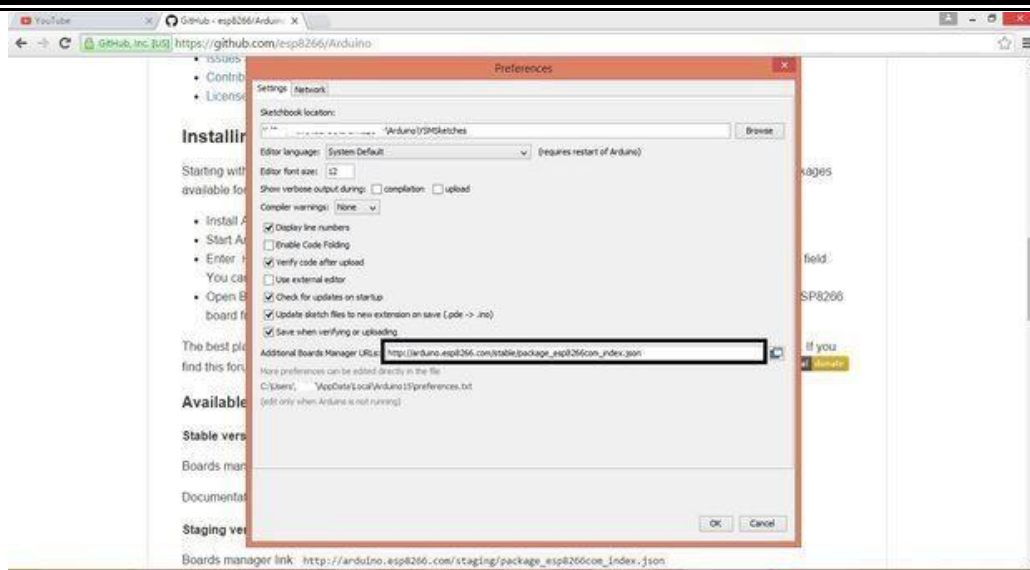
Sau khi đã cài đặt xong, mở phần mềm lên ta sẽ thấy một giao diện rất dễ nhìn và thân thiện hơn so với đa phần các phần mềm lập trình khác. Arduino IDE làm việc với dưới dạng bản soạn thảo gọi là Sketch, ta sẽ soạn các lệnh lập trình (code) vào Sketch rồi sử dụng thao tác biên tập và upload chương trình đó xuống board Arduino đã cắm vào máy. Khi tiến hành nạp code thì ta cần phải chắc chắn rằng phần mềm đã nhận được tín hiệu của board Arduino (Arduino COM port detect); bản Sketch đang soạn nạp đúng với board Arduino tương ứng (khi cần soạn hai Sketch giao tiếp giữa hai board Arduino và cắm vào cùng máy tính thì vấn đề như vậy sẽ bắt đầu phát sinh). Khi cắm board Arduino vào máy tính cổng COM sẽ được nhận và ta vào phần Tools -> Port để chọn cổng COM kết nối Arduino IDE với board. Sau khi máy đã nhận cổng COM thì ta cần điều chỉnh phần mềm lập trình Arduino xác nhận đúng loại board đang muốn nạp.

Cài đặt Driver cho NodeMCU

Trang web tải:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

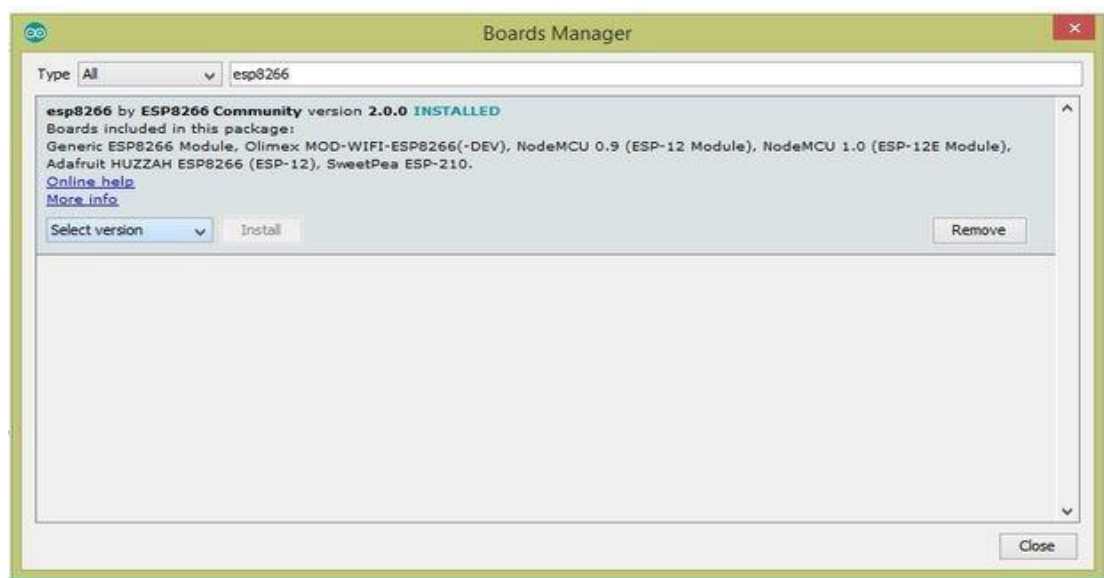
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 4.16 Cài đặt Driver cho NodeMCU 1

Bước 1: Trong màn hình chính của Arduino, ta chọn File -> Preference, copy đường link trên vào mục Additional Boards Manager URLs; click chọn OK và khởi động lại Arduino IDE.

Bước 2: Cài đặt Firmware ESP8266 cho Arduino IDE.



Hình 4.17 Cài đặt Driver cho NodeMCU 2

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

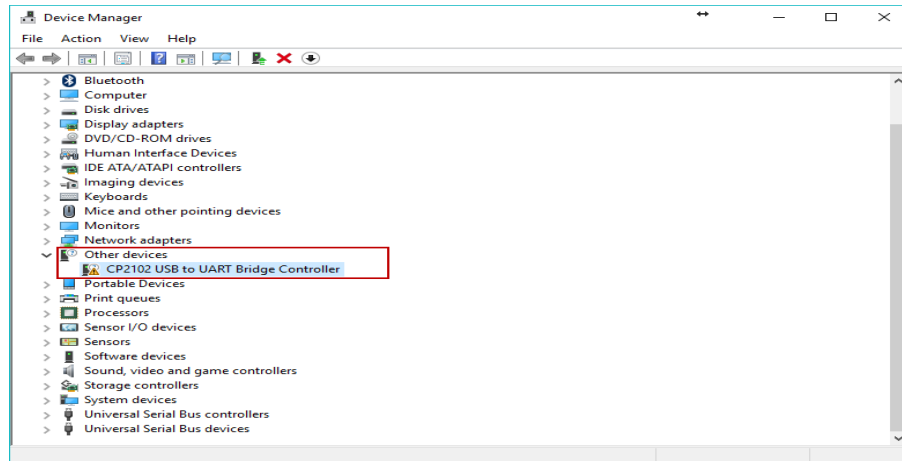
Vào Device Manager để kiểm tra xem driver đã nhận được hay chưa. Nếu trong trường hợp xuất hiện dấu chấm than như hình 4.12 bên dưới thì tức là máy tính đang sử dụng chưa có driver (đây là một trong những lỗi rất thường gặp khi máy tính không giao

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

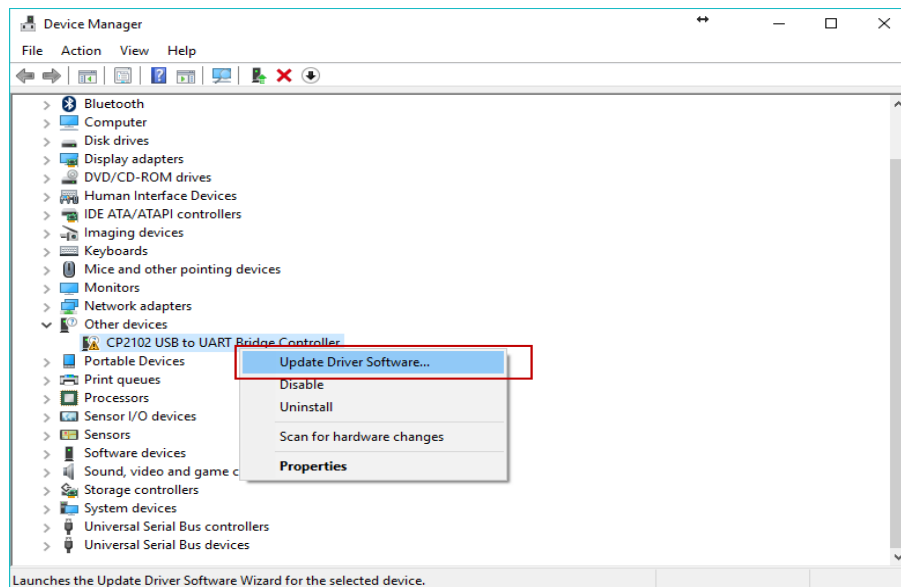
tiếp được với module Wifi). Ta cần tải driver bằng link bên dưới rồi giải nén. Tiếp theo click phải chuột để chọn Update Driver Software.

