

ĐẠI HỌC QUỐC GIA
TRƯỜNG ĐẠI HỌC BÁCH KHOA TP HỒ CHÍ MINH



ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP

XÂY DỰNG HỆ THỐNG THU THẬP DỮ LIỆU
TRÊN NỀN TẢNG CẢM BIẾN KHÔNG DÂY MULTI-HOP
VÀ HỆ ĐIỀU HÀNH NHÚNG
PHỤC VỤ CHO ỨNG DỤNG INTERNET OF THINGS

GVHD: TS. LÊ TRỌNG NHÂN
GVPB: TS. PHẠM QUỐC CƯỜNG

Tên thành viên - Nhóm LoRa

MSSV	Tên
1512067	Lê Phương Nam
1512009	Nguyễn Văn Minh

TP Hồ Chí Minh - 2018

DẠI HỌC QUỐC GIA
TRƯỜNG ĐẠI HỌC BÁCH KHOA TP HỒ CHÍ MINH



ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP

XÂY DỰNG HỆ THỐNG THU THẬP DỮ LIỆU TRÊN NỀN TẢNG
CẢM BIẾN KHÔNG DÂY MULTI-HOP VÀ HỆ ĐIỀU HÀNH NHÚNG
PHỤC VỤ CHO ỨNG DỤNG INTERNET OF THINGS

Chuyên ngành: Kỹ Thuật Máy Tính

Người hướng dẫn khoa học:
TS. Lê Trọng Nhân

TP Hồ Chí Minh - 2018

Xây dựng hệ thống thu thập dữ liệu
trên nền tảng mạng cảm biến
không dây multi-hop và hệ điều
hành nhúng phục vụ cho ứng dụng
Internet of Things

Version 1.0

Revision history

Date	Version	Description
17/09/2018	0.1	Tìm hiểu chuẩn giao tiếp không dây LoRa Tìm hiểu hệ điều hành nhúng Android Things đang được hỗ trợ bởi Google cho các mạng cảm biến IoTs
02/10/2018	0.2	Nạp hệ điều hành nhúng Android Things lên mạch Raspberry Pi 3
10/10/2018	0.3	Tìm hiểu và thực hiện giao tiếp SPI giữa Raspberry Pi 3B và mạch LoRa Hat
08/12/2017	1.0	Đề xuất hướng phát triển trong giai đoạn luận văn, chỉnh sửa hoàn thiện chương 1,2,3 và điều chỉnh định dạng tài liệu

MỤC LỤC

Chương 1. Giới thiệu tổng quan	14
1.1 Tổng quan về mạng IoT và công nghiệp 4.0	14
1.1.1 Mạng IOT	14
1.1.2 Công nghiệp 4.0	15
1.2 Các ứng dụng tiềm năng dựa trên IoTs	16
1.3 Các thách thức hiện tại của mạng IoTs	24
1.4 Giới thiệu đề luận văn	29
1.4.1 Mục tiêu của luận văn	30
1.4.2 Mục tiêu cụ thể của giai đoạn thực tập	30
1.4.3 Các ứng dụng tiềm năng cho kết quả của luận văn .	30
Chương 2. Kiến trúc tổng quan của hệ thống	31
2.1 Giới thiệu	31
2.2 Kiến trúc hệ thống	32
2.3 Kiến trúc node cảm biến	33
2.4 Kiến trúc node gateway	34
2.5 Mô hình giao tiếp không dây	35
2.5.1 Giao tiếp ở gateway	35
2.5.2 Giao tiếp ở các node	36
Chương 3. Gateway IoTs dựa trên nền tảng Android Things	38
3.1 Tại sao phải cần hệ điều hành?	38
3.1.1 Vậy nền tảng (platform) là gì?	38
3.1.2 Vậy hệ điều hành là gì?	39
3.1.3 Tại sao lại cần đến hệ điều hành?	39

3.2	Giới thiệu hệ điều hành Android Things	40
3.3	Tại sao nên sử dụng Android Things?	40
3.4	Các nền tảng nhúng hỗ trợ Android Things	41
3.5	Xây dựng ứng dụng đầu tiên với Androids Things	41
3.5.1	Cài đặt JDK	42
3.5.2	Cài đặt Android Studio	50
3.5.3	Cài đặt hệ điều hành Android Things	55
3.5.4	Các công cụ hỗ trợ	84
3.5.5	Project LED Blinky	90
Chương 4. Hiện thực giao tiếp không dây cho node cảm biến		95
4.1	Giới thiệu	95
4.1.1	LoRa	96
4.1.2	Arduino	97
4.2	Cài đặt công cụ hỗ trợ	98
4.2.1	Arduino IDE	98
4.2.2	Thêm thư viện RadioHead	102
4.3	Hướng dẫn chi tiết	105
4.4	Các lỗi thường gặp và cách khắc phục	109
4.5	Do đặc hiệu suất của hệ thống	114
4.5.1	Khoảng cách gửi nhận dữ liệu	114
4.5.2	Ước lượng thời gian hoạt động của node cảm biến .	114
Chương 5. Hiện thực nhận dữ liệu cho gateway		115
5.1	Giới thiệu	115
5.1.1	LoRa	115
5.1.2	Android Things	115
5.2	Phát triển driver cho gateway: SPI	116
5.2.1	Giới thiệu SPI	116
5.2.2	Nguyên lý hoạt động	118
5.2.3	Driver SPI của Raspberry Pi 3 và LoRa HAT	120

Chương 6. Tổng kết	125
6.1 Các kết quả đạt được trong giai đoạn thực tập	125
6.1.1 Thực hiện kết nối giữa raspberry và Lora	125
6.1.2 Thực hiện kết nối giữa arduino và Lora	126
6.1.3 Truyền nhận giữ liệu ở các node Lora	126
6.2 Hướng phát triển trong giai đoạn luận văn	128
6.2.1 Hiện thực server	128
6.2.2 Hiện thực ứng dụng người dùng bằng ngôn ngữ hai nền tảng	129
6.2.3 Thực hiện kết nối nhiều node	129
Tài liệu tham khảo	130

LỜI CẢM ƠN

Để hoàn thành tài liệu này, chúng em xin tỏ lòng biết ơn sâu sắc đến thầy Lê Trọng Nhân, đã tận tình hướng dẫn trong suốt quá trình thực hiện đề tài.

Chúng em chân thành cảm ơn quý thầy cô trong khoa Khoa học và kỹ thuật máy tính trường Đại học Bách Khoa TP.HCM đã tận tình truyền đạt kiến thức trong những năm chúng em học tập. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình thực hiện đề tài mà còn là hành trang quý báu để chúng em bước vào đời một cách vững chắc và tự tin.

Cuối cùng, chúng em xin chúc quý thầy cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý.

DANH MỤC VIẾT TẮT

Từ viết tắt	Giải thích
IoT	Internet of Things
AT	Android Things
Rpi3	Raspberry Pi 3 Model B
CS1	Đại học Bách Khoa, cơ sở 1 - Lý Thường Kiệt
CS2	Đại học Bách Khoa, cơ sở 2 - Thủ Đức
KTXA	Ký túc xá khu A, Đại Học Quốc Gia - Thủ Đức
KTXXHH	Ký túc xá xã hội hóa, Đại Học Quốc Gia - Thủ Đức

Bảng 1: Chú giải từ viết tắt

DANH SÁCH HÌNH VẼ

1.1	Smarthome.	17
1.2	Thiết bị đeo tay thông minh.	18
1.3	Ô tô thông minh.	18
1.4	Internet công nghiệp.	19
1.5	Smart city.	20
1.6	IoT trong nông nghiệp.	21
1.7	Bán lẻ thông minh.	22
1.8	Năng lượng.	22
1.9	Sức khỏe.	23
1.10	IoT và chăn nuôi gia cầm, sản xuất nông trại.	24
1.11	Dự đoán IoT đến năm 2020	29
2.1	Kiến trúc tổng quan hệ thống	31
2.2	Kiến trúc hệ thống	33
2.3	Kết nối với node cảm biến.	33
2.4	Kết nối với node gateway.	34
2.5	Gateway kết nối với các node ban đầu.	35
2.6	node(5) kết nối với hai Gateway.	35
2.7	node(5) ngắt kết nối với Gateway1.	36
2.8	các node ở xa kết nối với Gateway.	36
2.9	Bắt tín hiệu các Gateway bên cạnh.	37
2.10	Kết nối với node phù hợp.	37
3.1	Oracle JDK	43
3.2	Chọn Phiên bản JDK	43

Đề cương luận văn 2018

3.3	File JDK tải về	44
3.4	Chạy File JDK tải về	45
3.5	Cài đặt đường dẫn file JDK tải về	46
3.6	Chọn Properties.	47
3.7	Chọn Advanced System Settings.	47
3.8	Chọn Advanced.	48
3.9	Tạo một biến môi trường có tên "JAVA_HOME".	48
3.10	Nhập Variable name và Variable value.	49
3.11	Nhấn vào Edit.	49
3.12	Nhấn vào Edit text.	50
3.13	Hoàn tất thêm vào "%JAVA_HOME%/bin".	50
3.14	File android studio.	52
3.15	Nhấn Next để cài đặt.	52
3.16	Nhấn Next để cài đặt.	53
3.17	Chọn ví trí lưu, nhấn Next để cài đặt.	53
3.18	Nhấn Install để cài đặt.	54
3.19	Nhấn Finish để cài đặt.	54
3.20	Màn hình giao diện - tạo Image Boot.	55
3.21	Nhấn + để tạo PRODUCT mới.	55
3.22	Thêm thông tin PRODUCT vào bảng.	56
3.23	Nhấn CREATE để tạo sản phẩm.	57
3.24	Tạo thành công sản phẩm.	57
3.25	Ấn NEW để tạo mới Image Boot.	58
3.26	Chọn Start from scratch.	58
3.27	Nhập tên Image Boot.	58
3.28	Chọn phiên bản Android Things mới nhất.	59
3.29	Chọn NEXT tại bước này.	59
3.30	Chọn NEXT tại bước này - Xem thêm 3.5.4.1.	60
3.31	Chọn NEXT tại bước này.	60
3.32	Thông tin về bản Image Boot.	61
3.33	Ấn CREATE BUILD để tạo.	61

3.34	Ấn Download tại bản Image Boot muốn tải.	61
3.35	Chọn Development.	62
3.36	Chờ để tải xuống.	62
3.37	Đã tải thành công.	62
3.38	Nhấn Cancel sau khi cắm thẻ nhớ vô máy.	63
3.39	Nhấn Refresh để tự động tìm thẻ nhớ.	63
3.40	Nhấn Format để định dạng.	64
3.41	Nhấn Yes.	64
3.42	Nhấn OK.	65
3.43	Giải nén tập tin Image Boot vừa tải.	65
3.44	Nhận được tập tin "iot_rpi3.img".	65
3.45	Chạy phần mềm Win32 Disk Image.	66
3.46	Chọn đường dẫn tới tập tin Image Boot đã được giải nén.	66
3.47	Chọn thẻ nhớ.	67
3.48	Chọn thẻ nhớ.	67
3.49	Nhấn Yes để tiếp tục.	68
3.50	Chờ đợi quá trình ghi hệ điều hành.	68
3.51	Nhấn Cancel.	69
3.52	Nhấn OK để tắt thông báo.	69
3.53	Nhấn Exit để thoát chương trình.	70
3.54	Nhấn Cancel sau khi cắm thẻ nhớ vô máy.	70
3.55	Nhấn DOWNLOAD để tải phần mềm.	71
3.56	Giải nén phần mềm.	71
3.57	Chuột phải, chọn Run as administrator.	71
3.58	Nhấn 1.	72
3.59	Nhấn 1 để tiếp tục.	72
3.60	Lựa chọn phiên bản để ghi.	72
3.61	Nhấn 1 nếu cài bản mặc định.	73
3.62	Nhấn 2 để cài bản đã tải tại Bước 1.	73
3.63	Dán đĩa chỉ chứa tập tin Image Boot .zip.	74
3.64	Nhấn ENTER để chọn thẻ nhớ.	74

Đề cương luận văn 2018

3.65 Chọn thẻ nhớ rồi nhấn ENTER.	75
3.66 Nhấn y, rồi nhấn ENTER.	75
3.67 Nhấn n rồi nhấn ENTER.	76
3.68 Nhấn ENTER để thoát.	76
3.69 Cắm các kết nối trước rồi mới cắm nguồn.	77
3.70 Chờ hệ điều hành khởi động.	77
3.71 Chọn Networks.	78
3.72 Chọn Wi-Fi.	78
3.73 Ấn nút bên phải để bật wifi.	79
3.74 Danh sách wifi khả dụng.	79
3.75 Nhập mật khẩu và nhấn CONNECT.	79
3.76 Nhấn < để về màn hình chính.	80
3.77 Networks hiển thị IP của mạch Rpi3.	80
3.78 Cắm dây Ethernet và cắm nguồn.	81
3.79 Chạy chương trình Advanced IP Scanner để quét địa chỉ IP của Rpi3.	81
3.80 Truy cập thư mục platform-tools.	82
3.81 Chạy Command Prompt tại thư mục platform-tools.	82
3.82 Kết nối với thiết bị "adb connect <IP>".	82
3.83 Kết nối với mạch Raspberry Pi 3 thành công.	83
3.84 Hủy kết nối với một thiết bị "adb disconnect <IP>".	83
3.85 Hủy kết nối với tất cả thiết bị "adb disconnect".	83
3.86 Chọn UPLOAD.	84
3.87 Chọn biểu tượng thư mục.	85
3.88 Chọn file "bootanimation.zip", nhấn OK.	85
3.89 Tải tập tin thành công, ấn NEXT để tiếp tục.	86
3.90 Lệnh "ipconfig" trên Windows.	87
3.91 Lệnh "ifconfig" trên Mac/Linux.	87
3.92 Nhấn Scan để quét.	88
3.93 Tệp adb được tải về.	89
3.94 Giao diện adb.	90

3.95 Mạch blink led.	91
3.96 Giao diện android.	92
3.97 IP raspberry.	92
3.98 Gõ lệnh adb.	93
3.99 Kết nối PC với raspberry.	93
3.100 Hiển thị kết nối trên androidstudio.	94
3.101 Kết quả hiển thị trên raspberry.	94
 4.1 Gateway LoRa	95
4.2 Giới thiệu công nghệ LoRa [15]	96
4.3 Giới thiệu LoRaWan	97
4.4 Giới thiệu Arduino	98
4.5 Truy cập trang tải Arduino	98
4.6 Chọn JUST DOWNLOAD	99
4.7 Chọn I Agree	99
4.8 Chọn Next	100
4.9 Chọn Install	100
4.10 Chờ đợi cài đặt Arduino IDE	101
4.11 Chọn Install để cài đặt driver	101
4.12 Tải tập thư viện từ Github	102
4.13 Nhấn OK để tải	102
4.14 Thêm thư viện cho Arduino IDE	103
4.15 Chọn OK để thêm	103
4.16 Kiểm tra thư viện đã được thêm	104
4.17 Mở code bên gửi "rf95_client"	105
4.18 Nhấn dấu mũi tên ⇒ để Upload	106
4.19 Mở code bên nhận "rf95_server"	107
4.20 Thay đổi chân số 9 thành số 8	108
4.21 Nhấn dấu mũi tên ⇒ để Upload	109
4.22 Xem phần thông báo lỗi	110
4.23 Kiểm tra "Tools" → "Board: Arduino/Genuino Uno"	110
4.24 Kiểm tra "Tools" → "Port" hoặc "Serial Port"	111

Đề cương luận văn 2018

4.25 Nối chân jump cho mạch LoRa	111
5.1 Gateway LoRa	115
5.2 Khái niệm SPI	116
5.3 Sơ đồ đường đi của SPI	117
5.4 Nguyên lý hoạt động của SPI	118
5.5 Dạng sóng của 4 mode SPI	119
6.1 Kết nối giữa Lora và raspberry	125
6.2 Kết nối giữa Lora và arduino	126
6.3 Tuyền dữ liệu giữa 2 node	127
6.4 Tuyền dữ liệu giữa 3 node	127
6.5 Tuyền dữ liệu giữa 3 node	128
6.6 Tuyền dữ liệu qua node trung gian	128

DANH SÁCH BẢNG

1	Chú giải từ viết tắt	6
4.1	Khoảng cách gửi nhận dữ liệu đo đặc được	114
5.1	Các mode trong giao tiếp SPI [11]	119

CHƯƠNG 1

GIỚI THIỆU TỔNG QUAN

1.1 Tổng quan về mạng IoT và công nghiệp 4.0

1.1.1 Mạng IOT

Internet of things là Mạng lưới vạn vật kết nối Internet hoặc là Mạng lưới thiết bị kết nối Internet viết tắt là IoT. khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. IoT đã phát triển từ công nghệ không dây, công nghệ vi cơ điện tử và Internet. Hay có thể hiểu là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.

Việc kết nối các thiết bị có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại... Các thiết bị có thể là điện thoại thông minh, máy pha cafe, máy giặt, tai nghe, bóng đèn, và nhiều thiết bị khác[1].

Ước tính trước năm 2020, sẽ có ít nhất 26 tỉ thiết bị được kết nối với Internet. IoT là một mạng lưới khổng lồ nơi mà tất cả mọi “thứ” kết nối với nhau (con người với con người, con người với máy móc, máy với máy).Mạng lưới kết nối khổng lồ, vô tận này đem đến nhiều lợi ích cho cả cá nhân và doanh nghiệp. Xe hơi thông minh hay nhà thông minh là

những ví dụ điển hình. Ví dụ bạn vừa thức dậy, lập tức chiếc đồng hồ báo thức báo hiệu cho máy pha cà phê, đồng thời rèm cửa cũng tự động được kéo và vòi sen cũng tự động mở nước cho bạn. Một ví dụ khác, bạn bị trễ buổi họp vì kẹt xe, chiếc xe hơi thông minh của bạn sẽ lập tức gửi tin nhắn thông báo đến nhân viên của bạn hoặc tìm một hướng đi khác. Trên diện rộng, một khi công nghệ đã phát triển đến một tầm cao mới, nó có thể biến thành phố thành các siêu đô thị thông minh nhằm giải quyết các vấn đề như thiếu hụt năng lượng và quản lý rác thải. Khả năng ứng dụng IoT vào cuộc sống hằng ngày của chúng ta là vô tận[2].