Trang web tải:

https://www.pololu.com/file/download/pololu-cp2102-windows-121204.zip?file_id=0J14



Hình 4.18 Cài đặt Driver cho NodeMCU3



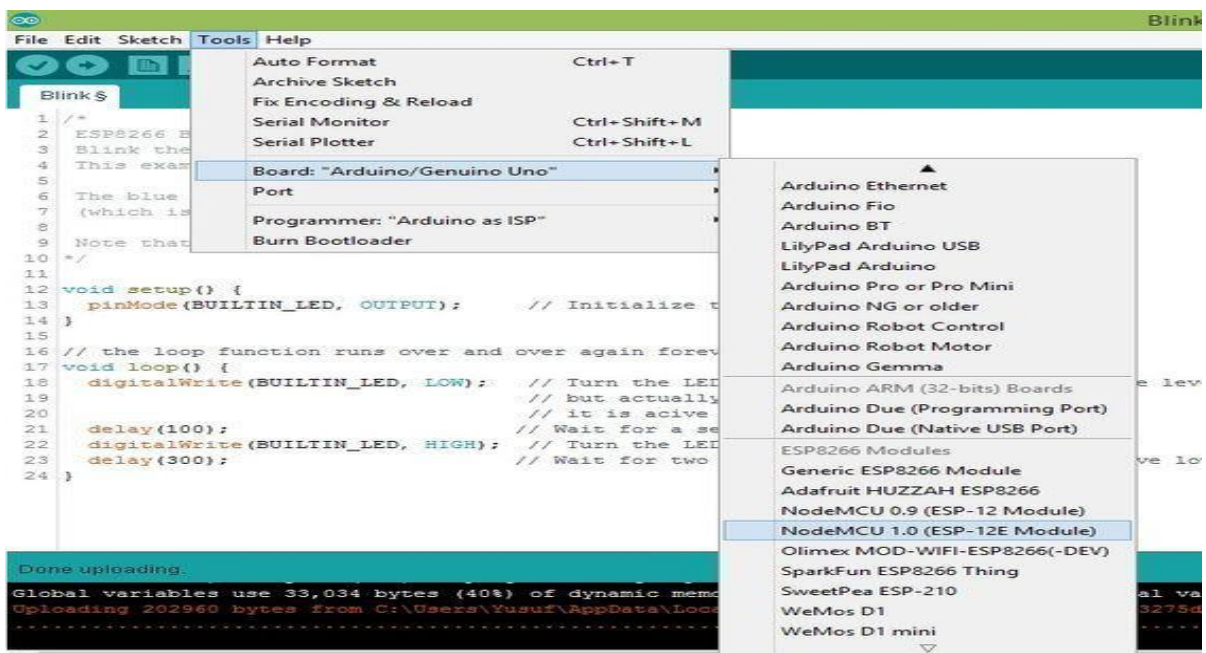
Hình 4.19 Cài đặt Driver cho NodeMCU4

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

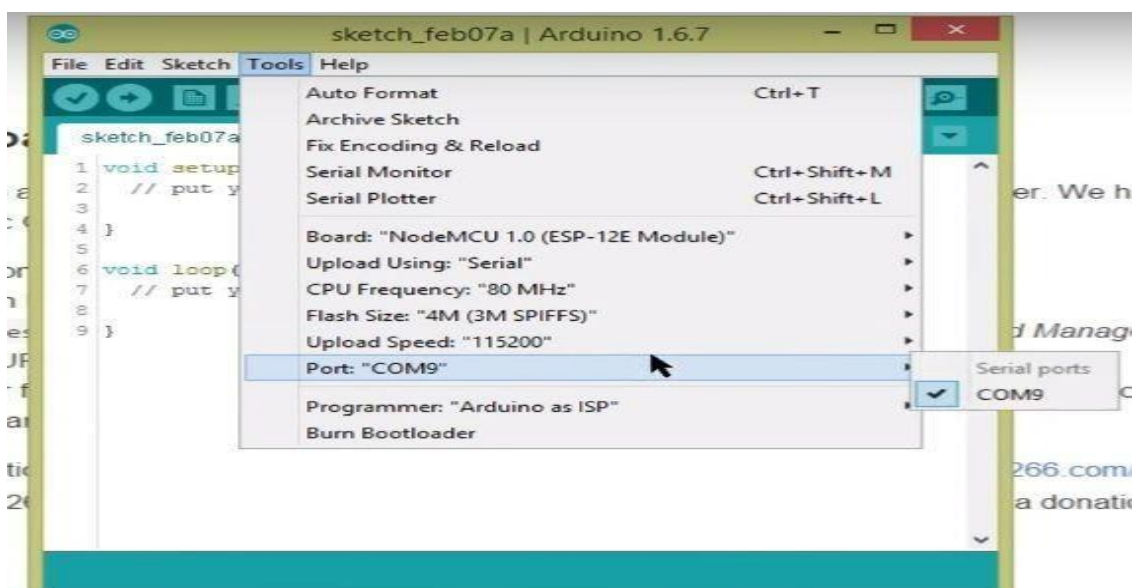
Tiếp tục chọn “Browse my computer for driver software” và trở đến thư mục có chứa driver để cài đặt ở trên. Vào Device Manager một lần nữa để kiểm tra xem driver đã cài đặt được chưa.

Chọn phần cứng để lập trình

Vào Tools -> Board -> chọn loại board cần lập trình và chọn **Port** mà board đang kết nối vào máy tính.



Hình 4.20 Chọn phần cứng để lập trình

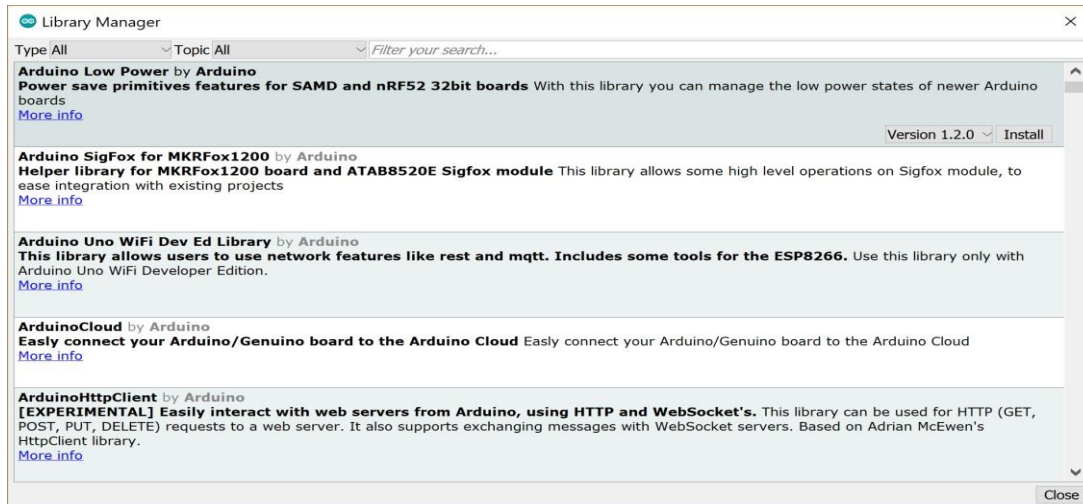


Hình 4.21 Chọn Port kết nối

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Cài đặt thư viện bổ sung cho Arduino IDE

Để giúp cho việc lập trình dễ dàng hơn, ta cài đặt bộ thư viện bổ sung bằng việc chọn Sketch -> Include Library -> Add .ZIP Library -> trở đến thư mục bổ sung định dạng .zip hoặc ta có thể cài đặt gián tiếp bằng cách vào Manage Libraries, tìm tên thư viện cần tải, bấm **Install** để cài đặt.



Hình 4.22 Cài đặt thư viện cho Arduino IDE

4.4.3 Phần mềm lập trình cho điện thoại, máy tính

Giới thiệu phần mềm Android Studio



Hình 4.23 Giao diện phần mềm Android Studio

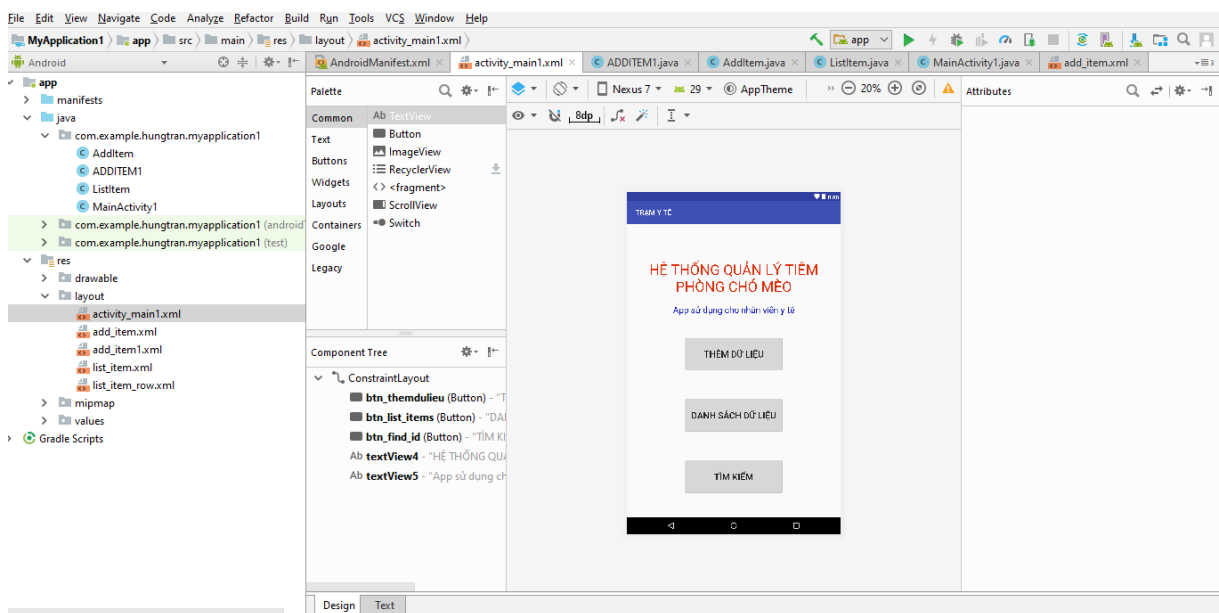
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Hiện nay, ứng dụng Android có thể được viết bằng nhiều phần mềm như Visual Studio, Android Studio, MIT AppInventor, Eclipse, ... Trong số đó thì Android Studio là phần mềm được Google hỗ trợ mạnh mẽ và nhóm chọn phần mềm này để viết ứng dụng điều khiển, hiển thị các thông số của tủ đồ.

Android Studio là một phần mềm bao gồm các công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như smartphone, tablet, ... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/ deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển ứng dụng từ đơn giản đến phức tạp.

Trong lập trình Android Studio có 3 bước quan trọng là thiết kế giao diện, ánh xạ và lập trình Java để kết nối tới các đối tượng trong layout.

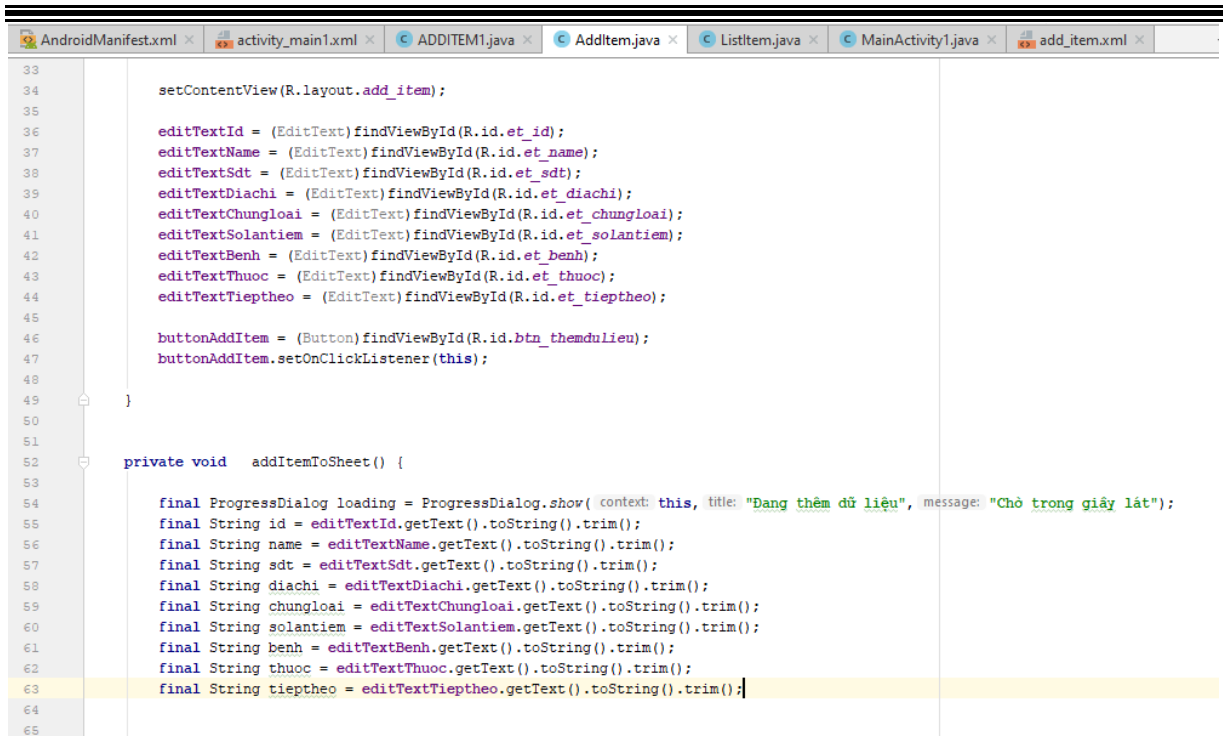
- Thiết kế giao diện: sắp xếp các đối tượng như Image, View, TextView, Button... trong các layout hỗ trợ sẵn như Linear Layout, Relative Layout, ...



Hình 4.24 Giao diện phần thiết kế giao diện cho ứng dụng

- Ánh xạ: đây là giai đoạn kết nối các đối tượng trong giao diện với các câu lệnh để điều khiển các đối tượng đó.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



```
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

setContentView(R.layout.add_item);

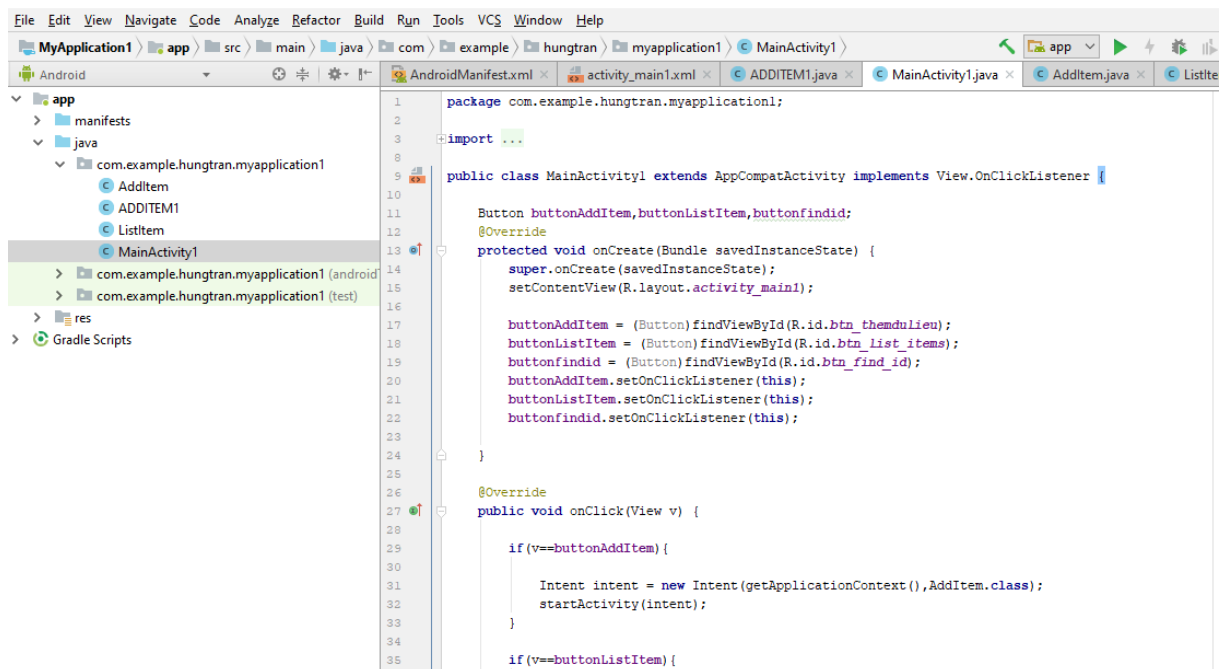
editTextId = (EditText) findViewById(R.id.et_id);
editTextName = (EditText) findViewById(R.id.et_name);
editTextSdt = (EditText) findViewById(R.id.et_sdt);
editTextDiachi = (EditText) findViewById(R.id.et_diachi);
editTextChungloai = (EditText) findViewById(R.id.et_chungloai);
editTextSolantiem = (EditText) findViewById(R.id.et_solantiem);
editTextBenh = (EditText) findViewById(R.id.et_benh);
editTextThuoc = (EditText) findViewById(R.id.et_thuoc);
editTextTieptheo = (EditText) findViewById(R.id.et_tieptheo);

buttonAddItem = (Button) findViewById(R.id.btn_themdulieu);
buttonAddItem.setOnClickListener(this);
}

private void addItemToSheet() {
    final ProgressDialog loading = ProgressDialog.show(context, this, title: "Đang thêm dữ liệu", message: "Chờ trong giây lát");
    final String id = editTextId.getText().toString().trim();
    final String name = editTextName.getText().toString().trim();
    final String sdt = editTextSdt.getText().toString().trim();
    final String diachi = editTextDiachi.getText().toString().trim();
    final String chungloai = editTextChungloai.getText().toString().trim();
    final String solantiem = editTextSolantiem.getText().toString().trim();
    final String benh = editTextBenh.getText().toString().trim();
    final String thuoc = editTextThuoc.getText().toString().trim();
    final String tieptheo = editTextTieptheo.getText().toString().trim();
}
```