1.1.2 Công nghiệp 4.0

Industry 4.0 mô tả về một môi trường mà máy tính, tự động hóa và con người nói chung sẽ cùng nhau làm việc theo một cách hoàn toàn mới. Những con robot, hay máy móc nói chung, sẽ được kết nối vào những hệ thống máy tính. Các hệ thống này sử dụng thuật toán machine learning để học hỏi và điều khiển máy móc, cần rất ít hoặc thậm chí là không cần sự can thiệp nào từ con người cả. Đây là lý do mà nhiều người gọi Industry 4.0 như là một "nhà máy thông minh". Và để có đủ dữ liệu phục vụ cho Industry 4.0, các máy móc phải "cống hiến" dữ liệu ngược lại về hệ thống trung tâm cũng như thu thập dữ liệu từ các nguồn bên ngoài thì quyết định được máy đưa ra mới chính xác.

Lợi ích công nghiệp 4.0

1. Sản xuất nhanh hơn, tốn ít sức người hơn, dữ liệu thu thập đầy đủ hơn, quyết định được đưa ra nhanh chóng hơn.
2. Con người sẽ được làm những việc vui vẻ hơn, hấp dẫn hơn, không bị nhảm chán. Những thứ này để máy làm.
3. Trong những môi trường làm việc nguy hiểm, con người không phải xuất hiện nên giảm tỉ lệ tử vong, bệnh tật cho người lao động.
4. Kiểm soát được món hàng từ nguyên vật liệu cho đến khi thành hình và chuyển đến tay người tiêu dùng

5. Đảm bảo chất lượng đồng đều giữa các mẻ thành phẩm (vì máy làm tự động, không phải người làm)
6. Khi có dữ liệu càng chi tiết và càng nhiều, các thuật toán machine learning lại càng chạy chính xác hơn để đưa ra những quyết định tốt hơn.
7. Các công ty sẽ giảm chi phí, tăng thị phần, lợi nhuận.

Dữ liệu trong Cách mạng công nghiệp 4.0 rất quan trọng. Để các thuật toán machine learning chạy được, và chạy đúng, cần rất nhiều dữ liệu thu thập liên tục từ nhiều máy móc, hệ thống khác nhau: hệ thống nhân sự, hệ thống kế toán, hệ thống vận hành kho bãi, nhà máy, hệ thống bán hàng... Nếu dữ liệu không được thu thập đúng và đủ, thuật toán sẽ chạy sai, công thức sẽ bị lệch khiến thành phẩm đưa ra không giống như ý muốn.

Do đó, dữ liệu là thứ quan trọng số 1, không chỉ với thế giới Internet of Things mà còn với Industry 4.0. Không có dữ liệu, những thứ mà người ta vẽ ra về Industry 4.0 chỉ là trên lý thuyết và mãi mãi không bao giờ áp dụng ra thực tế được[9].

1.2 Các ứng dụng tiềm năng dựa trên IoTs

Internet of Things giúp kết nối các thiết bị và nâng cao tính tự động hóa của các loại máy móc. Để hiểu hơn về cách áp dụng IoT sau đây là các ứng dụng tiềm năng của IoTs.

1. Smart home

Có thể nói smart home chính là ứng dụng được tìm kiếm nhiều nhất trên google. Vậy như thế nào được hiểu là một ngôi nhà thông minh? Bạn sẽ có thể bật điều hòa, bình nóng lạnh trước khi về nhà hay thậm chí tắt đèn ngay khi bạn không có nhà, bạn có thể mở cửa cho bạn bè vào chơi trong khi bạn vẫn còn ở cơ quan. Các công ty đang xây dựng và sản xuất hàng loạt các sản phẩm để làm cho cuộc sống con người

đơn giản và thuận tiện hơn. Smart home chính là bậc thang mang tính cách mạng của quá trình phát triển xu hướng IoT. Sự xuất hiện của smart home được dự đoán sẽ trở nên phổ biến như smart phone hiện nay.

Để sống trong một ngôi nhà thông minh như vậy chủ nhà phải bỏ ra chi phí sở hữu nhà vô cùng lớn. Các sản phẩm trong các ngôi nhà thông minh được dự đoán sẽ giúp tiết kiệm thời gian, năng lượng và tiền bạc.



Hình 1.1: Smarthome.

2. Các thiết bị đeo thông minh

Hiện nay ở nhiều nước đã xuất hiện các thiết bị đeo trên người với những tính năng vô cùng thông minh như: tai nghe, các loại kính, ba lô, vòng tay siêu thông minh,... Những thiết bị này dần bùng nổ tại các thị trường trên toàn thế giới. Google và Samsung là những công ty lớn có những khoản đầu tư khổng lồ cho việc tạo ra các thiết bị như vậy. Các thiết bị đeo được cài đặt cảm biến và các phần mềm thu thập dữ liệu, thông tin người dùng. Các thiết bị này bao gồm các yêu cầu về thể chất, sức khỏe và có tính giải trí cao. Điều kiện tiên quyết cho các thiết kế này là công suất cực thấp và kích thước nhỏ gọn, có tính thẩm mỹ cao.



Hình 1.2: Thiết bị đeo tay thông minh.

3. Những chiếc ô tô được kết nối



Hình 1.3: Ô tô thông minh.

Các nhà sản xuất ô tô đã bước qua giai đoạn tập trung vào việc tối ưu hóa các chức năng nội bộ của một chiếc xe. Giờ đây họ quan tâm đến việc tối ưu hóa sự hài lòng của người sử dụng với việc nâng cao trải nghiệm trong xe hơi.

Một chiếc xe được kết nối là chiếc xe có khả năng tối ưu hóa hoạt động, bảo trì cũng như sự thoải mái của khách hàng khi sử dụng.

Các thương hiệu lớn như BMW, Tesla, ... đang nỗ lực cho cuộc cách mạng tiếp theo của ngành sản xuất ô tô.

4. Internet công nghiệp



Hình 1.4: Internet công nghiệp.

Industrial Internet là tiếng vang mới trong ngành công nghiệp, được gọi tắt là IIoT (Industrial Internet of Thing). IIoT hỗ trợ kĩ thuật công nghiệp với các cảm biến, phần mềm lớn để tạo ra những cỗ máy vô cùng thông minh. Máy móc sẽ có tính chính xác và nhất quán hơn con người trong giao tiếp thông qua dữ liệu. Từ những dữ liệu thu thập được giúp các công ty, nhà quản lí giải quyết các vấn đề sớm hơn, đạt hiệu quả cao hơn.

IIoT có tiềm năng lớn về kiểm soát chất lượng và tính bền vững. Những ứng dụng trao đổi thông tin giữa nhà cung cấp, nhà phân phối và nhà bán lẻ về thông tin hàng hóa, hàng tồn kho sẽ làm tăng hiệu quả chuỗi cung ứng.

5. Smart city Thành phố thông minh là một ứng dụng của IoT tạo được sự tò mò của đông đảo người dân. Giám sát thông minh, vận chuyển tự

động, hệ thống quản lý năng lượng thông minh hơn, phân phối nước, an ninh đô thị và giám sát môi trường tất cả là ví dụ về internet của các ứng dụng cho thành phố thông minh. IoT giúp giải quyết các vấn đề gặp phải tại các thành phố lớn đó là ô nhiễm môi trường, tắc nghẽn giao thông và thiếu năng lượng. Một ví dụ có thể kể đến của các thiết bị được sử dụng truyền thông di động như: thùng rác thông minh, chúng sẽ gửi cảnh báo đến bộ phận vệ sinh môi trường khi cần dọn sạch.

Bằng cách cài đặt ứng dụng và dùng các thiết bị thông minh chúng ta hoàn toàn có thể dễ dàng tìm thấy các cây xăng, siêu thị, quán ăn hay thậm chí là những bãi gửi xe miễn phí. Ngoài ra hệ thống điện cũng được bảo vệ bởi các cảm biến sẽ giúp phát hiện nhanh chóng các vấn đề gây nhiễu, trực trặc, hay các vấn đề về lắp đặt.



Hình 1.5: Smart city.

6. IoT trong nông nghiệp Với sự gia tăng liên tục của dân số đồng nghĩa với việc nhu cầu sử dụng lương thực tăng lên nhiều lần. Nông dân có thể áp dụng các kỹ thuật mới, công nghệ tiên tiến để tăng sản lượng sản xuất nông nghiệp. Nông nghiệp thông minh có thể nói là lĩnh vực phát triển nhanh nhất với IoT.

Những thông tin người nông dân thu được giúp họ có những quyết định đầu tư sáng suốt tránh tình trạng “được mùa mất giá, được giá mất mùa” như hiện nay. Cảm biến độ ẩm, chất dinh dưỡng của đất, mức độ hấp thụ nước góp phần quan trọng vào việc kiểm soát sự tăng trưởng của cây trồng giúp người gieo trồng có thể xác định, tùy chỉnh lượng phân bón cần thiết.



Hình 1.6: IoT trong nông nghiệp.

7. Bán lẻ thông minh IoT tạo nên một sự kết nối mật thiết giữa nhà bán lẻ với khách hàng giúp nâng cao trải nghiệm của khách hàng khi đến với cửa hàng. Tiềm năng phát triển IoT trong lĩnh vực bán lẻ là vô cùng lớn.

Smart phone là thiết bị phổ biến nhất được sử dụng để các nhà bán lẻ duy trì kết nối với khách hàng của mình khi khách hàng đến với cửa hàng hay thậm chí ngay cả khi họ ra khỏi cửa hàng. Tương tác qua điện thoại và việc sử dụng các công nghệ giúp cho các nhà bán lẻ phục vụ khách hàng tốt hơn, thay đổi cách bài trí cửa hàng cho phù hợp với nhu cầu tiêu dùng.



Hình 1.7: Bán lẻ thông minh.

8. Năng lượng



Hình 1.8: Năng lượng.

Mạng lưới điện trong vài năm tới sẽ trở nên thông minh và đáng tin cậy hơn. Khái niệm lưới điện thông minh đang trở nên phổ biến trên toàn thế giới.

Dữ liệu được thu thập một cách tự động để phân tích hành vi tiêu dùng điện của người dùng và nhà cung cấp để góp phần nâng cao

hiệu quả sử dụng điện. Lưới điện thông minh giúp phát hiện nguồn ngắn điện thông minh ở cấp độ các hộ gia đình.

9. Sức khỏe

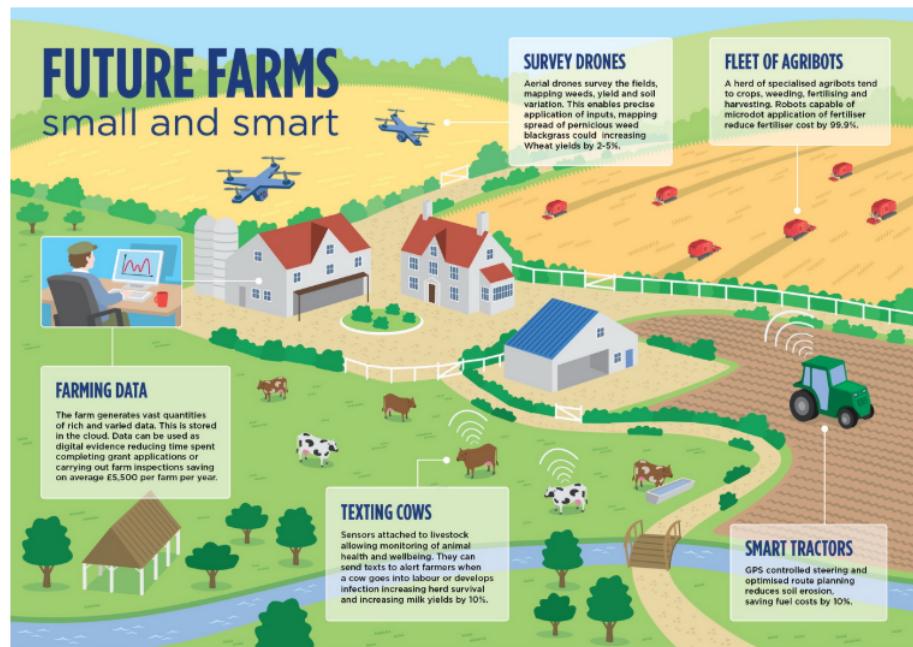


Hình 1.9: Sức khỏe.

Đây có thể nói là một lĩnh vực chưa được khai phá hết của Internet of Things bởi những ứng dụng không ngờ mà nó mang lại. Một hệ thống chăm sóc sức khỏe được kết nối cùng các thiết bị y tế thông minh mang lại tiềm năng to lớn cho các công ty đầu tư sản xuất. IoT trong chăm sóc sức khỏe giúp mọi người có cuộc sống khỏe mạnh hơn bằng việc đeo các thiết bị kết nối. Các dữ liệu thu thập được giúp phân tích sức khỏe của người dùng thiết bị kết nối và nhà cung cấp, sản xuất sẽ có được những thiết kế để chống lại bệnh tật.

10. IoT và chăn nuôi gia cầm, sản xuất nông trại. Kiểm soát các khâu trong quy trình chăn nuôi giúp tiết kiệm thời gian và chi phí. Sử dụng các công cụ IoT để thu thập dữ liệu về sức khỏe của gia súc, các chủ trang trại có thể biết sớm về bệnh tật của động vật giúp ngăn ngừa số lượng lớn các gia súc bị bệnh bởi virus lây lan. Với những dữ liệu

thu thập được cũng giúp chủ trang trại tăng nhanh được sản lượng gia súc, gia cầm.



Hình 1.10: IoT và chăn nuôi gia cầm, sản xuất nông trại.

Trong tương lai những gì IoT mang lại sẽ vượt xa những ứng dụng kể trên. IoT sẽ mang lại sự thay đổi tầm vĩ mô trong cách con người sống và làm việc[13].

1.3 Các thách thức hiện tại của mạng IoTs

Internet of Things - IoT có thể xem là đại diện của mọi thiết bị thông minh nhưng xu hướng này còn gặp nhiều khó khăn trong việc triển khai một cách toàn diện. IoT theo định nghĩa hiểu đơn giản là các bộ cảm biến có thể giao tiếp với Internet mà không cần có sự can thiệp của con người. Định nghĩa đó bao gồm cả các thiết bị đeo và điện thoại thông minh, hay các thiết bị có thể tự động giao tiếp nếu được cấp phép. IoT cũng bao gồm các thiết bị thông minh dành cho xe hơi, nhà ở và cả một thành phố hay thiết bị y tế hoặc công cụ trong các ngành công nghiệp. Những lợi ích

và tiềm năng phát triển của IoT ở khắp mọi nơi, ở đâu có kết nối đều có khả năng xuất hiện các thiết bị với định danh của riêng mình, mang đến giá trị thông qua truyền tải, trao đổi thông tin, dữ liệu. Có thể thuật ngữ này sẽ biến mất khi IoT trở thành một chuẩn mực, và có thể được thay thế bằng một cái tên gọi khác nhằm ám chỉ những thiết bị không thể giao tiếp. Nhưng một số thách thức có thể làm trì hoãn tương lai của IoT và sau đây là những trở ngại đáng chú ý.

- **An ninh và bảo mật dữ liệu:** IoT đã trở thành một mối quan tâm đáng lo ngại về an ninh, nó đã thu hút sự chú ý của các công ty công nghệ nổi tiếng và các cơ quan chính phủ trên toàn thế giới. Việc hack vào các thiết bị IoT như màn hình theo dõi em bé, tủ lạnh thông minh, máy bơm nước, điều hòa nhiệt độ, camera hay thậm chí là radio trong ô tô đang trở thành mối lo ngại về an ninh do sự phát triển của IoT gây ra. Vì vậy, các thiết bị mới được thêm vào mạng sẽ là yếu tố để các hacker có thể tấn công và xâm nhập dữ liệu trái phép vì số lượng lỗ hổng bảo mật là khá đáng kể. Mọi vấn đề bảo mật chỉ tốt khi chúng ta có thể chỉ ra các điểm yếu của thiết bị, và đối với một thế giới kết nối như hiện nay thì điều đó có rất nhiều.

Điều này giải thích lý do tại sao Samsung đã dành nỗ lực đáng kể vào nền tảng ARTIK dành cho IoT trong thời gian gần đây. ARTIK có 3 mẫu module chứa tất cả các thành phần – bộ cảm biến, vi xử lý, bộ nhớ tích hợp, và kèm theo đó là khả năng kết nối không dây cần thiết cho các nhà sản xuất để tạo ra thiết bị thông minh. Tất cả các module ARTIK đều được hang trang bị một khoá an toàn nhằm giúp các nhà phát triển mã hoá dữ liệu tốt hơn so với phần mềm mã hoá mặc định.

Đối với các thiết bị cá nhân có khả năng kết nối Internet thì vấn đề an ninh và sự riêng tư là những mối quan tâm hàng đầu. Đây có lẽ là

những sản phẩm điển hình để được trang bị hệ thống mã hóa, nhưng vẫn đề an ninh và sự riêng tư lại có đặc thù riêng khác nhau. An ninh bảo mật thường gắn liền với công nghệ còn sự riêng tư thì thường liên quan đến con người và tính pháp lý. Các nhà sản xuất thiết bị IoT cần phải hiểu rằng an ninh và sự riêng tư không thể đồng nhất hoặc áp dụng chung mọi quy tắc. Khả năng giao tiếp tự động của thiết bị IoT làm cho việc đảm bảo sự riêng tư khó khăn hơn bởi các mô hình sản phẩm được khuyến khích sử dụng trước khi có sự đồng thuận của người dùng ở những thời điểm khác nhau.