Hình 4.25 Ảnh xạ các đối tượng trong giao diện với các câu lệnh

- Lập trình Java: thực hiện các lệnh viết trên nền tảng Java để tác động tới các đối tượng trong layout, gửi dữ liệu lên Firebase, kích hoạt Countdown Timer...



```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

package com.example.hungtran.myapplication1;

import ...

public class MainActivity1 extends AppCompatActivity implements View.OnClickListener {

    Button buttonAddItem,buttonListItem,buttonfindid;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main1);

        buttonAddItem = (Button) findViewById(R.id.btn_themdulieu);
        buttonListItem = (Button) findViewById(R.id.btn_list_items);
        buttonfindid = (Button) findViewById(R.id.btn_find_id);
        buttonAddItem.setOnClickListener(this);
        buttonListItem.setOnClickListener(this);
        buttonfindid.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {

        if (v==buttonAddItem) {

            Intent intent = new Intent(getApplicationContext(),AddItem.class);
            startActivity(intent);
        }

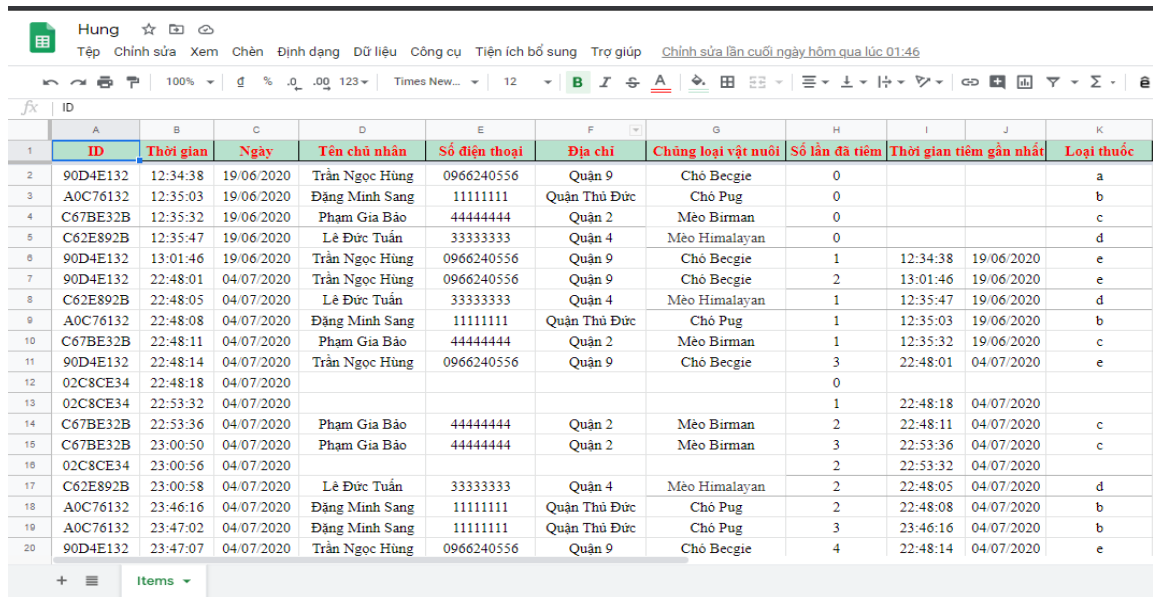
        if (v==buttonListItem) {
```

Hình 4.26 Giao diện phân lập trình Java

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

4.4.4 Hướng dẫn xây dựng code đối với google sheet

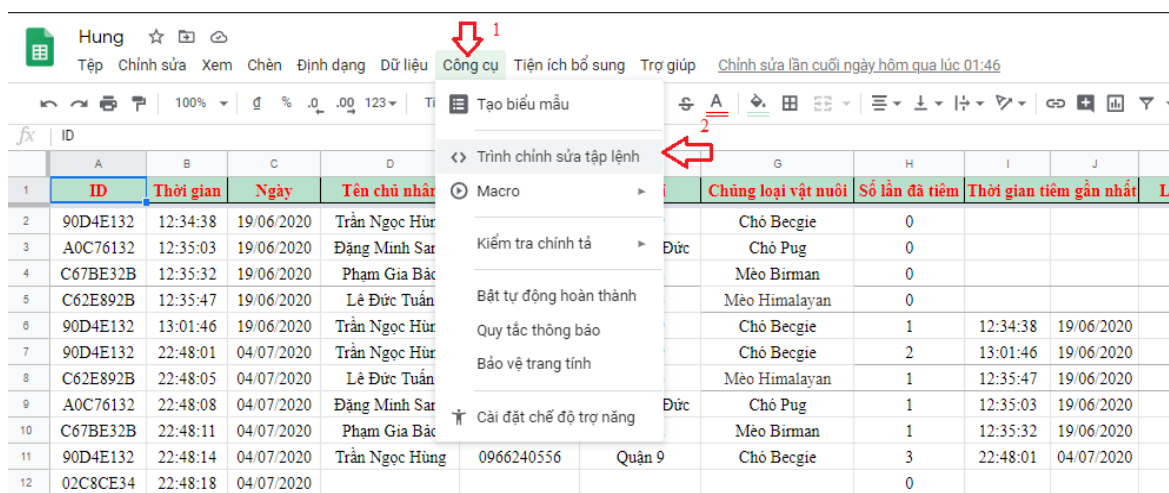
Google Sheet – một ứng dụng web nằm trong bộ G Suite của Google không xa lạ gì đối với nhân viên văn phòng hoặc người dùng internet muốn sử dụng Excel trên nền tảng web **miễn phí**. Tuy được xây dựng trên nền nhưng Google Sheet đã hỗ trợ rất nhiều tính năng cơ bản của excel nó có thể thay thế được khoảng 70-80% những tính năng của Microsoft Excel.



ID	Thời gian	Ngày	Tên chủ nhân	Số điện thoại	Địa chỉ	Chủng loại vật nuôi	Số lần đã tiêm	Thời gian tiêm gần nhất	Loại thuốc	
90D4E132	12:34:38	19/06/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beegie	0		a	
A0C76132	12:35:03	19/06/2020	Đặng Minh Sang	11111111	Quận Thủ Đức	Chó Pug	0		b	
C67BE32B	12:35:32	19/06/2020	Phạm Gia Bảo	44444444	Quận 2	Mèo Birman	0		c	
C62E892B	12:35:47	19/06/2020	Lê Đức Tuấn	33333333	Quận 4	Mèo Himalayan	0		d	
90D4E132	13:01:46	19/06/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beegie	1	12:34:38	19/06/2020	e
90D4E132	22:48:01	04/07/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beegie	2	13:01:46	19/06/2020	e
C62E892B	22:48:05	04/07/2020	Lê Đức Tuấn	33333333	Quận 4	Mèo Himalayan	1	12:35:47	19/06/2020	d
A0C76132	22:48:08	04/07/2020	Đặng Minh Sang	11111111	Quận Thủ Đức	Chó Pug	1	12:35:03	19/06/2020	b
C67BE32B	22:48:11	04/07/2020	Phạm Gia Bảo	44444444	Quận 2	Mèo Birman	1	12:35:32	19/06/2020	c
90D4E132	22:48:14	04/07/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beegie	3	22:48:01	04/07/2020	e
02C8CE34	22:48:18	04/07/2020					0			
02C8CE34	22:53:32	04/07/2020					1	22:48:18	04/07/2020	
C67BE32B	22:53:36	04/07/2020	Phạm Gia Bảo	44444444	Quận 2	Mèo Birman	2	22:48:11	04/07/2020	c
C67BE32B	23:00:50	04/07/2020	Phạm Gia Bảo	44444444	Quận 2	Mèo Birman	3	22:53:36	04/07/2020	c
02C8CE34	23:00:56	04/07/2020					2	22:53:32	04/07/2020	
C62E892B	23:00:58	04/07/2020	Lê Đức Tuấn	33333333	Quận 4	Mèo Himalayan	2	22:48:05	04/07/2020	d
A0C76132	23:46:16	04/07/2020	Đặng Minh Sang	11111111	Quận Thủ Đức	Chó Pug	2	22:48:08	04/07/2020	b
A0C76132	23:47:02	04/07/2020	Đặng Minh Sang	11111111	Quận Thủ Đức	Chó Pug	3	23:46:16	04/07/2020	b
90D4E132	23:47:07	04/07/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beegie	4	22:48:14	04/07/2020	e

Hình 4.27 Giao diện trang google sheet

Trên màn hình của Google Sheet, để có thể lập trình code cho nó có thể chạy tự động thì đầu tiên chúng ta click vào **Công cụ > Trình chỉnh sửa tập tin**. Một màn hình của Apps Script sẽ được hiện ra cho bạn viết code javascript.



Hình 4.28 Các bước mở trang code lập trình

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

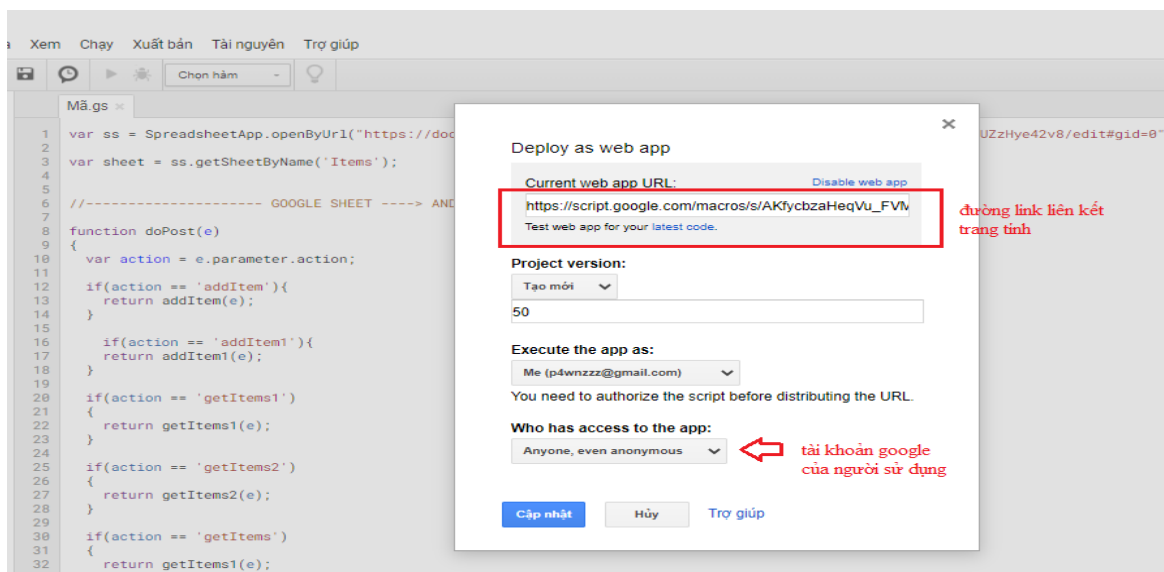
Sau đó chúng ta sẽ có một giao diện mới được mở ra để có thể viết code cho giao diện excel chính.



```
1 var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1hU0BJBGLrwmKrVWfrjIqFrtzELhi3knkpUzZhye42v8/edit#gid=0");
2
3 var sheet = ss.getSheetByName('Items');
4
5
6 //----- GOOGLE SHEET -----> ANDROID -----
7
8 function doPost(e)
9 {
10  var action = e.parameter.action;
11
12  if(action == 'addItem'){
13    return addItem(e);
14  }
15
16  if(action == 'addItem1'){
17    return addItem1(e);
18  }
19
20  if(action == 'getItems1')
21  {
22    return getItems1(e);
23  }
24
25  if(action == 'getItems2')
26  {
27    return getItems2(e);
28  }
29
30  if(action == 'getItems')
31  {
32    return getItems1(e);
33  }
34 }
```

Hình 4.29 Giao diện trang code lập trình

Ở đây sẽ sử dụng các hàm của ngôn ngữ java để viết chương trình. Sau khi viết xong chương trình code thì chúng ta cần phải cấp quyền cho google script sử dụng tài khoản google của bạn để chạy chương trình.



Hình 4.30 Các bước cho phép google script liên kết

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Sau khi cho phép google script chạy xong thì chúng ta sẽ lấy phần đường link kết nối để lập trình trên android studio cũng như ESP8266 để chúng có thể kết nối với google sheet thông qua đường link đó.

4.5 VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC

Hướng dẫn sử dụng sản phẩm

Bước 1: Cấp nguồn cho module sử dụng trong trạm y tế thông qua adapter 5v 1A. Còn module sử dụng cho nhân viên đô thị thì có một công tắc để đóng tắt nguồn cho moduel.

Bước 2:

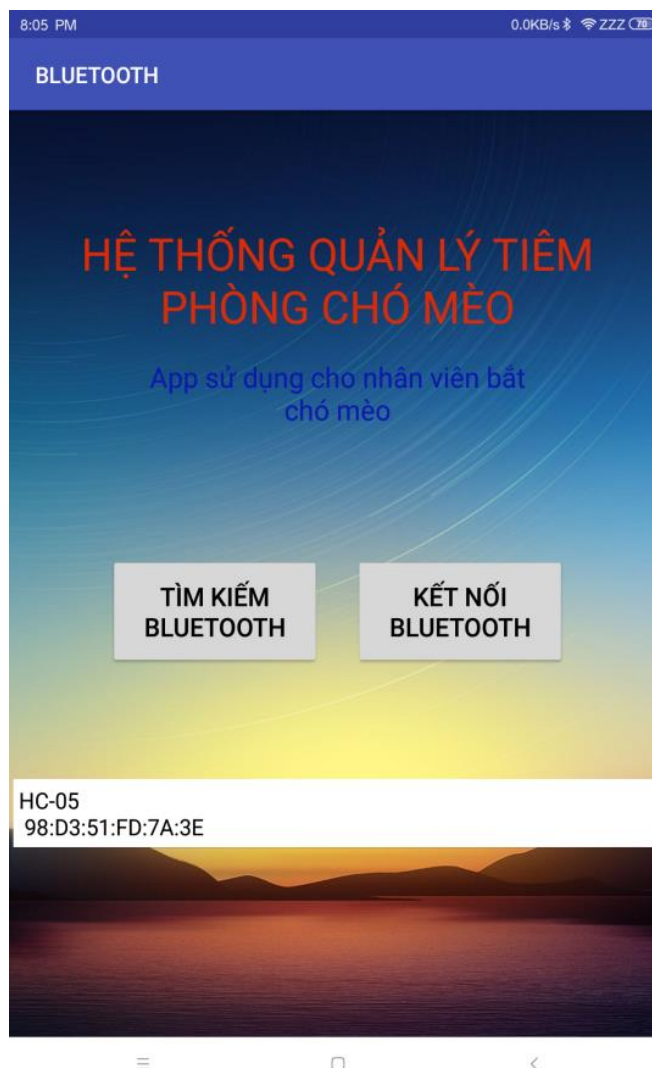
- Đối với điện thoại thông minh sử dụng hệ điều hành android của những nhân viên trạm y tế hay khách hàng thì chỉ cần có mạng 3G, 4G hoặc wifi. Sau đó mở app lên và nó sẽ có giao diện hiển thị đúng những gì mình muốn sử dụng.