- **Tiêu chuẩn chung:** Việc thiếu các tiêu chuẩn, đặc biệt là trường hợp sử dụng nhiều giao thức kết nối như hiện nay, là một cản trở cho IoT phát triển. Nhiều giao thức kết nối đặc biệt đang nổi lên với mức tiêu thụ năng lượng thấp như LTE Cat.0, 802.11ah, Sigfox hay OnRamp. Công nghệ bộ xử lý hiện cũng chưa thực sự hào hứng với thị trường IoT khi chuẩn giao thức không thực sự rõ ràng.

Các hãng công nghệ như LG, Panasonic, Sharp, Silicon Image, TP-Link, HTC, Qualcomm và hơn 100 thành viên khác đã thành lập nên liên minh AllSeen, dẫn đầu là Hiệp hội Linux. Tiêu chí của liên minh này là xóa bỏ những rào cản cũng như thúc đẩy sự sáng tạo trong việc phát triển Internet of Things. Nhóm này đã xây dựng nền tảng nguồn mở AllJoyn cho phép các sản phẩm IoT có thể giao tiếp với nhau thông qua nhiều dạng kết nối từ Wi-Fi, Ethernet, và cả đường dây điện. AllJoyn có thể tương thích với mọi hệ điều hành hiện nay và cũng không bắt buộc các thiết bị phải kết nối vào Internet bởi chúng có thể liên lạc ở cấp độ ngang hàng.

Open Internet Consortium (OIC): Đây được xem là đối thủ của AllSeen Alliance, tổ chức OIC được các ông lớn công nghệ gồm Intel, Broadcom, Dell và Samsung chống lưng nhằm phát triển các tiêu

chuẩn và chứng nhận cho các thiết bị Internet of Things. Các tiêu chuẩn này cũng xoay quanh khả năng giao tiếp và chứng thực thiết bị dựa trên các giao thức kết nối khác nhau gồm Wi-Fi, Bluetooth và cả NFC.

Thread Group: Tổ chức phi lợi nhuận này được thành lập bởi Nest Labs (thuộc Google), Samsung, ARM, Freescale, Silicon Labs... Thread Group tạo ra một giao thức mạng không dây dựa trên IP, cho phép các thiết bị phần cứng trong nhà kết nối với đám mây. Mục tiêu mà Thread Group còn nhắm đến việc giảm mức tiêu thụ năng lượng của thiết bị và đảm bảo tính an toàn bảo mật kết nối với IPv6. Tổ chức này dường như chỉ tập trung vào nền tảng hạ tầng hoạt động của IoT chứ không can thiệp quá nhiều vào phần cứng. Điều này cũng giúp dễ dàng tương thích với các tiêu chuẩn khác như AllSeen hay OIC.

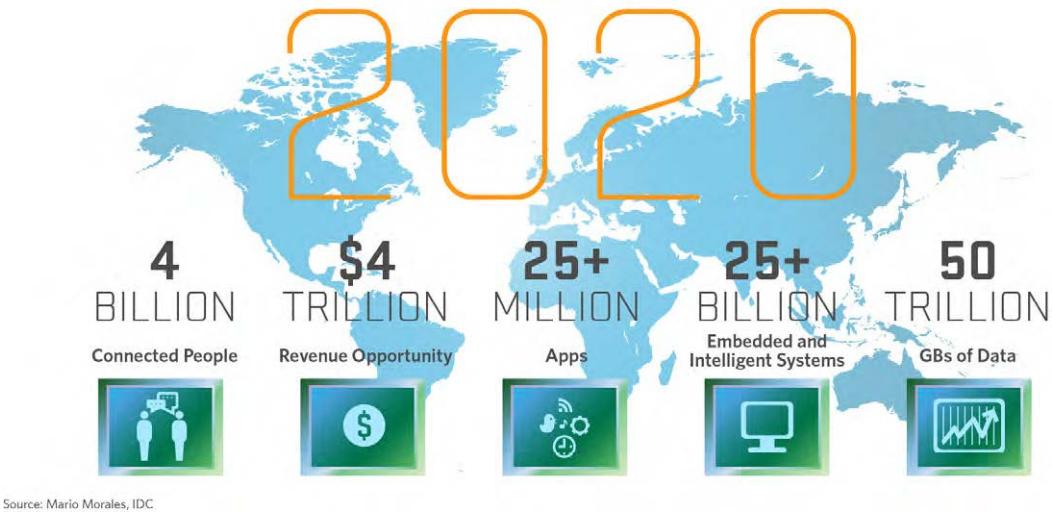
Industrial Internet Consortium (IIC): Đây là tổ chức thứ 2 mà Intel tham gia vào nhằm phát triển IoT, ngoài ra General Electric, Cisco Systems, IBM và nhà mạng AT&T là những thành viên thành viên tích cực nhất. Tuy nhiên, IIC tập trung vào mảng thiết bị IoT dùng cho doanh nghiệp và đảm bảo mọi thứ cùng hoạt động tốt ở mọi phân khúc thị trường. Ngoài ra, IIC giúp cải tiến các hệ thống máy móc lỗi thời có thể tham gia vào hệ thống IoT.

IEEE P2413: Viện kĩ thuật điện tử (IEEE) là một trong những tổ chức chính quy có nhiệm vụ đặt ra các tiêu chuẩn quan trọng trong thế giới công nghệ. Nhưng trong xu hướng IoT thì IEEE bị các công ty công nghệ cho rằng quá chậm chạp trong việc thiết lập tiêu chuẩn. IEEE quy tụ 23 nhà sản xuất có liên quan và cùng nghiên cứu tạo nên bộ chuẩn chung cho thiết bị.

- **Kết nối:** Kết nối rất nhiều thiết bị sẽ là một trong những thách thức lớn nhất của tương lai IoT, và sẽ phải tái cấu trúc các mô hình truyền thông hiện tại và công nghệ kết nối. Hiện tại, chúng ta đang dựa và mô hình tập trung, server-client để xác thực, ủy quyền các nút khác nhau trong một mạng.

Mô hình này là đủ cho hệ sinh thái IoT hiện tại, khi hàng chục, hàng trăm, hàng nghìn thiết bị kết nối. Nhưng khi IoT phát triển, hàng tỷ và hàng trăm tỷ thiết bị kết nối, hệ thống tập trung hiện tại sẽ bị thắt cổ chai. Các hệ thống lớn để đáp ứng số lượng lớn thiết bị này sẽ yêu cầu sự đầu tư khổng lồ vào các máy chủ đám mây để có thể xử lý lượng trao đổi thông tin khổng lồ.

IDC và EMC ước tính rằng tới năm 2020, sẽ có 44 nghìn tỉ gigabyte dữ liệu trên thế giới và 10 trong số đó là đến từ các thiết bị IoT. Mức này tương đương với 4.4 nghìn tỉ gigabyte, đồng thời cũng tương đương với mức dung lượng tất cả dữ liệu trên thế giới ước tính vào năm 2013. Nói cách khác, sức chứa của tất cả các trung tâm dữ liệu trực tuyến trong năm 2013 chỉ đủ để lưu trữ dữ liệu IoT trong năm 2020. Điều này sẽ có ảnh hưởng lớn tới cách các trung tâm dữ liệu làm việc để chúng có thể chịu được lượng dữ liệu lưu trữ trên đám mây khi các dữ liệu này được tạo ra và trao đổi.



Hình 1.11: Dự đoán IoT đến năm 2020

Còn quan trọng hơn cả là yêu cầu kết nối để giúp tất cả các thiết bị này giao tiếp với nhau, để có thể gửi dữ liệu đến và đi khỏi kho lưu trữ dữ liệu trực tuyến. Giao thức IPv6 được tạo dã phán nàò nghĩ tới tương lai này khi nó hỗ trợ 2^{128} địa chỉ, so với chỉ 2^{32} địa chỉ của giao thức IPv4. Lưu lượng dữ liệu trên 1 thiết bị cá nhân có thể không quá nhiều. Bluetooth cũng có thể đủ dùng với nhiều mạng cảm biến đòi hỏi mức năng lượng thấp, và rất nhiều trong số này cũng được tải lên rải rác chứ không stream mọi thông tin cùng 1 lúc.

Thế nhưng, tổng hợp hàng tỉ thiết bị trực tuyến vào năm 2020 và tất cả lượng băng thông sẽ là rất lớn. Quản lý dung lượng kết nối mạng một cách cẩn trọng sẽ là việc vô cùng thiết yếu.

1.4 Giới thiệu đề luận văn

Với sự phát triển của Internet of Things (IoTs), rất nhiều ứng dụng tiện ích đã được hiện thực. Tuy nhiên, các ứng dụng hiện tại đều tập trung vào việc gửi dữ liệu trực tiếp từ node cảm biến về node trung tâm. Điều này

làm cản trở việc triển khai ứng dụng trên diện rộng, khi tầm bao phủ của ứng dụng lớn hơn phạm vi gửi dữ liệu không dây. Do đó, việc phát triển một nền tảng cho phép gửi dữ liệu qua nốt trung gian (multi-hop) sẽ mở ra nhiều cơ hội cho việc triển khai ứng dụng. Bên cạnh đó, node trung tâm cũng cần dựa trên một hệ điều hành, thay vì chủ yếu là nền tảng vi điều khiển như hiện tại, để có thể hỗ trợ nhiều xử lý đa dạng và phức tạp (ví dụ: kết nối 3G, cập nhật firmware từ xa) cũng như nâng cao độ ổn định của hệ thống.

1.4.1 Mục tiêu của luận văn

Xây dựng hệ thống thu thập dữ liệu trên nền tảng mạng cảm biến không dây multi-hop và hệ điều hành nhúng phục vụ cho ứng dụng Internet of Things.

1.4.2 Mục tiêu cụ thể của giai đoạn thực tập

Tìm hiểu các chuẩn giao tiếp không dây hiện có, lựa chọn tiêu chuẩn giao tiếp phù hợp dựa vào các tiêu chí: tầm xa, năng lượng, độ ổn định (tỉ lệ lỗi thấp).

Tìm hiểu các hệ điều hành nhúng đang được hỗ trợ phổ biến cho các mạng cảm biến IoTs (ví dụ: Android things, Windows things,...). Lựa chọn hệ điều hành phù hợp cho đề tài.

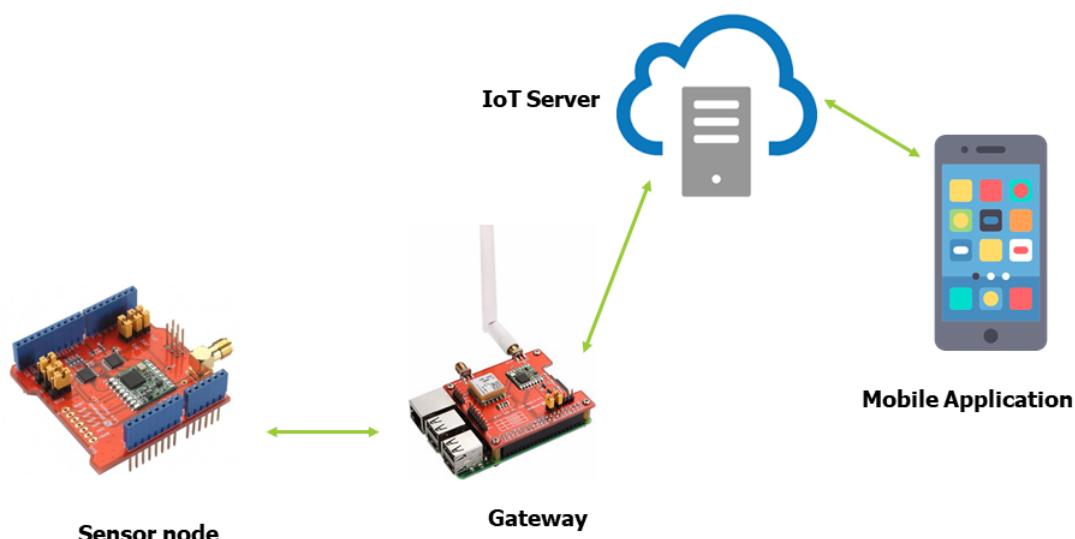
1.4.3 Các ứng dụng tiềm năng cho kết quả của luận văn

Xây dựng một ứng dụng giám sát trên phạm vi rộng (ví dụ môi trường nước hoặc môi trường không khí) để kiểm tra tính đúng đắn của đề tài.

CHƯƠNG 2

KIẾN TRÚC TỔNG QUAN CỦA HỆ THỐNG

2.1 Giới thiệu



Hình 2.1: Kiến trúc tổng quan hệ thống

Hệ thống Lora được sử dụng trong IOT vì LoRa có thể được áp dụng rộng rãi trong các ứng dụng thu thập dữ liệu như sensor network trong đó các sensor node có thể gửi giá trị đo đạc về trung tâm cách xa hàng km và có thể hoạt động với battery trong thời gian dài trước khi cần thay pin . Một đặc thù riêng biệt, ở môi trường nông nghiệp thì việc truyền dữ liệu từ các cảm biến và trung tâm, và từ trung tâm tới các thiết bị chấp hành sẽ gặp phải các khó khăn về khoảng cách, dễ bị tác động của môi trường,...đến hệ thống không hoạt động ổn định. Ngoài Zigbee, LoRa là một lựa

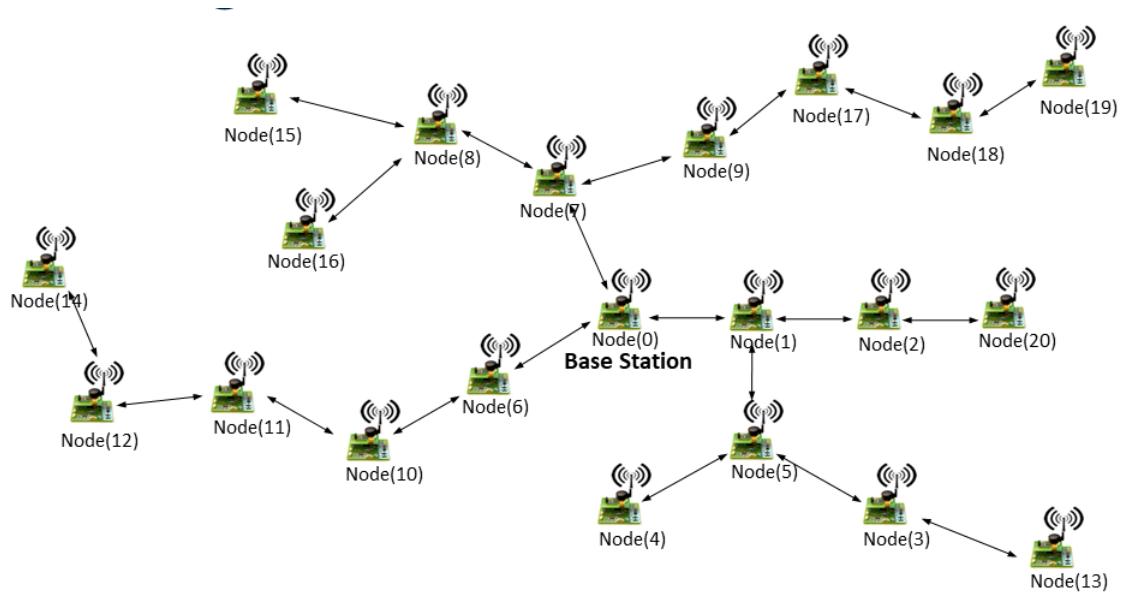
chọn tuyệt vời. Về tổng quan hệ thống bao gồm:

1. Các sensor node bao gồm 1 board lora và 1 board arduino. Các node này có nhiệm vụ truyền nhận dữ liệu trong mạng. Giúp dữ liệu từ các cảm biến có thể đến gateway.
2. Gateway bao gồm 1 Lora và 1 raspberry. với khả năng tính toán của raspberry thì sẽ đầy được nhiều dữ liệu lên server cùng một lúc.
3. Server trực tiếp sử dụng mqtt.MQTT (Message Queuing Telemetry Transport) là một giao thức gởi dạng publish/subscribe sử dụng cho các thiết bị Internet of Things với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định.
Bởi vì giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng M2M.
4. Application là điện thoại hoặc PC. Giúp người dùng theo dõi dữ liệu trong hệ thống mạng.

2.2 Kiến trúc hệ thống

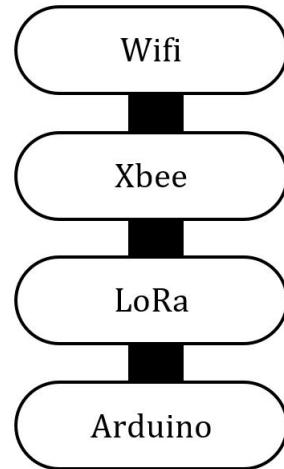
Hệ thống bao gồm các gateway được đặt ở những nơi có điện và có khả năng kết nối internet. Vì gateway tiếp nhận nhiều dữ liệu nên tiêu tốn nhiều điện cần đặt ở nơi có nguồn điện ổn định.Internet để kết nối lên server.

Gateway sẽ kết nối dữ liệu với các node xung quanh để truyền dữ liệu, cấp ID, giúp tìm đường truyền...Các node sẽ thực hiện kết nối với các node xung quanh sao cho dữ liệu truyền đi là tốt nhất.



Hình 2.2: Kiến trúc hệ thống

2.3 Kiến trúc node cảm biến



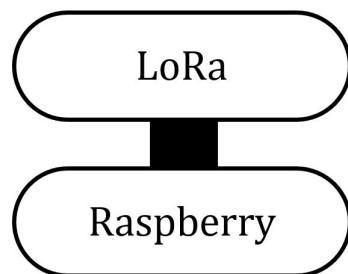
Hình 2.3: Kết nối với node cảm biến.

Bao gồm 4 module chính:

- WiFi - module giao tiếp không dây, sử dụng mạng wifi. Dùng để cập nhật firmware cho các node cảm biến.

- XBee - module giao tiếp không dây, sử dụng tín hiệu radio có tần sóng ngắn. Dùng để cập nhật firmware cho các node cảm biến.
- LoRa - module giao tiếp không dây, sử dụng tín hiệu sóng vô tuyến (radio) với dải băng tần không được cấp phép (433MHz). Dùng để gửi dữ liệu từ các node cảm biến đo đạc được bằng các sensor được gắn trên đó.
- Raspberry Pi 3 - Android Things, điều khiển các hoạt động của LoRa (gửi/nhận dữ liệu, quản lý các node cảm biến trong mạng lưới). Đồng thời xử lý dữ liệu nhận được theo chuẩn đã quy định trước khi đưa tất cả dữ liệu lên server.

2.4 Kiến trúc node gateway



Hình 2.4: Kết nối với node gateway.

Bao gồm 2 module chính:

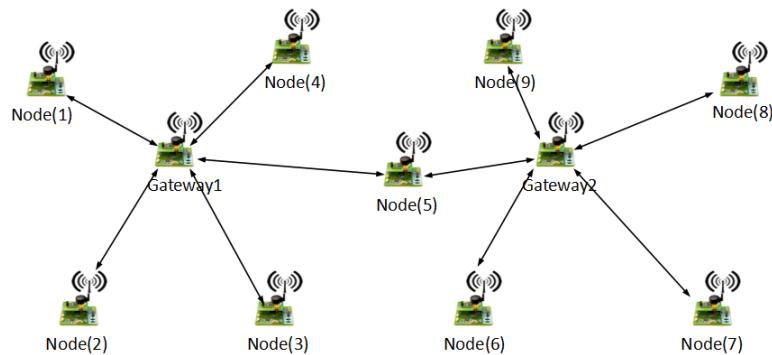
- LoRa - module giao tiếp không dây, sử dụng tín hiệu sóng vô tuyến (radio) với dải băng tần không được cấp phép (433MHz). Dùng để thu thập dữ liệu từ các node cảm biến đo đạc được bằng các sensor được gắn trên đó.
- Raspberry Pi 3 - Android Things, điều khiển các hoạt động của LoRa (gửi/nhận dữ liệu, quản lý các node cảm biến trong mạng lưới). Đồng

thời xử lý dữ liệu nhận được theo chuẩn đã quy định trước khi đưa tất cả dữ liệu lên server.

2.5 Mô hình giao tiếp không dây

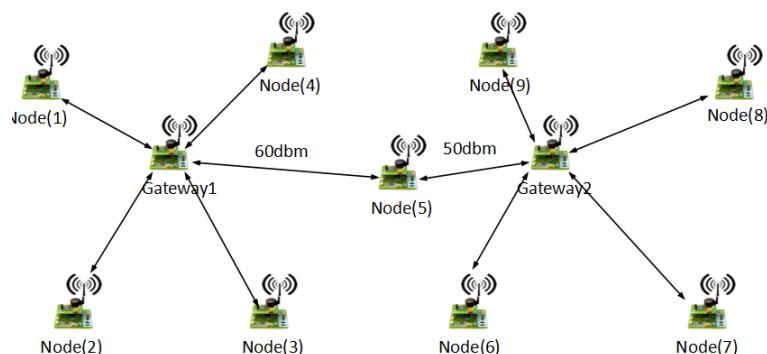
2.5.1 Giao tiếp ở gateway

1. Ban đầu, Gateway sẽ kết nối với tất cả các node có cường độ tín hiệu thu(RSSI) lớn hơn -70dbm.



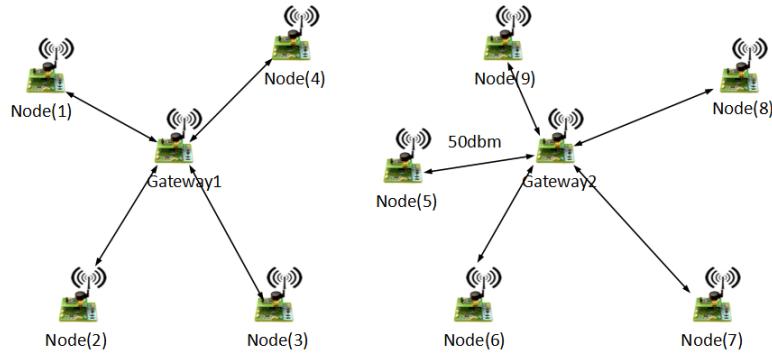
Hình 2.5: Gateway kết nối với các node ban đầu.

2. Giả sử Node có kết nối với 2 Gateway, thì nó sẽ thực hiện kết nối với node có cường độ tín hiệu thu lớn hơn.



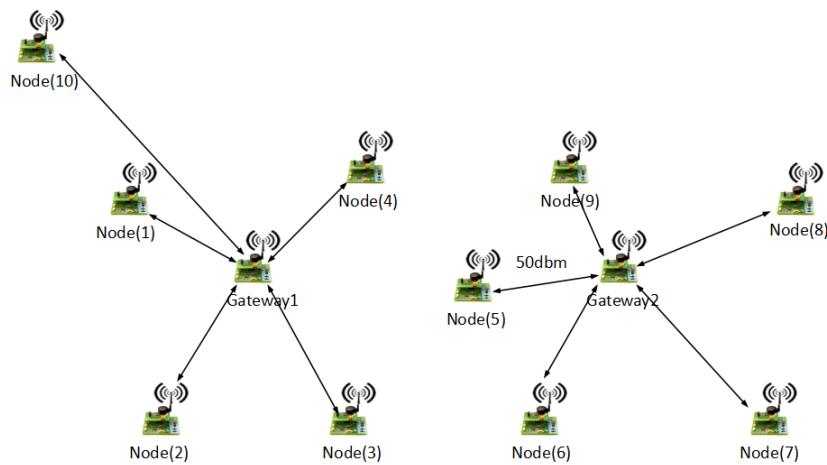
Hình 2.6: node(5) kết nối với hai Gateway.

3. với các node ở xa, cường độ tín hiệu thu thấp, nhung không kết nối được với node trung gian nào khác, thì bây giờ thực hiện kết nối với gateway.



Hình 2.7: node(5) ngắt kết nối với Gateway1.

4. Gateway sẽ cấp địa chỉ IP cho các node kết nối với nó. Và liên tục cập nhật các node mới, đồng thời kiểm tra xem có tất cả có bao nhiêu node kết nối với nó.



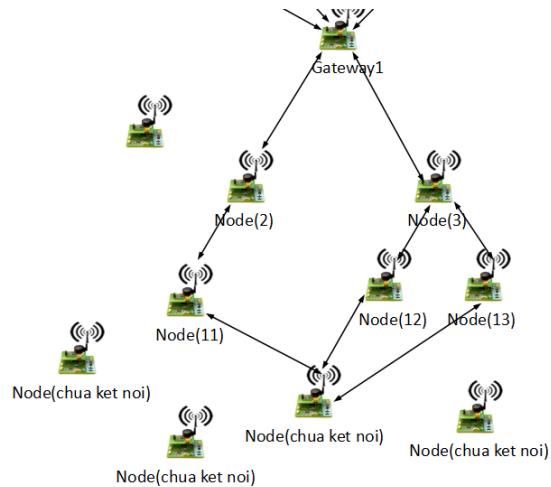
Hình 2.8: các node ở xa kết nối với Gateway.

2.5.2 Giao tiếp ở các node

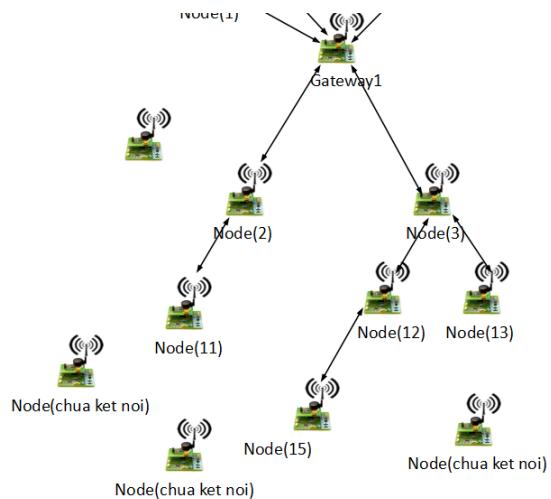
1. các node chưa có kết nối gửi lên Gateway, thì nó sẽ gửi dữ liệu đi đến các node khác, kiểm tra xem node đó đã kết nối hay chưa, rồi

Đề cương luận văn 2018

dựa vào các thông số như cường độ thu tín hiệu, số node trung gian cần phải qua để đến được gateway, số node hiện tại gateway đã kết nối, rồi quyết định kết nối với node phù hợp nhất.



Hình 2.9: Bắt tín hiệu các Gateway bên cạnh.



Hình 2.10: Kết nối với node phù hợp.

CHƯƠNG 3

GATEWAY IOTS DỰA TRÊN NỀN TẢNG ANDROID THINGS

3.1 Tại sao phải cần hệ điều hành?

"Để đạt được giá trị từ Internet of Things (IoT), việc cần phải có một nền tảng (platform) để tạo và quản lý ứng dụng, chạy các phân tích, lưu trữ và bảo mật dữ liệu của bạn là một điều cần thiết. Giống như một hệ điều hành dành cho máy tính xách tay, một nền tảng làm rất nhiều thứ đằng sau đó, giúp cho cuộc sống của các nhà phát triển, nhà quản lý và người dùng dễ dàng hơn và ít tốn kém hơn." [4]

3.1.1 Vậy nền tảng (platform) là gì?

"Nói chung, nền tảng (platform) là phần mềm và phần cứng, có thể bao gồm môi trường hoạt động, lưu trữ, sức mạnh máy tính, bảo mật, công cụ phát triển và nhiều chức năng phổ biến khác. Nền tảng được thiết kế để hỗ trợ nhiều chương trình ứng dụng nhỏ hơn mà thực sự giải quyết các vấn đề kinh doanh.

Nền tảng hữu ích vì chúng rút ra rất nhiều chức năng chung từ logic ứng dụng cụ thể. Ví dụ, bất kể bạn đang cố gắng viết một ứng dụng để tối ưu hóa mức tiêu hao nhiên liệu hoặc không gian lớp học, cơ bản bạn cần khá nhiều công nghệ giống nhau. Các nhà phát triển ứng dụng chỉ muốn tập trung vào vấn đề cụ thể mà họ đang giải quyết và sử dụng các khả năng

chung để tính toán sức mạnh hoặc lưu trữ hoặc bảo mật. Một nền tảng tốt làm giảm đáng kể chi phí phát triển và duy trì các ứng dụng.

Trong Internet of Things, các nền tảng được thiết kế để triển khai các ứng dụng giám sát, quản lý và kiểm soát các thiết bị được kết nối (hình bên dưới). Các nền tảng IoT phải xử lý các vấn đề như kết nối và trích xuất dữ liệu từ một số lượng khổng lồ các điểm cuối khác nhau, đôi khi ở các vị trí không thuận tiện với kết nối chập chờn." [4]

3.1.2 Vậy hệ điều hành là gì?

Hệ điều hành chính là một nền tảng phần mềm. Chi tiết hơn, nó là một môi trường hoạt động, được thiết kế để hỗ trợ cho các chương trình ứng dụng, hoặc tham chí là công cụ phát triển.[5]

Tuy nhiên, với hệ điều hành dành cho IoT sẽ khó sử dụng cho nhiều mục đích hoặc ứng dụng hàng loạt trên mọi sản phẩm, bởi vậy cần có nhiều hệ điều hành khác nhau trong lĩnh vực IoT để đáp ứng nhu cầu thực tế.[5]

3.1.3 Tại sao lại cần đến hệ điều hành?

Trong đề tài luận văn này, hệ điều hành IoT được lựa chọn phải ít phức tạp hơn, đòi hỏi khả năng xử lý dữ liệu có độ trễ thấp nhất có thể, nhưng vẫn có đầy đủ khả năng và đáp ứng được các yêu cầu về tiêu thụ năng lượng, không đòi hỏi nhiều về tài nguyên như bộ xử lý hay bộ nhớ RAM.[5]

Việc sử dụng hệ điều hành sẽ giúp cho việc cập nhật firmware một cách dễ dàng, thông qua mạng internet - wifi (hoặc công nghệ truyền không dây ZigBee), nếu trong mạng lưới có tới hàng trăm thiết bị đều cần phải cập nhật. Ngoài ra, hệ điều hành sẽ giúp tăng tính bảo mật, tăng sức mạnh, khả năng lưu trữ cho các thiết bị IoT. Đồng thời sẽ giúp quản lý được năng lượng sử dụng.

Vậy nên, nhóm đã quyết định sử dụng hệ điều hành Android Things trong đề tài này. Vì hiện nay, Android Things đang phát triển và nhận được sự hỗ trợ rất lớn của Google và cộng đồng mạng (cộng đồng phát triển trên nền tảng Android).

3.2 Giới thiệu hệ điều hành Android Things

"Android things là hệ điều hành được quản lý bởi Google dựa trên nền tảng Android cho phép bạn có thể xây dựng những ứng dụng IoT mà bạn không cần phải có quá nhiều kiến thức về hệ thống nhúng." [6]

"Về cơ bản, Android Things là một bản cập nhật và làm mới lại của Brillo, hệ điều hành dựa trên nền tảng Android dành cho các thiết bị thông minh và các sản phẩm Internet of Things (IoT) được giới thiệu vào năm 2015." [7]

"Android Things là một phiên bản rút gọn của Android có thể chạy trên những nguyên mẫu phần cứng khác nhau, để dễ dàng tạo ra thiết bị kết nối Internet of Things (IoT). Điều này làm cho mã nhúng có thể được truy cập đối với các nhà phát triển những người có thể không có kinh nghiệm trước đó. Với Android Things, Google cũng cung cấp một thư viện mà bạn có thể sử dụng để xây dựng các ứng dụng đọc từ và ghi vào các chân nối khác nhau trên bảng mạch, cho phép bạn đấu vào đó các cảm biến và bộ thiết bị điều tiết khác nhau để tương tác với thế giới." [8]

3.3 Tại sao nên sử dụng Android Things?

- Do Android Things là một phân mảng rộng của nền tảng Android nên bạn hoàn toàn có thể sử dụng các tool đã quen thuộc với Android Developer như Android Studio và Android SDK để phát triển chúng.
- Android Things OS được quản lý bởi Google nên nó an toàn.
- Được hỗ trợ trong hệ sinh thái phong phú của Google : Google Cloud, Tensor Flow, Play Services, Assistant SDK, Firebase...

- Bạn có thể sử dụng những ngôn ngữ lập trình High-Level như Java, Kotlin để xây dựng ứng dụng IoT.[6]

3.4 Các nền tảng nhúng hỗ trợ Android Things

"Tại thời điểm này, Android Things hỗ trợ ba nguyên mẫu phần cứng:

- Raspberry Pi 3 Model B (hoặc Raspberry Pi 2).
- Edison Intel cùng với bo mạch Arduino.
- NXP Pico i.MX6UL.

Mặc dù điều này có vẻ như là một sự hạn chế, nhưng một danh sách hạn chế các phần cứng cho phép Google hỗ trợ đầy đủ các nguyên mẫu bo mạch phổ biến và cung cấp cho các nhà phát triển một nền tảng vững chắc đã được thử nghiệm và kiểm chứng.

Ngoài ba bảng mạch đã đề cập, Android Things sẽ sớm hỗ trợ Intel Joule 570x và NXP Argon i.MX6UL, đem lại cho bạn nhiều tùy chọn phần cứng cho sự phát triển."^[8]

3.5 Xây dựng ứng dụng đầu tiên với Android Things

Để xây dựng ứng dụng với Android Things cần phải đáp ứng một số yêu cầu sau,

Về phần mềm:

- Môi trường Java.
- Ứng dụng lập trình Android (Android Studio).
- Và một số ứng dụng phụ khác khi mới bắt đầu sử dụng: Win32 Disk Imager/Etcher, SD Card Formatter, hoặc android-things-setup-utility.

- Trình duyệt web Google Chrome (*khuyên dùng*).

Về phần cứng:

- Raspberry Pi 3 Model B.
- Thẻ nhớ MicroSD (từ 8GB trở lên).
- Đầu đọc thẻ nhớ MicroSD.
- Nguồn 5V – 2.5A (*khuyên dùng*).
- Và một số thiết bị phần cứng khác khi mới bắt đầu sử dụng: màn hình HDMI/cảm ứng, cáp HDMI, chuột, bàn phím rời, cáp Ethernet (nếu không có sẵn mạng wifi).

3.5.1 Cài đặt JDK

JDK là một bộ công cụ phát triển Java, nó dành cho những người lập trình Java để phát triển ứng dụng. Về cơ bản nó bao gồm:

1. JRE (Java Runtime Environment) là một môi trường chạy ứng dụng Java.
2. Javac: Một chương trình để dịch mã mà bạn viết thành mã bytecode, khi ứng dụng Java chạy nó dịch mã bytecode thành mã máy tính và thực thi, điều đó có nghĩa là bytecode chỉ là một mã trung gian.
3. Archive (jar): Là một chương trình nén các file thành một file duy nhất có đuôi jar. Thường dùng để đóng gói các file class.
4. Javadoc: Là một công cụ tạo ra tài liệu hướng dẫn sử dụng API.
5. Và các công cụ khác cần thiết cho phát triển Java.