Hình 4.31 Giao diện app sử dụng

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

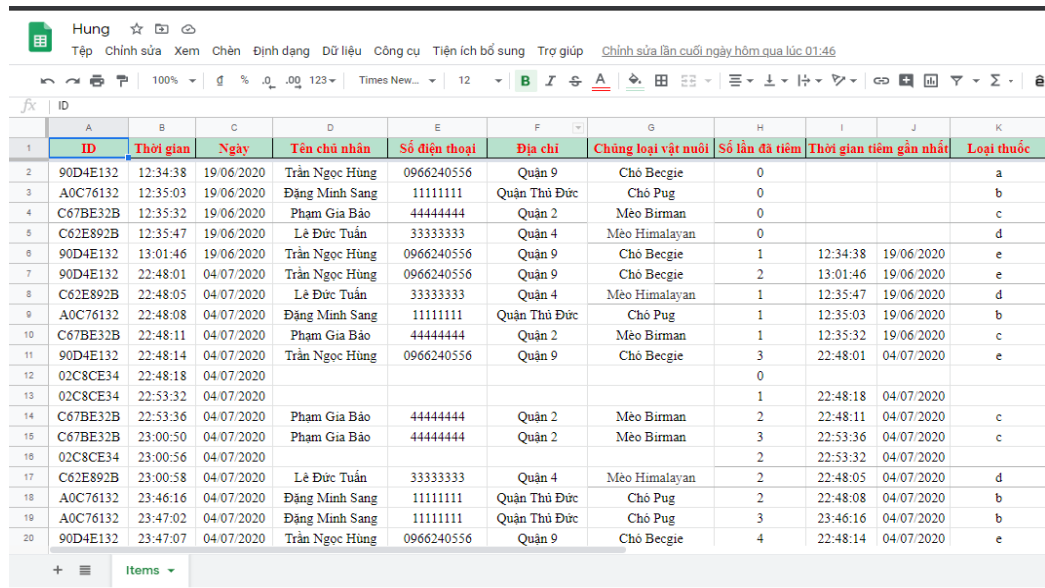
- Với những nhân viên đô thị đi bắt những con chó, mèo bị lạc ngoài đường thì sẽ có một ứng dụng riêng. Đầu tiên chúng ta cần nhấn vào **tìm kiếm** để xem xung quanh có các thiết bị Bluetooth nào, tiếp đến chọn thiết bị cần rồi nhấn vào **kết nối**. khi đó sẽ có màn hình mới hiện lên, khi chúng ta quét thẻ qua module thì UID sẽ được hiện thị lên app, chúng ta chỉ cần nhấn **tìm kiếm** để hiển thị những thông tin cần thiết.



Hình 4.32 Giao diện hiển thị của app dành cho nhân viên

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Đối với trang giao diện google sheet thì chỉ cần chúng ta quét thẻ thông qua module ở trạm y tế thì các thông tin sẽ được hiển thị lên màn hình máy tính.



The screenshot shows a Google Sheet spreadsheet with the following data:

ID	Thời gian	Ngày	Tên chủ nhân	Số điện thoại	Địa chỉ	Chủng loại vật nuôi	Số lần đã tiêm	Thời gian tiêm gần nhất	Loại thuốc	
90D4E132	12:34:38	19/06/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beagle	0		a	
A0C76132	12:35:03	19/06/2020	Đặng Minh Sang	11111111	Quận Thủ Đức	Chó Pug	0		b	
C67BE32B	12:35:32	19/06/2020	Phạm Gia Bảo	44444444	Quận 2	Mèo Birman	0		c	
C62E892B	12:35:47	19/06/2020	Lê Đức Tuấn	33333333	Quận 4	Mèo Himalayan	0		d	
90D4E132	13:01:46	19/06/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beagle	1	12:34:38	19/06/2020	e
90D4E132	22:48:01	04/07/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beagle	2	13:01:46	19/06/2020	e
C62E892B	22:48:05	04/07/2020	Lê Đức Tuấn	33333333	Quận 4	Mèo Himalayan	1	12:35:47	19/06/2020	d
A0C76132	22:48:08	04/07/2020	Đặng Minh Sang	11111111	Quận Thủ Đức	Chó Pug	1	12:35:03	19/06/2020	b
C67BE32B	22:48:11	04/07/2020	Phạm Gia Bảo	44444444	Quận 2	Mèo Birman	1	12:35:32	19/06/2020	c
90D4E132	22:48:14	04/07/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beagle	3	22:48:01	04/07/2020	e
02C8CE34	22:48:18	04/07/2020					0			
02C8CE34	22:53:32	04/07/2020					1	22:48:18	04/07/2020	
C67BE32B	22:53:36	04/07/2020	Phạm Gia Bảo	44444444	Quận 2	Mèo Birman	2	22:48:11	04/07/2020	c
C67BE32B	23:00:50	04/07/2020	Phạm Gia Bảo	44444444	Quận 2	Mèo Birman	3	22:53:36	04/07/2020	c
02C8CE34	23:00:56	04/07/2020					2	22:53:32	04/07/2020	
C62E892B	23:00:58	04/07/2020	Lê Đức Tuấn	33333333	Quận 4	Mèo Himalayan	2	22:48:05	04/07/2020	d
A0C76132	23:46:16	04/07/2020	Đặng Minh Sang	11111111	Quận Thủ Đức	Chó Pug	2	22:48:08	04/07/2020	b
A0C76132	23:47:02	04/07/2020	Đặng Minh Sang	11111111	Quận Thủ Đức	Chó Pug	3	23:46:16	04/07/2020	b
90D4E132	23:47:07	04/07/2020	Trần Ngọc Hùng	0966240556	Quận 9	Chó Beagle	4	22:48:14	04/07/2020	e

Hình 4.33 Giao diện hiển thị thông tin

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

5.1 GIỚI THIỆU

Chương này trình bày kết quả của cả quá trình nghiên cứu làm đề tài trong thời gian 15 tuần. Bên cạnh đó là nhận xét, đánh giá và đề xuất hướng phát triển của sản phẩm mô hình để hoàn thiện và có thể đi vào thực tế.

5.2 KẾT QUẢ ĐẠT ĐƯỢC

Các chuẩn truyền dữ liệu

- Tìm hiểu và nắm được các kiến thức cần thiết của các chuẩn truyền dữ liệu UART, SPI, I2C.

Về phần cứng

- Biết cách sử dụng phần mềm altium để thiết kế hộp chứa board mạch.
- Học được cách khảo sát các khối ngoại vi, thông số kỹ thuật, cách giao tiếp chúng với module giao tiếp wifi ESP NodeMCU.
- Nâng cao được kỹ năng thi công mạch (hàn linh kiện, kiểm tra các thành phần trong mạch).
- Biết cách tính toán các giá trị điện áp dòng điện trong hệ thống để chọn nguồn phù hợp.
- Biết kết nối các linh kiện với nhau sao cho phù hợp tạo thành một hệ thống hoàn chỉnh.

Phần mềm lập trình

- Biết lập trình cho arduino, ESP8266 sử dụng phần mềm adduino IDE.
- Biết sử dụng phần mềm android studio viết được app mobile chạy trên điện thoại android.

Lập trình google sheet

- Kết nối google sheet với điện thoại hay ESP8266 thông qua đường link truy cập.
- Xây dựng giao diện google sheet sao cho hợp lý.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

5.3 KẾT QUẢ THỰC NGHIỆM

5.3.1 Thiết kế thi công phần cứng



Hình 5.1. Hình ảnh thực tế mặt trước sau khi hoàn thiện

Theo như hình 5.1 , khi vừa bật thiết bị sẽ hiển thị “BAT DAU QUET THE” để người dùng có thể tiến hành quét thẻ kiểm tra. Với mỗi thẻ sẽ có 1 mã UID khác nhau tượng trưng cho mỗi con chó mèo khác nhau, và sẽ hiển thị mã đó lên màn hình của thiết bị.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

5.3.2 Kết quả phần mềm



Hình 5.2. Giao diện chính

Theo như hình 5.2 ta thấy giao diện chính của màn hình sẽ có 3 mục chính bao gồm thêm dữ liệu từ điện thoại lên trang tính, danh sách dữ liệu và tìm kiếm thông tin.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

TRẠM Y TẾ

ID : _____

Tên : _____

SĐT : _____

Địa Chỉ : _____

Giống loại : _____

Số lần tiêm : _____

Loại thuốc : _____

Bệnh : _____

Thời gian tiêm tiếp theo : _____

THÊM DỮ LIỆU

Hình 5.3 Giao diện của mục thêm dữ liệu

TRẠM Y TẾ

Search : Nhập SĐT

ID : 90D4E132
Tên : Trần Ngọc Hùng
SĐT : 0966240556
Địa Chỉ : Quận 9
Giống loại : Chó Becgie
Số lần tiêm : 0
Ngày tiêm gần nhất :
Loại thuốc : a
Bệnh : Care virus.
Thời gian tiêm tiếp theo : 19/10/2020

ID : A0C76132
Tên : Đặng Minh Sang
SĐT : 11111111
Địa Chỉ : Quận Thủ Đức
Giống loại : Chó Pug
Số lần tiêm : 0
Ngày tiêm gần nhất :

TRẠM Y TẾ

TRA CỨU THÔNG TIN

Nhập số điện thoại của khách hàng

TÌM KIẾM

Hình 5.4 Giao diện của danh sách dữ liệu và tìm kiếm

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Danh sách này sẽ được tải từ trên trang tính google sheet về ứng dụng trên điện thoại. còn khi nhập tra cứu thông tin thì cũng sẽ có một danh sách được tải về theo số điện thoại được nhập vào.