Để tiến hành cài đặt JDK chúng ta thực hiện các bước sau.

1. Vào đường dẫn <https://www.oracle.com/technetwork/java/javase/downloads/index.html> để download JDK

Đề cương luận văn 2018



Hình 3.1: Oracle JDK

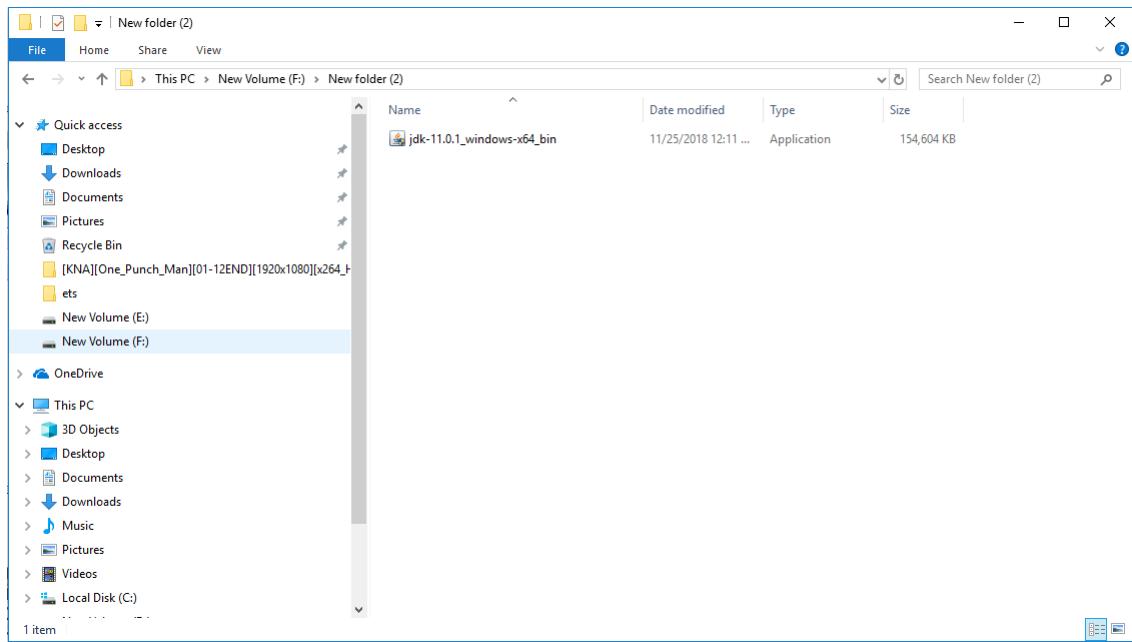
2. nhấn vào Oracle JDK để chọn JDK cần download
3. nhấn chọn Accept License Agreement và sau đó chọn phiên bản phù hợp với hệ điều hành mà bạn đang sử dụng

Java SE Development Kit 11.0.1		
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux	147.4 MB	jdk-11.0.1_linux-x64_bin.deb
Linux	154.09 MB	jdk-11.0.1_linux-x64_bin.rpm
Linux	171.43 MB	jdk-11.0.1_linux-x64_bin.tar.gz
macOS	166.2 MB	jdk-11.0.1_osx-x64_bin.dmg
macOS	166.55 MB	jdk-11.0.1_osx-x64_bin.tar.gz
Solaris SPARC	186.8 MB	jdk-11.0.1_solaris-sparcv9_bin.tar.gz
Windows	150.98 MB	jdk-11.0.1_windows-x64_bin.exe
Windows	170.99 MB	jdk-11.0.1_windows-x64_bin.zip

Hình 3.2: Chọn Phiên bản JDK

Đề cương luận văn 2018

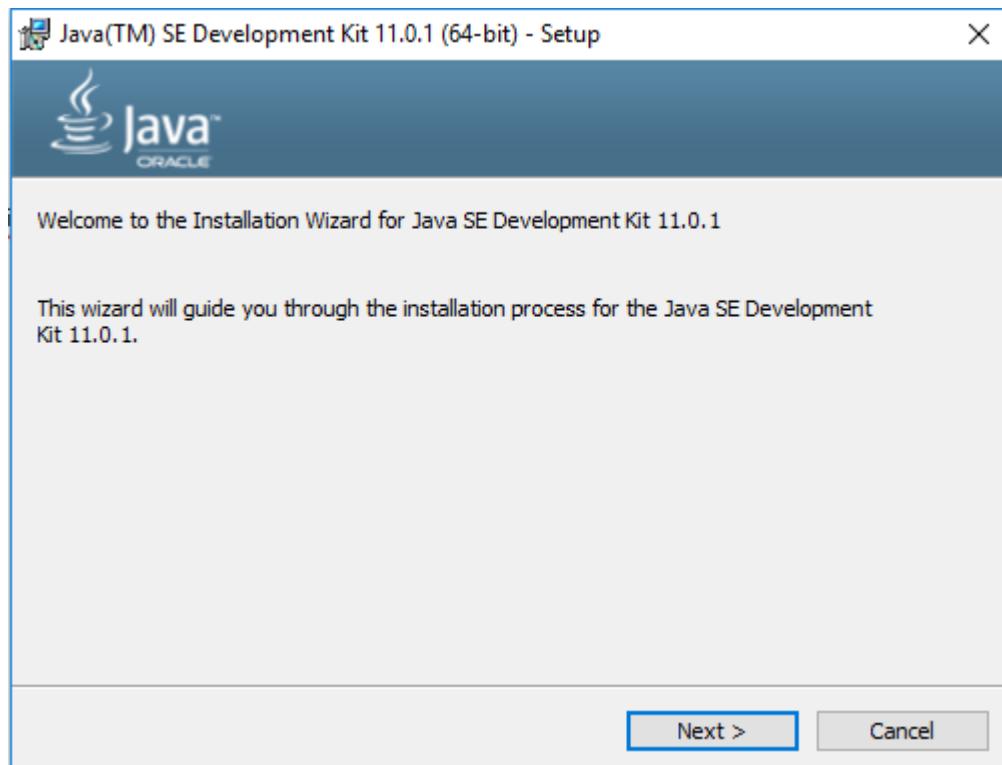
4. Kết quả download được:



Hình 3.3: File JDK tải về

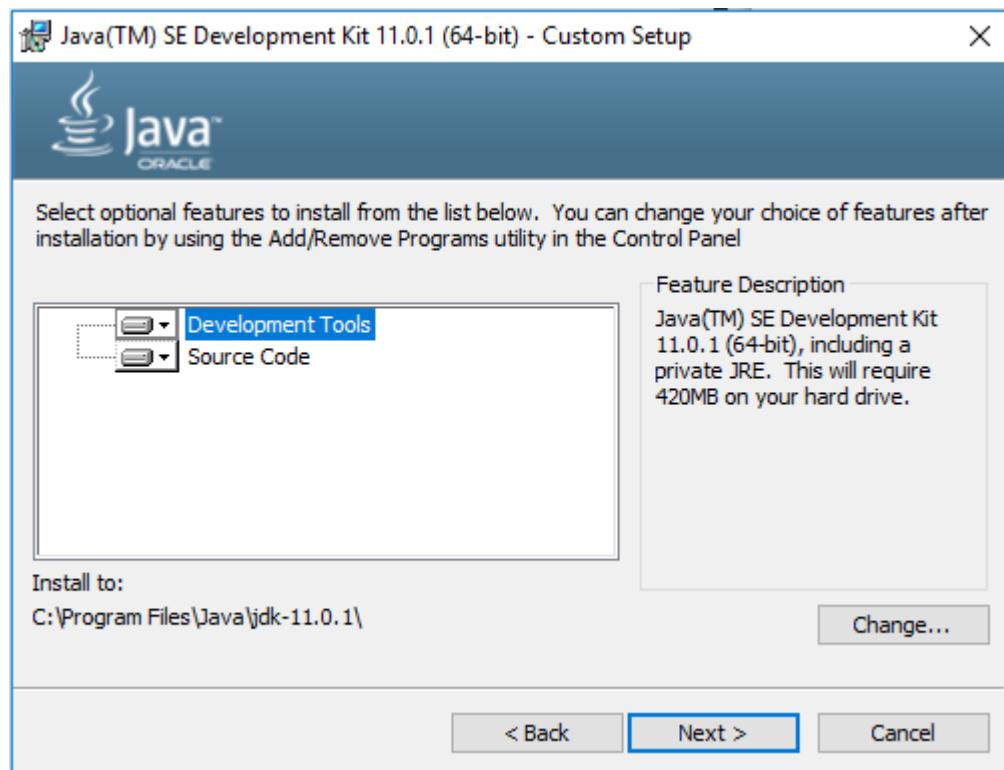
5. chạy file vừa download về, nhấn next để tiếp tục.

Đề cương luận văn 2018



Hình 3.4: Chạy File JDK tải về

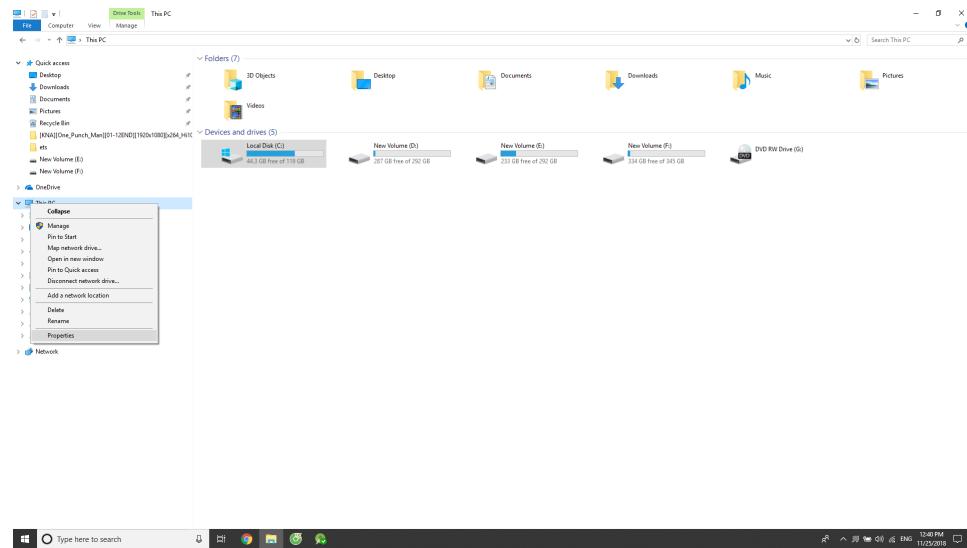
6. Nhập vào thư mục mà JDK sẽ được cài đặt ra, ở đây tôi đặt là:
`C:\Program File\Java\jdk-11.0.1\`



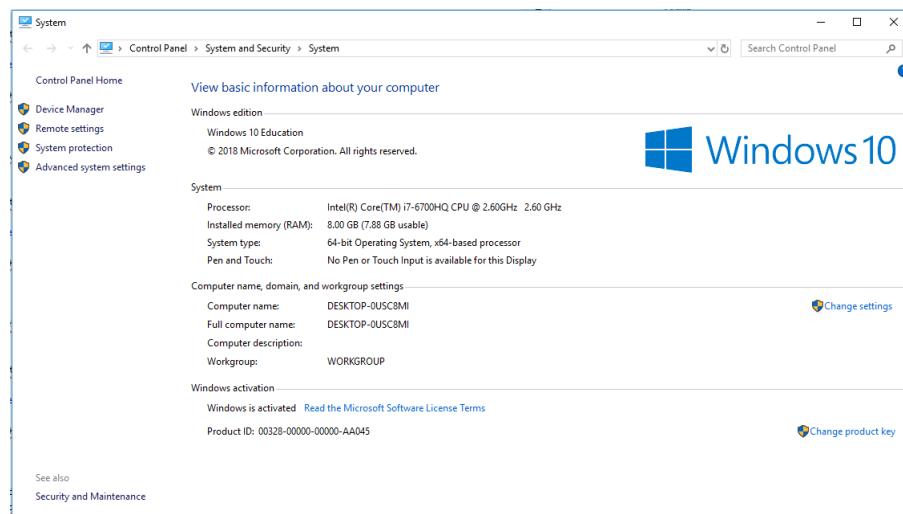
Hình 3.5: Cài đặt đường dẫn file JDK tải về

7. Nhấn next để tiếp tục, sau khi cài đặt xong nhấn close.
8. Nhấn phải chuột vào Computer, chọn Properties. Sau đó nhấn Advance System Settings. Chọn tiếp Advanced. Chọn Environment variables.

Đề cương luận văn 2018

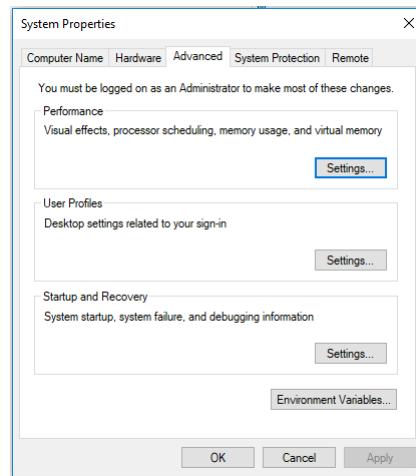


Hình 3.6: Chọn Properties.



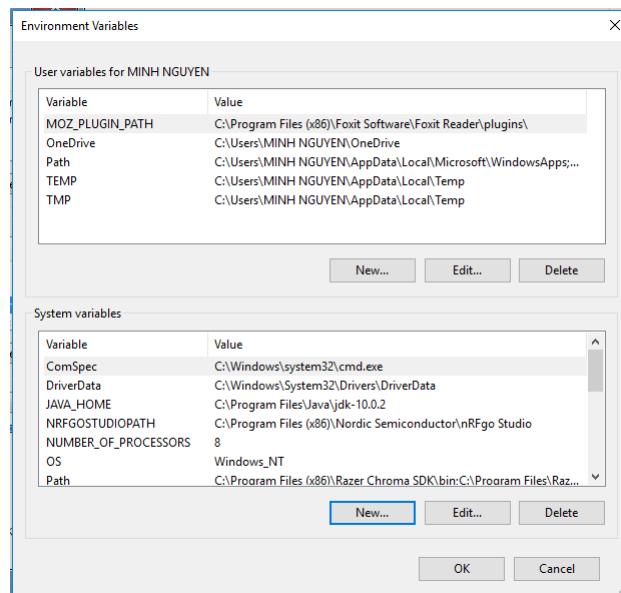
Hình 3.7: Chọn Advanced System Settings.

Đề cương luận văn 2018



Hình 3.8: Chọn Advanced.

9. Nhấn New để tạo mới một biến môi trường có tên "JAVA_HOME".



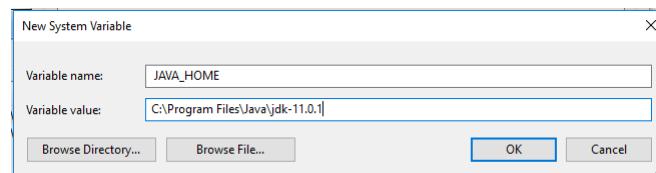
Hình 3.9: Tạo một biến môi trường có tên "JAVA_HOME".

Đề cương luận văn 2018

10. Nhập vào đường dẫn tới thư mục JDK.

Variable name: JAVA_HOME

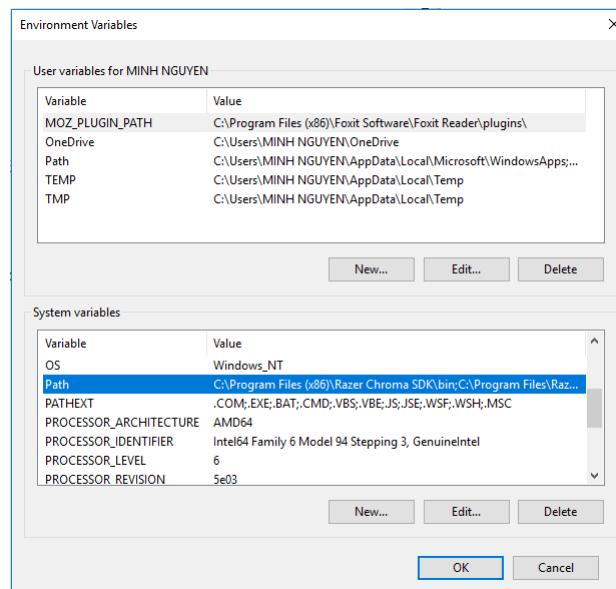
Variable value: C:/Program Files/Java/jdk-11.0.1



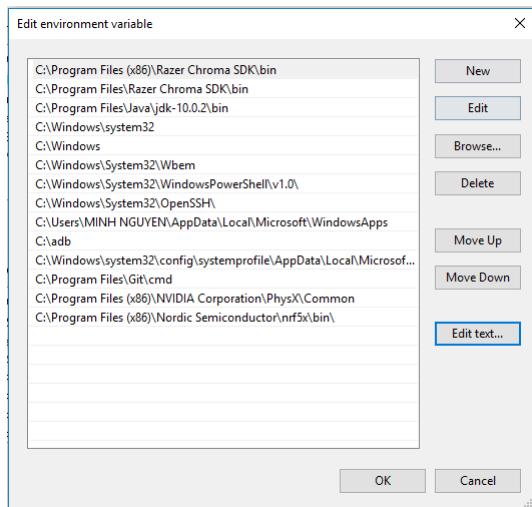
Hình 3.10: Nhập Variable name và Variable value.

11. Tiếp theo sửa đổi biến môi trường path

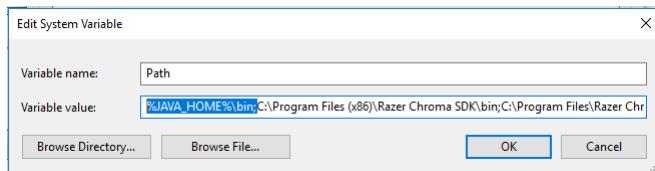
Nhấn vào Edit, sau đó nhấn vào Edit text. Thêm vào phía trước giá trị của biến môi trường path: %JAVA_HOME% /bin;



Hình 3.11: Nhấn vào Edit.



Hình 3.12: Nhấn vào Edit text.



Hình 3.13: Hoàn tất thêm vào "%JAVA_HOME%/bin".

3.5.2 Cài đặt Android Studio

Có nhiều công cụ để phát triển Android nhưng đến nay công cụ chính thức và mạnh mẽ nhất là Android Studio. Đây là IDE (Môi trường phát triển tích hợp) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng mà bạn có thể sử dụng hàng ngày.

Android Studio lần đầu tiên được công bố tại hội nghị Google I/O vào năm 2013 và được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau. Trước khi được phát hành, các nhà phát triển Android thường sử dụng các công cụ như Eclipse IDE, một IDE Java chung cũng hỗ trợ nhiều ngôn ngữ lập trình khác.

Android Studio khiến việc tạo ứng dụng trở nên dễ dàng hơn đáng kể so với phần mềm không chuyên dụng. Đối với người mới bắt đầu, có rất nhiều

thứ để học và nhiều thông tin có sẵn, thậm chí thông qua các kênh chính thức nhưng chúng có thể đã lỗi thời hoặc quá nhiều thông tin khiến họ cảm thấy choáng ngợp.

Chức năng của Android Studio là cung cấp giao diện để tạo các ứng dụng và xử lý phần lớn các công cụ quản lý file phức tạp đằng sau hậu trường. Ngôn ngữ lập trình được sử dụng ở đây là Java và được cài đặt riêng trên thiết bị của bạn. Android Studio rất đơn giản, bạn chỉ cần viết, chỉnh sửa và lưu các dự án của mình và các file trong dự án đó. Đồng thời, Android Studio sẽ cấp quyền truy cập vào Android SDK.

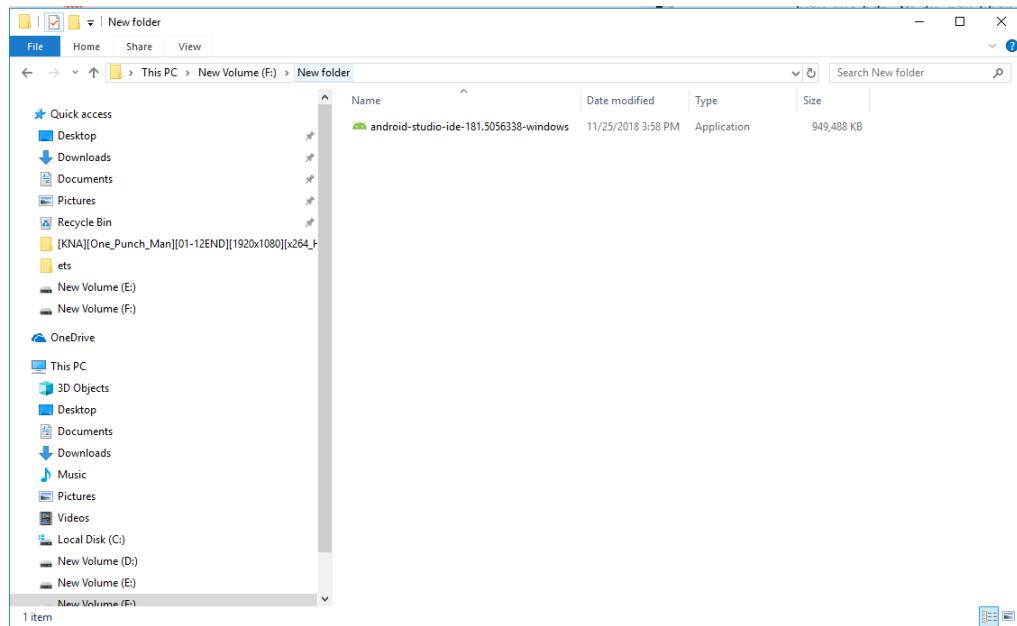
Hãy coi đây là đuôi cho code Java cho phép nó chạy trơn tru trên các thiết bị Android và tận dụng lợi thế của phần cứng gốc. Bạn cần sử dụng ngôn ngữ lập trình Java để viết các chương trình, Android SDK có nhiệm vụ kết nối các phần này lại với nhau. Cùng lúc đó Android Studio kích hoạt để chạy code, thông qua trình giả lập hoặc qua một phần cứng kết nối với thiết bị. Sau đó, bạn cũng có thể “gõ rồi” chương trình khi nó chạy và nhận phản hồi giải thích sự cố, v.v... để bạn có thể nhanh chóng giải quyết vấn đề.

Google đã nỗ lực rất nhiều để làm cho Android Studio trở nên mạnh mẽ và hữu ích nhất có thể. Nó cung cấp những gợi ý trực tiếp trong khi viết code và thường đề xuất những thay đổi cần thiết để sửa lỗi hoặc làm code hiệu quả hơn. Ví dụ, nếu không sử dụng biến, biến đó sẽ được tô đậm bằng màu xám. Và khi bắt đầu gõ một dòng code, Android Studio sẽ cung cấp danh sách gợi ý tự hoàn thành để giúp bạn hoàn thiện dòng code đó. Chức năng này rất hữu ích khi bạn không nhớ được chính xác cú pháp hoặc để tiết kiệm thời gian[3].

Hướng dẫn cài đặt android studio.

1. Vào trang web <https://developer.android.com/studio/> để download android studio.

Đề cương luận văn 2018



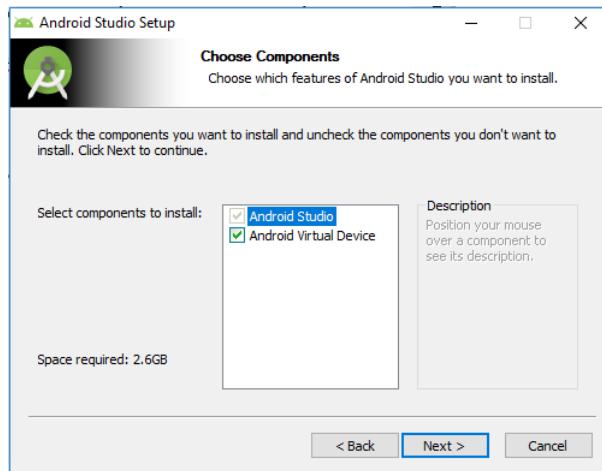
Hình 3.14: File android studio.

2. Chạy file android studio tải về

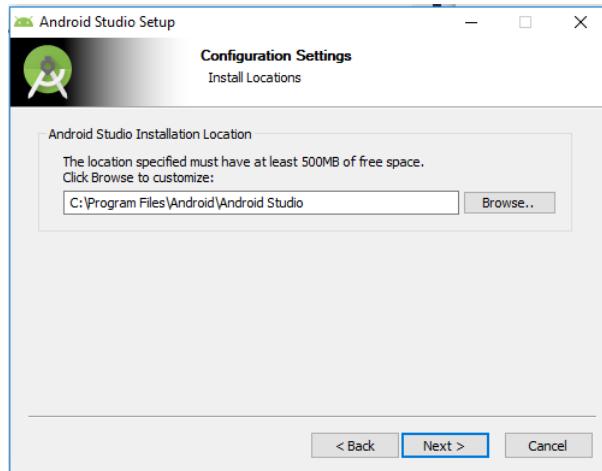


Hình 3.15: Nhấn Next để cài đặt.

Đề cương luận văn 2018

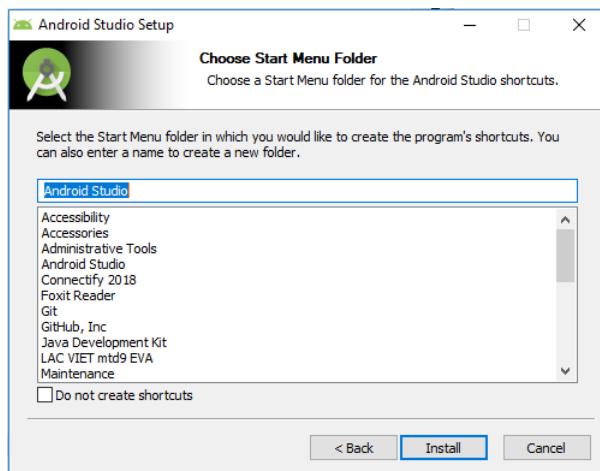


Hình 3.16: Nhấn Next để cài đặt.

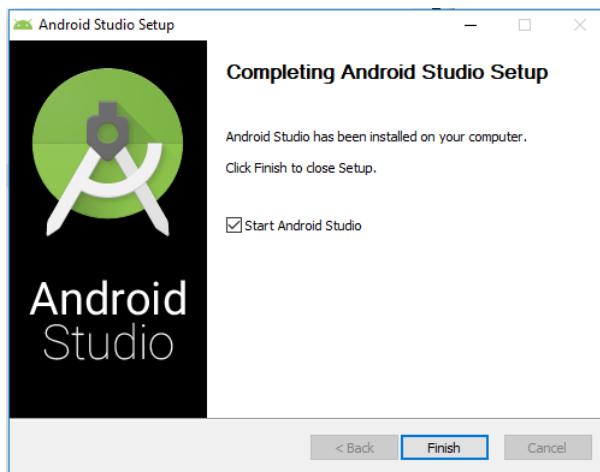


Hình 3.17: Chọn ví trí lưu, nhấn Next để cài đặt.

Đề cương luận văn 2018



Hình 3.18: Nhấn Install để cài đặt.



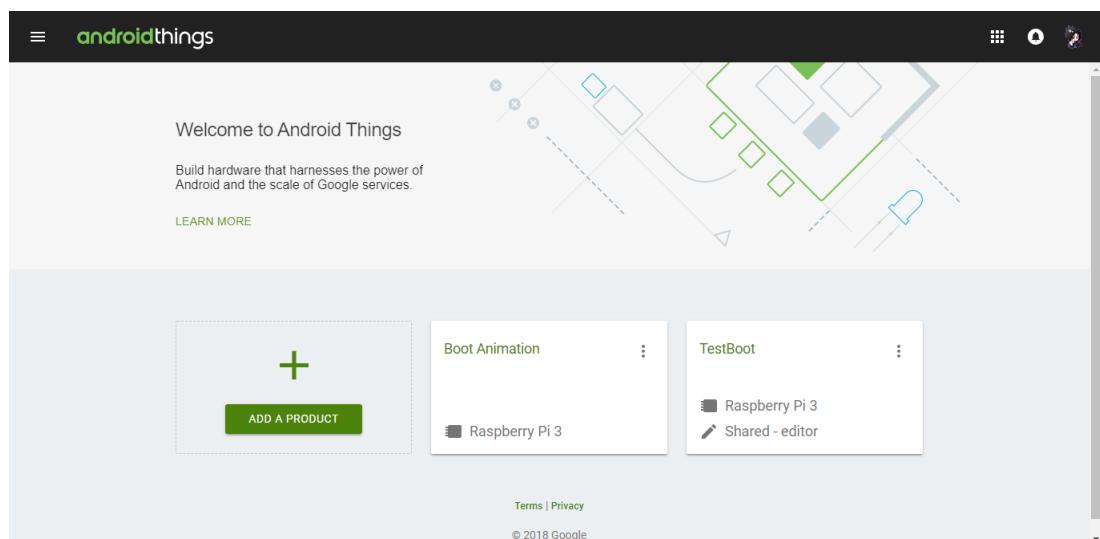
Hình 3.19: Nhấn Finish để cài đặt.

3.5.3 Cài đặt hệ điều hành Android Things

Để cài hệ điều hành Android Things, cần thực hiện theo 3 bước sau.

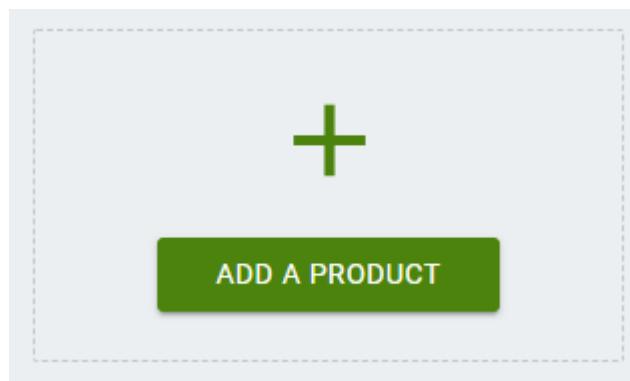
Bước 1: Tạo Image Boot cho hệ điều hành Android Things.

1. Vào trang <https://partner.android.com/things/console/>, đăng nhập với tài khoản của Google. Nên sử dụng trình duyệt Google Chrome để tránh trường hợp không vào được trang



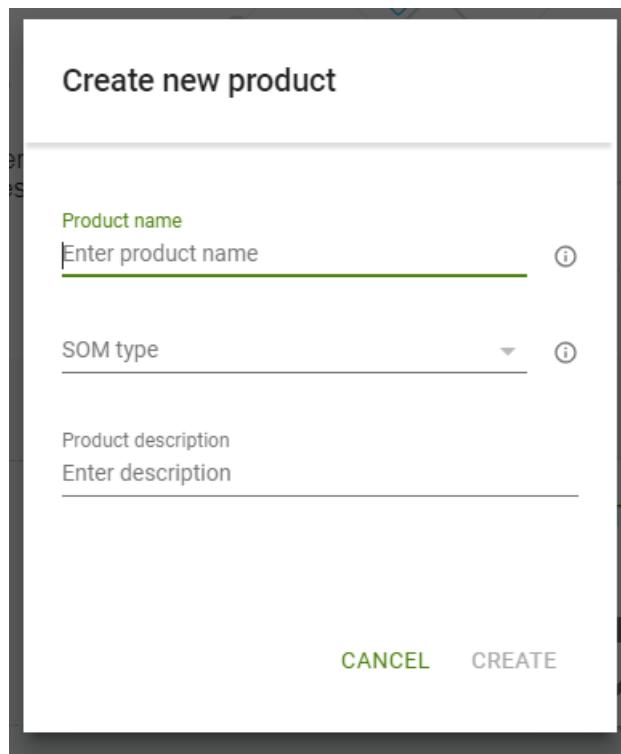
Hình 3.20: Màn hình giao diện - tạo Image Boot.

2. Nhấn vào "ADD A PRODUCT" hoặc biểu tượng dấu cộng "+".



Hình 3.21: Nhấn + để tạo PRODUCT mới.

Đề cương luận văn 2018



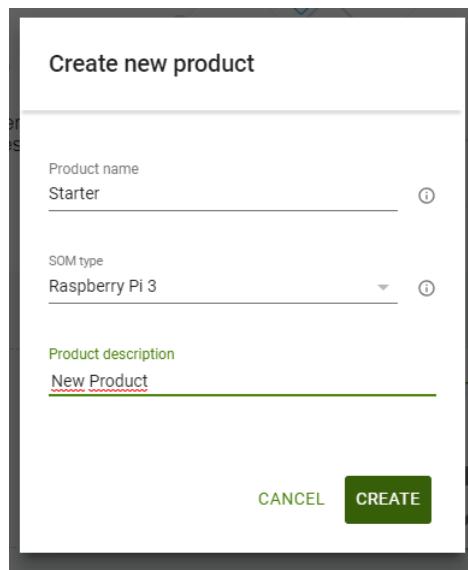
Hình 3.22: Thêm thông tin PRODUCT vào bảng.

3. Bảng "Create new product" hiện lên. Điền thông tin vào các trường:

- "Product name" - là tên sản phẩm mà mình sẽ làm việc, yêu cầu có ít nhất một ký tự chữ, ví dụ "Starter".
- "SOM type" - là mạch phần cứng được sử dụng, ở đây dùng Rpi3 nên chọn "Raspberry Pi 3".
- "Product description" là mô tả về sản phẩm, trường này không bắt buộc phải điền

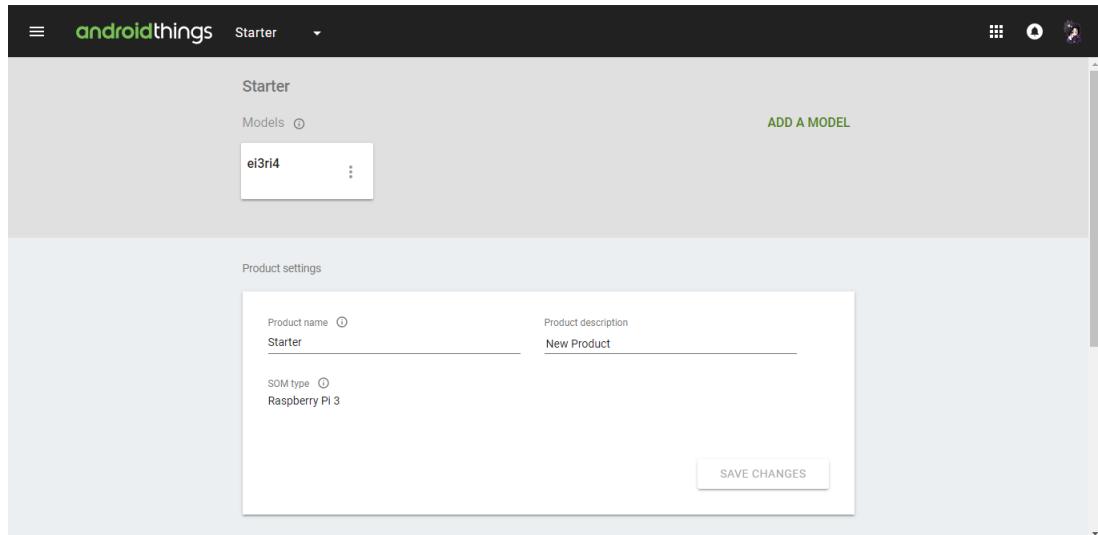
4. Sau khi điền xong, nhấn "Create" để tạo sản phẩm.