Hình 5.5 Giao diện của hệ thống dành cho đội bắt chó mèo thả rong

5.4 NHẬN XÉT, ĐÁNH GIÁ

5.4.1 Nhận xét

Sau thời gian 15 tuần nghiên cứu và thực hiện đề tài, hệ thống cơ bản đáp ứng được yêu cầu thiết kế ban đầu. Dưới đây là một số nhận xét:

Phản ứng, giao diện điều khiển:

- Mạch điều khiển trung tâm hoạt động ổn định, chính xác.
- Tốc độ quét và đưa thông tin lên trang tính google sheet của ESP8266 chưa được cao.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Quá trình quét mã và gửi qua điện thoại thông qua Bluetooth nhanh.
- Độ ổn định của thông tin được truyền khá ổn.
- Thời gian hoạt động khi chạy liên tục của board sử dụng cho các nhân viên đô thị là khoảng 3 tiếng.
- Dễ sử dụng, vận chuyển và cất giữ.

App android:

- Giao diện trực quan, đơn giản và dễ sử dụng.
- Hiển thị đầy đủ thông tin cần thiết cho người sử dụng biết và theo dõi.
- Đôi lúc có độ trễ nhất định khi tải các thông tin từ google sheet xuống điện thoại. Tuy nhiên đây là yếu tố khách quan do mạng wifi, việc này không ảnh hưởng nhiều đến người sử dụng.
- Hiện tại app chỉ chạy được trên nền tảng hệ điều hành android.
- Chưa thực hiện được việc gửi thông báo về thời gian cho các chủ vật nuôi.

Trang tính google sheet:

- Bố trí giao diện tương đối ổn định và dễ quan sát.
- Các dữ liệu tải lên đầy vào hàng không đồng bộ.
- Tốc độ hoạt động phụ thuộc vào wifi.

5.4.2 Đánh giá

Bảng 5.1: Số liệu thực nghiệm

Quá trình	Số lần thực nghiệm	Số lần thành công	Đánh giá	Tốc độ
Quét thẻ từ board trạm y tế	20	19	ĐẠT	1 giây
Quét thẻ từ board của nhân viên đô thị	20	20	ĐẠT	1 giây
Truyền thông tin giữa app và google sheet	20	18	ĐẠT	4 giây

Sau quá trình vận hành thử hệ thống, nhóm có những đánh giá sau đây: Hệ thống hoạt động ổn định đạt được khoảng 70% những mục tiêu ban đầu đề ra. Các board mạch được đựng trong các hộp nhỏ gọn và khá chắc chắn, đẹp mắt.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 KẾT LUẬN

Sau khoảng 15 tuần nghiên cứu và tìm hiểu, nhóm đã hoàn thành đồ án và thi công ra sản phẩm hệ thống quản lý tiêm phòng cho chó mèo đáp ứng được những yêu cầu đã đặt ra ban đầu:

- Sản phẩm được đóng hộp nhỏ gọn.
- Tốc độ truyền dữ liệu tương đối cao và chính xác.
- Các chức năng cơ bản của app hoạt động tốt.
- Có hệ thống pin để sử dụng khi đi ra ngoài.
- App android có giao diện trực quan, dễ theo dõi.
- Thời gian trễ khi giao tiếp giữa trang tính và App khá lớn khoảng 3 đến 4 giây.
- Thời gian duy trì hoạt động của Pin chưa cao.

6.2 HƯỚNG PHÁT TRIỂN

- Thiết kế board với mẫu mã đẹp mắt hơn.
- Tích hợp thêm khả năng ghi các thông tin vào thẻ.
- Gửi các thông tin về ngày tiêm phòng về cho chủ vật nuôi thông qua app.
- Bảo mật thông tin của chủ vật nuôi thông qua hệ thống đăng nhập tài khoản để sử dụng app cho chủ vật nuôi.
- Nâng cấp giao diện cũng như chức năng của App Android.
- Phát triển ứng dụng chạy trên nhiều thiết bị có kích thước màn hình khác nhau.
- Tăng dung lượng Pin để board mạch hoạt động nhiều giờ hơn khi làm việc.

TÀI LIỆU THAM KHẢO

- [1] Quốc Huy, Hướng dẫn cách sử dụng Google Sheet, Hà Nội, 2018.
- [2] “Dân điện tử, “Thiết Kế App Cho Android Và Ios Để Điều Khiển Esp8266 Sử Dụng Google Firebase, 3 10 2019
<http://dandientu.com/esp8266/thiet-ke-app-cho-android-va-ios-de-dieu-khien-esp8266-su-dung-google-firebase/> [3 10 2019]
- [3] Nguyễn Văn Hiệp, Đinh Quang Hiệp, “Giáo trình: lập trình Android Cơ Bản”, NXB Đại Học Quốc Gia 2015.

Website tham khảo

- [1] www.arduino.vn
- [2] www.wikipedia.org
- [3] www.alldatasheet.com

PHỤ LỤC

Code arduino

```
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#include <SoftwareSerial.h>

SoftwareSerial blue(2,3);

#define SS_PIN 10
#define RST_PIN 9

LiquidCrystal_I2C lcd(0x27,16,2);

MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of the class

int code[] = {32,154,149,117}; //This is the stored UID (Unlock Card)

int codeRead = 0;

String uidString;

int readsuccess;
byte readcard[4];
char str[32] = "";
String StrUID;

void setup() {

  Serial.begin(9600);
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522
  Serial.println("Arduino RFID reading UID");

  blue.begin(9600);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("BAT DAU");
  lcd.setCursor(0,1);
  lcd.print("          QUET THE");

}

void loop()
```

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

```
{
  readsuccess = getid();

  if(readsuccess)
  {
    lcd.clear();
    Serial.print("Ma UID:");
    lcd.setCursor(0,0);
    lcd.print("Ma UID:");

    lcd.setCursor(0,1);
    lcd.print("  " + StrUID);

    //Serial.print("#");
    Serial.print(StrUID);
    //Serial.print("~");
    blue.println(StrUID);
    delay(100);
  }
}

int getid() {
  if(!mfrc522.PICC_IsNewCardPresent()) {
    return 0;
  }
  if(!mfrc522.PICC_ReadCardSerial()) {
    return 0;
  }

  for(int i=0;i<4;i++){
    readcard[i]=mfrc522.uid.uidByte[i];
    array_to_string(readcard, 4, str);
    StrUID = str;
  }
  mfrc522.PICC_HaltA();
  return 1;
}

void array_to_string(byte array[], unsigned int len, char buffer[]) {
  for (unsigned int i = 0; i < len; i++)
  {
    byte nib1 = (array[i] >> 4) & 0x0F;
    byte nib2 = (array[i] >> 0) & 0x0F;
    buffer[i*2+0] = nib1 < 0xA ? '0' + nib1 : 'A' + nib1 - 0xA;
    buffer[i*2+1] = nib2 < 0xA ? '0' + nib2 : 'A' + nib2 - 0xA;
  }
  buffer[len*2] = '\0';
}
```

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Code ESP8266

```
#include<ESP8266WiFi.h>
#include "HTTPSRedirect.h"
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN D4
#define RST_PIN D2

MFRC522 mfrc522(SS_PIN, RST_PIN);

int readsuccess;
byte readcard[4];
char str[32] = "";
String StrUID;

//Khai báo thông tin về Wifi
const char* ssid = "ZZZ";
const char* password = "tamchinmuoi";

const char *GScriptID =
"AKfycbzaHeqVu_FVMa6B_vrsoLoCm1R5MJDELLJ7x4MB1xNpKdiwKAXL";
const char* host = "script.google.com";
const char *googleRedirectHost = "script.googleusercontent.com";
const int httpsPort = 443;

HTTPSRedirect client(httpsPort);

//Khai báo URL chứa thông tin gửi dữ liệu
String url = String("/macros/s/") + GScriptID + "/exec?";

void setup()
{
    Serial.begin(9600);

    Serial.println("Connecting to wifi: ");
    Serial.println(ssid);
    Serial.flush();
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }

    SPI.begin(); // Init SPI bus
    mfrc522.PCD_Init();

    Serial.print(" IP Address: ");
```

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

```
Serial.println(WiFi.localIP());

client.setInsecure();
  client.setPrintResponseBody(false);
client.setContentTypeHeader("application/json");
Serial.print(String("Kết nối tới "));
Serial.println(host);

}

void postData(String ID, String action){
  if(!client.connected()){
    Serial.println("Kết nối tới server...");
    client.connect(host, httpsPort);
  }

  String urlFinal = url + "ID=" + ID+"&action=" + action ;

  client.POST(urlFinal, host, "");
  Serial.println("UID = " + ID + " addItem" );
}

void loop()
{
  readsuccess = getid();

  if(readsuccess)
  {
    Serial.println("");
    postData(StrUID,"addItem");
  }
}

int getid() {
  if(!mfr522.PICC_IsNewCardPresent()) {
    return 0;
  }
  if(!mfr522.PICC_ReadCardSerial()) {
    return 0;
  }

  for(int i=0;i<4;i++){
    readcard[i]=mfr522.uid.uidByte[i];
    array_to_string(readcard, 4, str);
    StrUID = str;
  }
  mfr522.PICC_HaltA();
  return 1;
}

void array_to_string(byte array[], unsigned int len, char buffer[]) {
  for (unsigned int i = 0; i < len; i++)
  {
    byte nib1 = (array[i] >> 4) & 0x0F;
```

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

```
        byte nib2 = (array[i] >> 0) & 0x0F;
        buffer[i*2+0] = nib1 < 0xA ? '0' + nib1 : 'A' + nib1 - 0xA;
        buffer[i*2+1] = nib2 < 0xA ? '0' + nib2 : 'A' + nib2 - 0xA;
    }
    buffer[len*2] = '\\0';
}
```

Code google sheet

```
var ss =
SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1hUOJBGLr
wmKrVWfrjIqFrtzELhi3kNkpUZzHye42v8/edit#gid=0");

var sheet = ss.getSheetByName('Items');

//----- GOOGLE SHEET ----> ANDROID -----
-----

function doPost(e)
{
    var action = e.parameter.action;

    if(action == 'addItem'){
        return addItem(e);
    }

    if(action == 'addItem1'){
        return addItem1(e);
    }

    if(action == 'getItem1')
    {
        return getItem1(e);
    }

    if(action == 'getItem2')
    {
        return getItem2(e);
    }

    if(action == 'getItem')
    {
        return getItem1(e);
    }
}

function addItem1(e)
{
```

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

```
var id = e.parameter.id;
var name = e.parameter.name;
var sdt = e.parameter.sdt;
var diachi = e.parameter.diachi;
var chungloai = e.parameter.chungloai;
var solantiem = e.parameter.solantiem;
var benh = e.parameter.benh;
var thuoc = e.parameter.thuoc;
var tieptheo = e.parameter.tieptheo;

var day = Utilities.formatDate(new Date(), "GMT+7", "dd/MM/yyyy");
var time = Utilities.formatDate(new Date(), "GMT+7", "HH:mm:ss");

sheet.appendRow([id,time,day,name,sdt,diachi,chungloai,solantiem,"","",benh
,thuoc,tieptheo]);
return
ContentService.createTextOutput("Success").setMimeType(ContentService.MimeT
ype.TEXT);
}

function addItem(e)
{
var x =0;
var ID = e.parameter.ID;

var day = Utilities.formatDate(new Date(), "GMT+7", "dd/MM/yyyy");
var time = Utilities.formatDate(new Date(), "GMT+7", "HH:mm:ss");

var row = sheet.getLastRow() + 1;

sheet.appendRow([ID,time,day]);

for(var i=2; i<= row; i++)
{
var myValue = sheet.getRange(i , 1).getValue();
if( ( i != row) && ( myValue == ID))
{ x += 1;
var time1 = sheet.getRange(i , 2);
var date1 = sheet.getRange(i , 3);
var ten1 = sheet.getRange(i , 4);
var sdt1 = sheet.getRange(i , 5);
var dc1 = sheet.getRange(i , 6);
var chung1 = sheet.getRange(i , 7);
var thuoc1 = sheet.getRange(i , 11);
var benh1 = sheet.getRange(i , 12);
}

if ( ( i == row) && (myValue == ID))
{ var hthi =
SpreadsheetApp.getActiveSpreadsheet().getActiveSheet().getRange("H" + row);
hthi.setValue(x);
```

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

```
var timeT1 = sheet.getRange(row , 9);
var dateT1 = sheet.getRange(row , 10);
time1.copyTo(timeT1);
date1.copyTo(dateT1);

var tenT1 = sheet.getRange(row , 4);
var sdtT1 = sheet.getRange(row , 5);
ten1.copyTo(tenT1);
sdt1.copyTo(sdtT1);

var dcT1 = sheet.getRange(row , 6);
var chungT1 = sheet.getRange(row , 7);
dc1.copyTo(dcT1);
chung1.copyTo(chungT1);

var thuocT1 = sheet.getRange(row , 11);
varbenhT1 = sheet.getRange(row , 12);
thuoc1.copyTo(thuocT1);
benh1.copyTo(benhT1);

    }
}
}

///----- ANDROID -----> GOOGLE SHEET -----
-----

function doGet(e)
{
var action = e.parameter.action;

    if(action == 'getItems')
    {
        return getItems(e);
    }
}

function getItems(e)
{
    var UID = e.parameter.itemName;
    sheet.appendRow([UID]);

    var records={};

    var rows = sheet.getRange(2, 1, sheet.getLastRow() -
1, sheet.getLastColumn()).getValues();
    var data = [];

    for (var r = 0, l = rows.length; r < l; r++)
```


CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

```
{
    var row      = rows[r],
        record  = {};
    record['a'] = row[0];
    record['d']=row[3];
    record['e']=row[4];
    record['f']=row[5];
    record['g']=row[6];
    record['h']=row[7];
    record['j']=row[9];
    record['k']=row[10];
    record['l']=row[11];
    record['m']=row[12];

    data.push(record);

}
records.items = data;
var result=JSON.stringify(records);
return
ContentService.createTextOutput(result).setMimeType(ContentService.MimeType
.JSON);
}

//-----
-----

function getItems1(e)
{
    var UID = e.parameter.itemName;
    var i = 0;
    var records={};

    var rows = sheet.getRange(2, 1, sheet.getLastRow() -
1,sheet.getLastColumn()).getValues();
    var data = [];

    for (var r = 0, l = rows.length; r < l; r++)
    {
        var row      = rows[r],
            record  = {};

        if(UID==row[0])
        { i++;
        record['a']=row[0];
        record['d']=row[3];
        record['e']=row[4];
        record['f']=row[5];
        record['g']=row[6];
        record['h']=row[7];
        record['j']=row[9];
        record['k']=row[10];
```

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

```
record['l']=row[11];
record['m']=row[12];

data.push(record);
}

}
if (i== 0)
{ record['a']= "Không có dữ liệu";
  record['d']= "Không có dữ liệu";
  record['e']= "Không có dữ liệu";
  record['f']= "Không có dữ liệu";
  record['g']= "Không có dữ liệu";
  record['h']= "Không có dữ liệu";
  record['j']= "Không có dữ liệu";
  record['k']= "Không có dữ liệu";
  record['l']= "Không có dữ liệu";
  record['m']= "Không có dữ liệu";
  data.push(record);
}
records.items = data;
var result=JSON.stringify(records);
return
ContentService.createTextOutput(result).setMimeType(ContentService.MimeType
.JSON);
}

//-----
-----

function getItems2(e)
{
  var SDT = e.parameter.itemName;
  var i = 0;
  var records={};

  var rows = sheet.getRange(2, 1, sheet.getLastRow() -
1,sheet.getLastColumn()).getValues();
  var data = [];

  for (var r = 0, l = rows.length; r < l; r++)
  {
    var row      = rows[r],
        record   = {};

    if(SDT==row[4])
    { i++;
      record['a']=row[0];
      record['d']=row[3];
      record['e']=row[4];
      record['f']=row[5];
      record['g']=row[6];
```

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

```
record['h']=row[7];
record['j']=row[9];
    record['k']=row[10];
record['l']=row[11];
record['m']=row[12];

data.push(record);
}

}
if (i== 0)
{ record['a']= "Không có dữ liệu";
  record['d']= "Không có dữ liệu";
  record['e']= "Không có dữ liệu";
  record['f']= "Không có dữ liệu";
  record['g']= "Không có dữ liệu";
  record['h']= "Không có dữ liệu";
  record['j']= "Không có dữ liệu";
  record['k']= "Không có dữ liệu";
  record['l']= "Không có dữ liệu";
  record['m']= "Không có dữ liệu";
  data.push(record);
}
records.items = data;
var result=JSON.stringify(records);
return
ContentService.createTextOutput(result).setMimeType(ContentService.MimeType
.JSON);
}
```