Đề cương luận văn 2018



Hình 3.23: Nhấn CREATE để tạo sản phẩm.

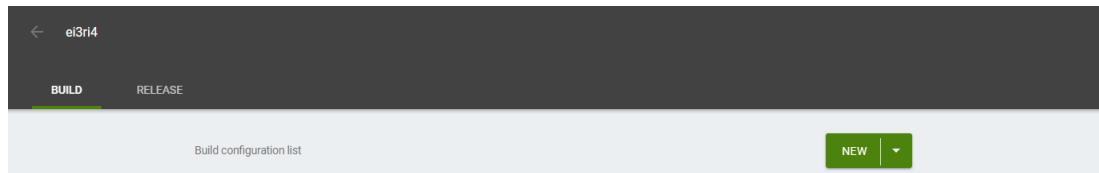
5. Sau khi tạo xong, màn hình sẽ hiện ra kết quả khi tạo thành công, ấn vô ô ("ei3ri4" - theo ví dụ) ở phần "Models".



Hình 3.24: Tạo thành công sản phẩm.

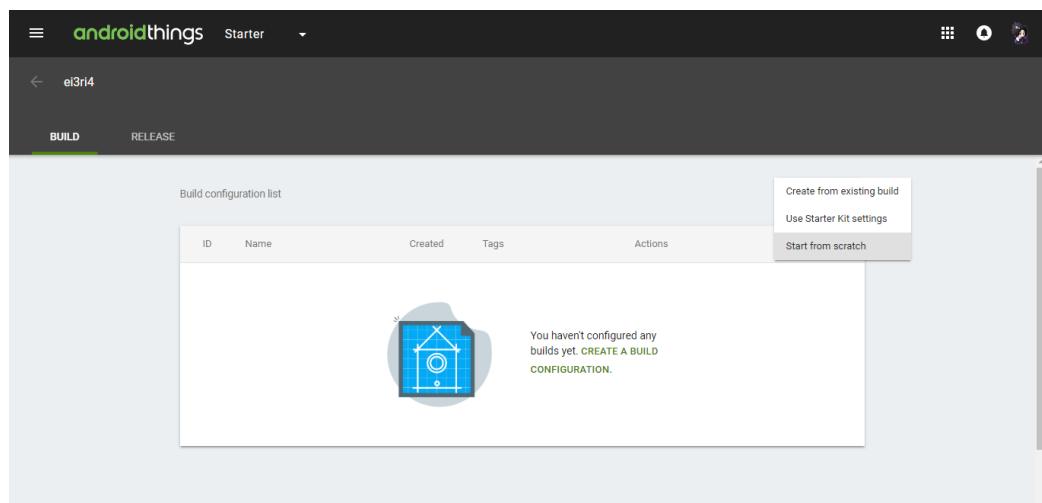
6. Tại tab "BUILD", ấn vào nút "NEW" ở gần bên phải giữa màn hình để tạo mới một Image Boot.

Đề cương luận văn 2018



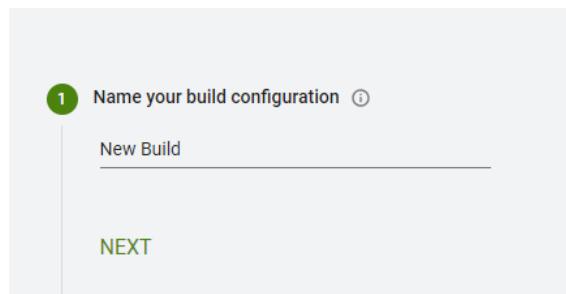
Hình 3.25: Ấn NEW để tạo mới Image Boot.

7. Tại khung xổ xuống, chọn "Start from scratch".



Hình 3.26: Chọn Start from scratch.

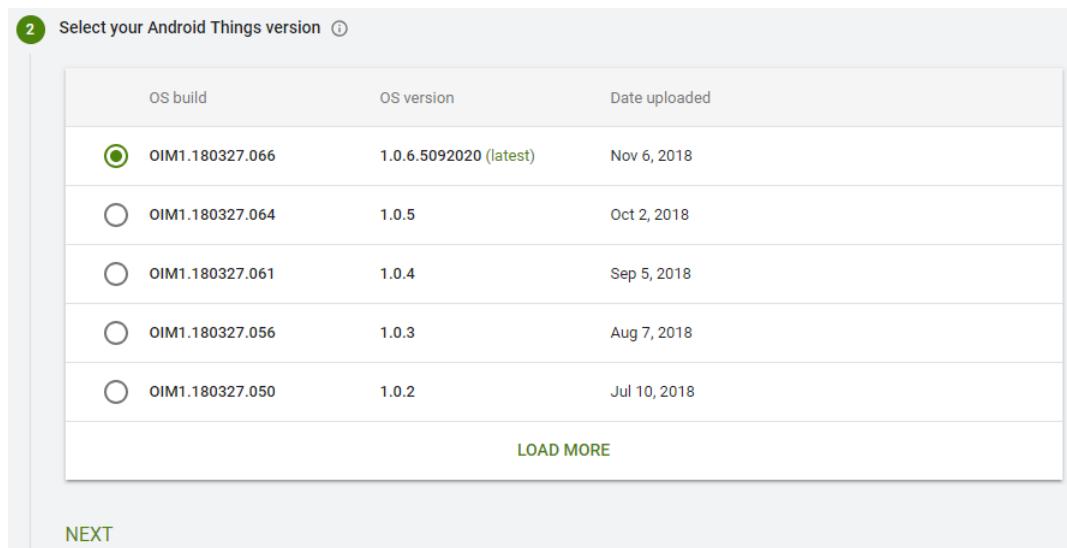
8. Tại trường "1", nhập tên Image Boot muốn tạo, ví dụ "New Build". Sau đó ấn "NEXT".



Hình 3.27: Nhập tên Image Boot.

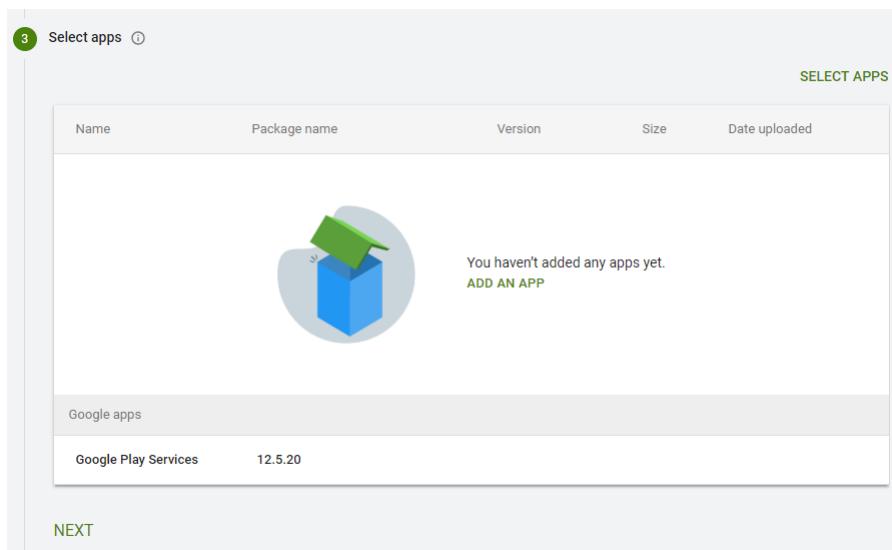
9. Tại trường "2", chọn phiên bản Android Things mới nhất, thường có chữ "(latest)" ở cột "OS version". Sau đó ấn "NEXT".

Đề cương luận văn 2018



Hình 3.28: Chọn phiên bản Android Things mới nhất.

10. Trường "3" là nơi để upload ứng dụng được chạy tự động đầu tiên sau khi hệ điều hành được khởi động. Nếu muốn thiết lập cho ứng dụng tự khởi chạy thì ấn chọn "SELECT APPS". Nhưng vẫn khuyên không nên chọn và ấn "NEXT".

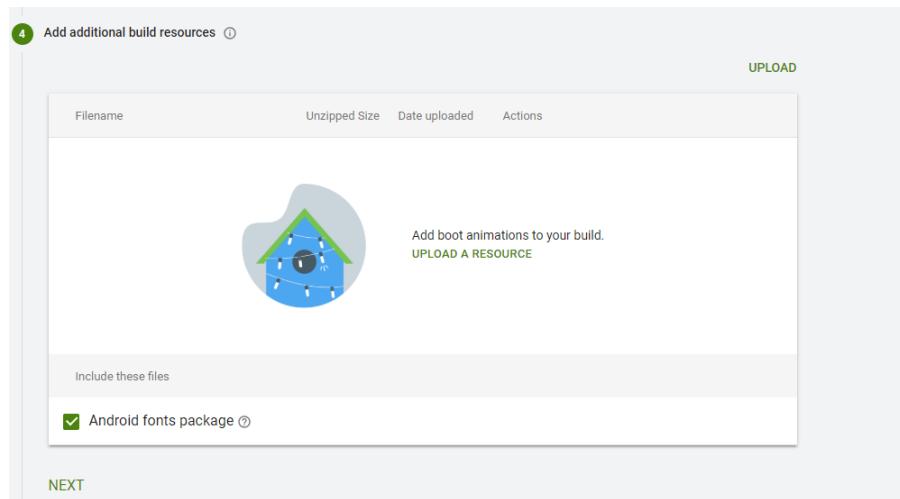


Hình 3.29: Chọn NEXT tại bước này.

11. Trường "4" sẽ là nơi để cài đặt hình ảnh load đầu tiên khi hệ điều hành được khởi động. Có thể thay đổi hình ảnh này, điều này sẽ được

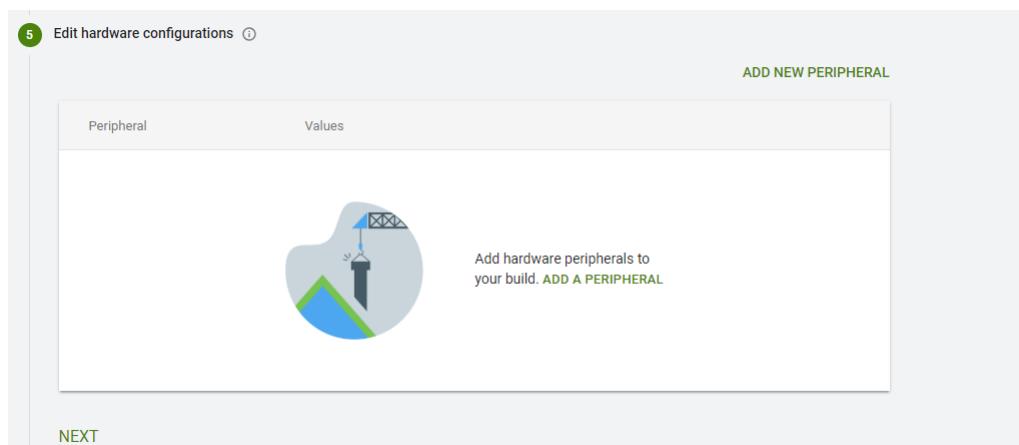
Đề cương luận văn 2018

hướng dẫn thêm ở mục 3.5.4.1. Sau đó ấn "NEXT".



Hình 3.30: Chọn NEXT tại bước này - Xem thêm 3.5.4.1.

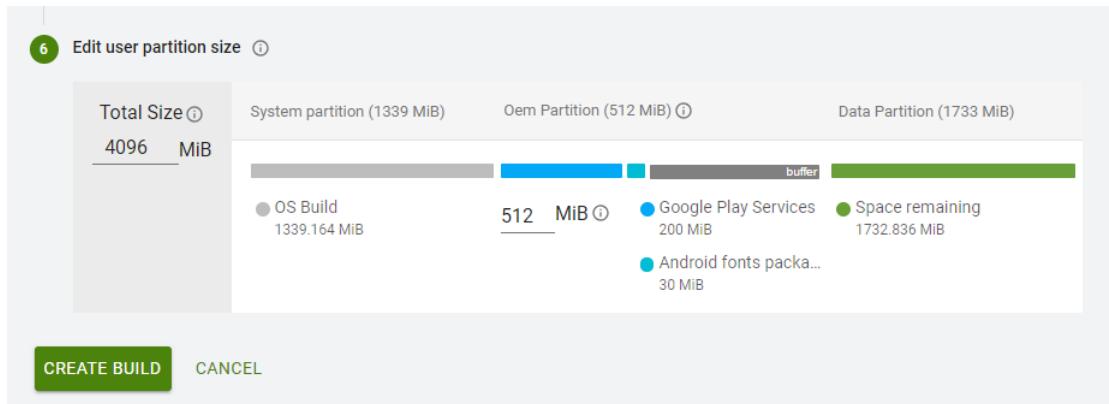
12. Trường "5" là phần chỉnh sửa thông tin về chân cắm của phần cứng. KHÔNG NÊN CAN THIỆP VÀO PHẦN NÀY, chỉ nên can thiệp nếu đã đủ khả năng làm việc với mạch. Sau đó ấn "NEXT".



Hình 3.31: Chọn NEXT tại bước này.

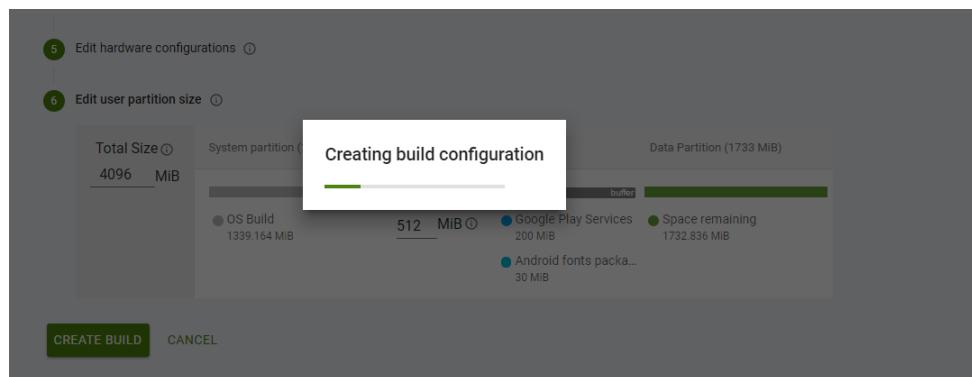
13. Tại trường "6" sẽ thống kê lại những thiết lập đã chọn. Ấn "CREATE BUILD" để tạo Image Boot.

Đề cương luận văn 2018



Hình 3.32: Thông tin về bản Image Boot.

14. Vậy là đã tạo xong một bản Image Boot rồi.



Hình 3.33: Ấn CREATE BUILD để tạo.

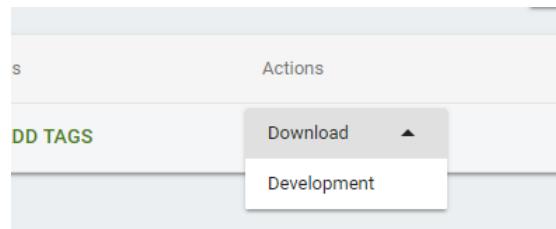
15. Để tải bản Image Boot, tại cột "Actions", ấn vô "Download".

ID	Name	Created	Tags	Actions
1	New Build	Nov 26, 2018	ADD TAGS	Download

Hình 3.34: Ấn Download tại bản Image Boot muốn tải.

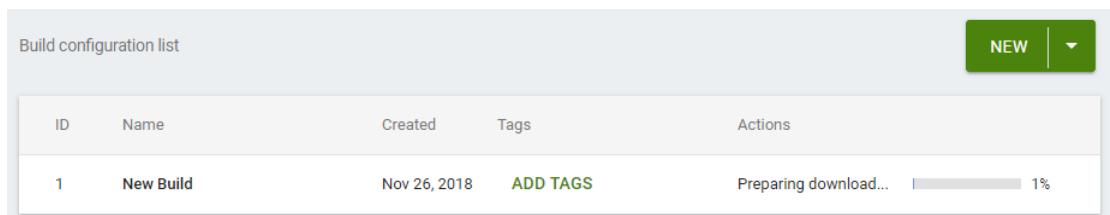
Đề cương luận văn 2018

16. Tiếp tục ấn vô "Development" để thực hiện quá trình tải.



Hình 3.35: Chọn Development.

17. Đợi từ 5 đến 10 phút là tải xong (tùy thuộc vào tốc độ mạng).



Hình 3.36: Chờ để tải xuống.

18. File Image Boot đã tải xong, dung lượng tầm khoảng 300 - 350 (MB), định dạng file là ".zip".

jquery-star-rating-master.zip	10/27/18 2:30 AM	WinRAR ZIP archive	0 KB
Starter_Raspberry Pi 3_1_New Build_development_build.zip	23/9/18 1:26 PM	WinRAR ZIP archive	15 KB
test (1).jpg	26/11/18 4:07 AM	WinRAR ZIP archive	323,365 KB
test.jpg	25/11/18 7:57 PM	JPG File	232 KB
test.jpg	25/11/18 7:51 PM	JPG File	232 KB

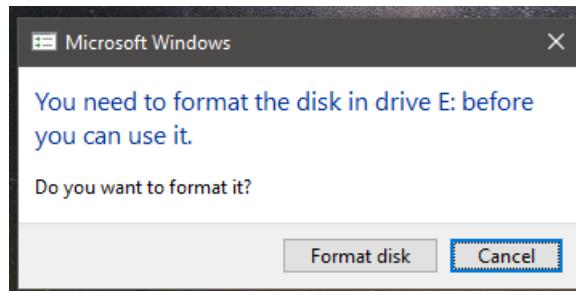
Hình 3.37: Đã tải thành công.

Bước 2: Cài đặt hệ điều hành lên thẻ nhớ (dung lượng từ 8GB trở lên).

Cách 1: Sử dụng phần mềm SD Card Formatter, Win32 Disk Image.

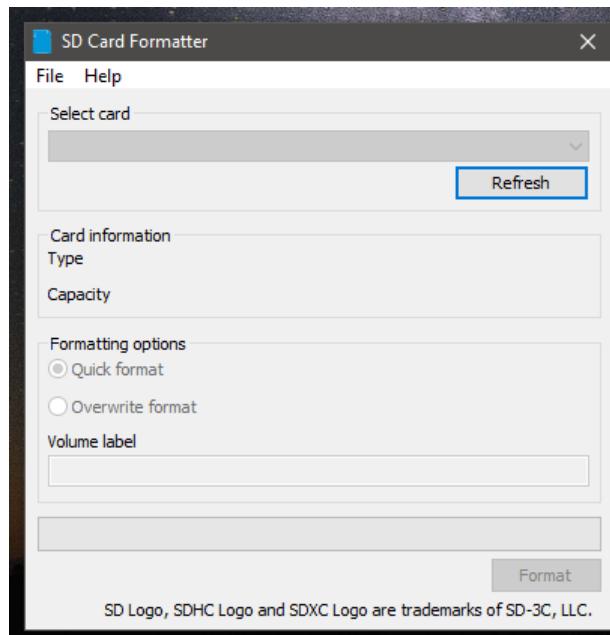
Đề cương luận văn 2018

1. Cắm thẻ nhớ MicroSD vào đầu đọc thẻ, sau đó cắm vào máy tính. Sẽ hiện ra thông báo yêu cầu định dạng lại thẻ nhớ, nhấn "Cancel".



Hình 3.38: Nhấn Cancel sau khi cắm thẻ nhớ vô máy.

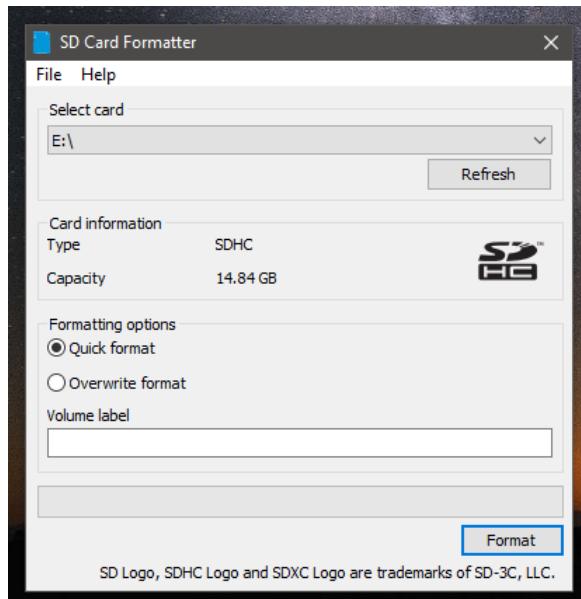
2. Chạy phần mềm SD Card Formatter để định dạng thẻ nhớ. Lưu ý, phải chọn đúng ổ đĩa của thẻ nhớ để tránh trường hợp định dạng nhầm ổ cứng máy tính. Án "Refresh" để tự động tìm thẻ nhớ đã được kết nối.



Hình 3.39: Nhấn Refresh để tự động tìm thẻ nhớ.

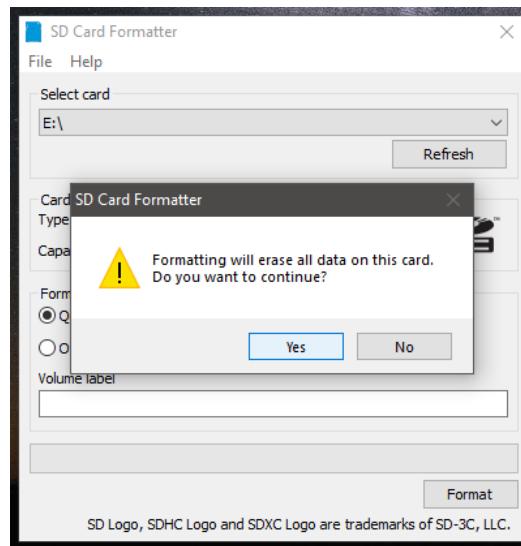
3. Sau khi tìm được, chọn thẻ nhớ dùng để cài hệ điều hành. Nhấn "Format" để định dạng.

Đề cương luận văn 2018



Hình 3.40: Nhấn Format để định dạng.

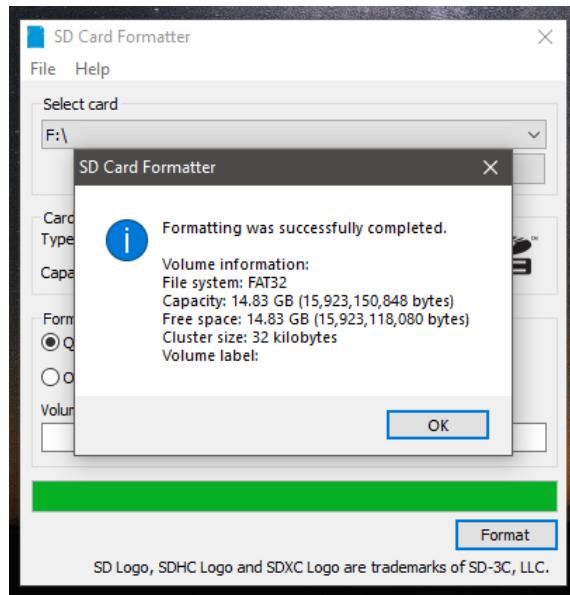
4. Thông báo nhắc nhở sẽ hiện lên, nhấn "Yes" để thực hiện quá trình định dạng thẻ nhớ.



Hình 3.41: Nhấn Yes.

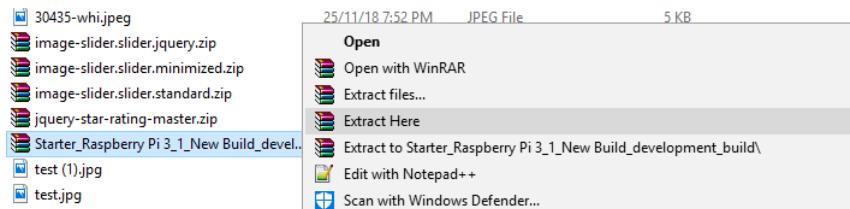
5. Định dạng xong, phần mềm sẽ hiển thị như hình 3.42. Nhấn "OK" và thoát.

Đề cương luận văn 2018



Hình 3.42: Nhấn OK.

6. Giải nén tập tin Image Boot vừa tải xong. Sau khi giải nén xong sẽ có một tập tin "iot_rpi3.img".



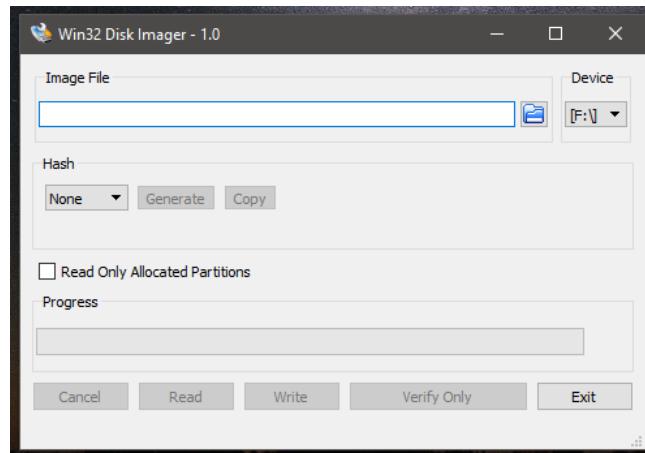
Hình 3.43: Giải nén tập tin Image Boot vừa tải.

30435-whi.jpeg	25/11/18 7:52 PM	JPEG File	5 KB
image-slider.slider.jquery.zip			
image-slider.slider.minimized.zip			
image-slider.slider.standard.zip			
jquery-star-rating-master.zip			
Starter_Raspberry Pi 3_1_New Build_devel..zip	26/11/18 4:07 AM	WinRAR ZIP archive	323,365 KB
test (1).jpg	25/11/18 7:57 PM	JPG File	232 KB
test.jpg	25/11/18 7:51 PM	JPG File	232 KB
iot_rpi3.img	25/11/18 1:03 PM	Disc Image File	4,194,304 KB

Hình 3.44: Nhận được tập tin "iot_rpi3.img".

7. Chạy phần mềm Win32 Disk Image để cài hệ điều hành lên thẻ nhớ.

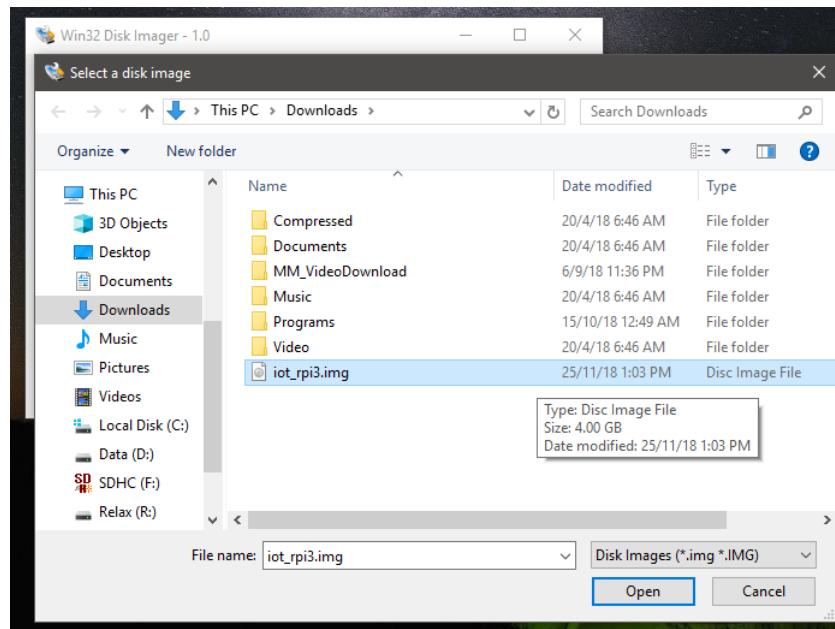
Đề cương luận văn 2018



Hình 3.45: Chạy phần mềm Win32 Disk Image.

8. Chọn đường dẫn tới tập tin "iot_rpi3.img" vừa được giải nén xong.

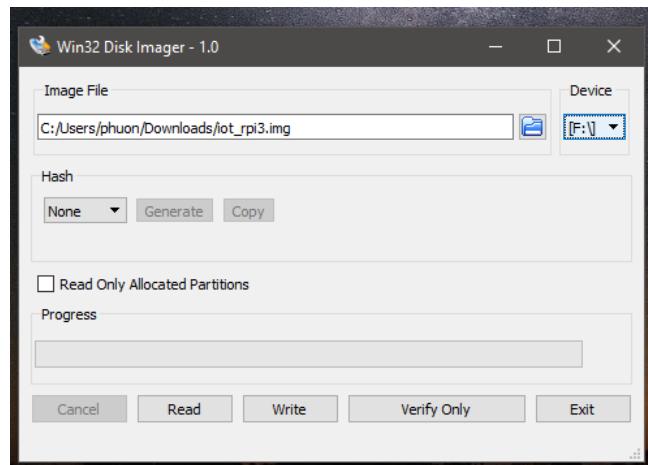
Ví dụ: C:/Download/



Hình 3.46: Chọn đường dẫn tới tập tin Image Boot đã được giải nén.

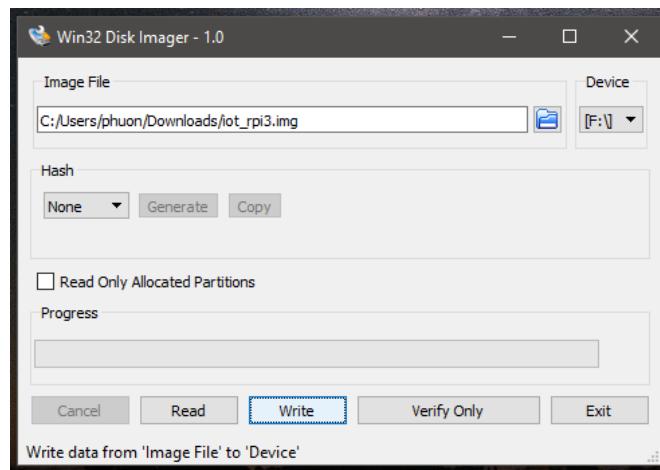
Đề cương luận văn 2018

9. Chọn thẻ nhớ đã được định dạng.



Hình 3.47: Chọn thẻ nhớ.

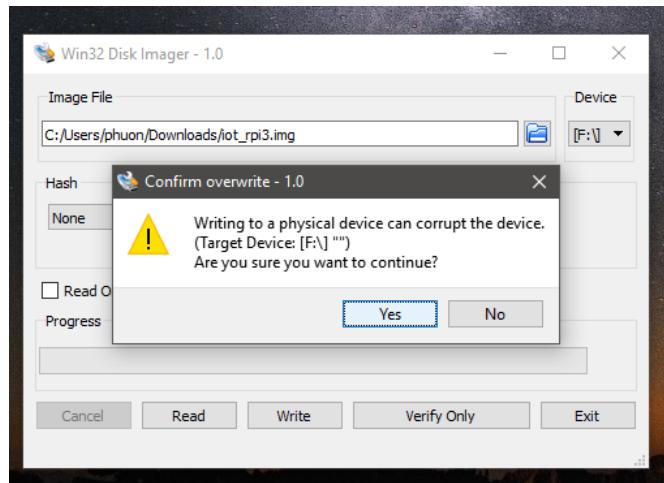
10. Ấn "Write" để ghi hệ điều hành Android Things vào thẻ nhớ.



Hình 3.48: Chọn thẻ nhớ.

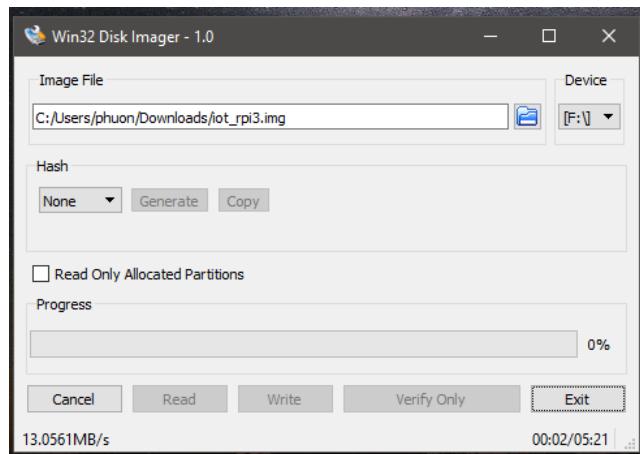
11. Chương trình sẽ hiện ra tin nhắn cảnh báo, ấn "Yes" để tiếp tục quá trình.

Đề cương luận văn 2018



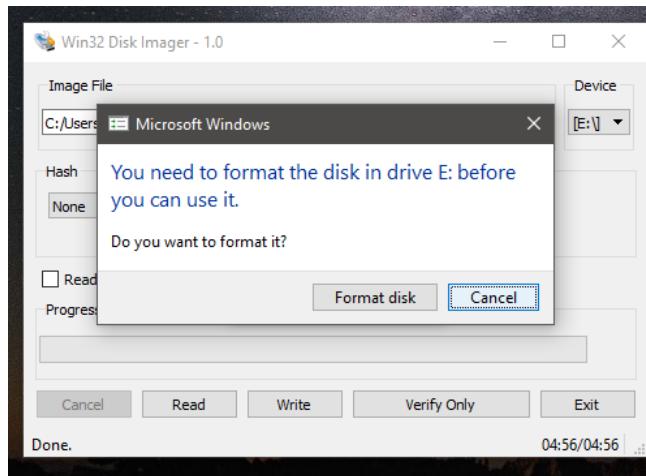
Hình 3.49: Nhấn Yes để tiếp tục.

12. Đợi từ 2 - 5 phút (tùy vào máy tính).



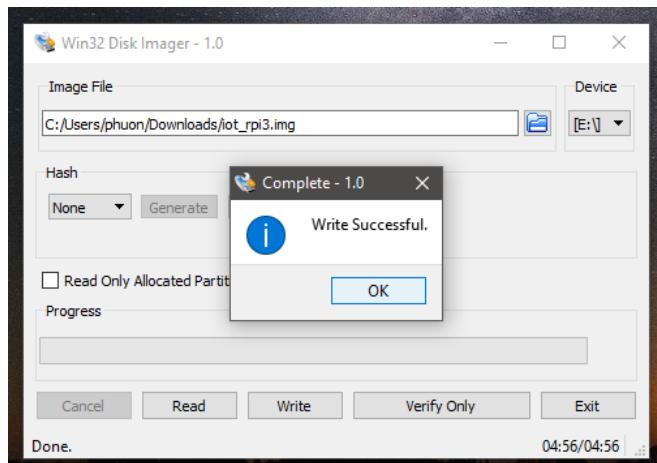
Hình 3.50: Chờ đợi quá trình ghi hệ điều hành.

13. Sau khi thực hiện quá trình ghi xong, máy tính sẽ hiển thị thông báo yêu cầu định dạng lại thẻ nhớ, ấn "Cancel" để thoát nhé.



Hình 3.51: Nhấn Cancel.

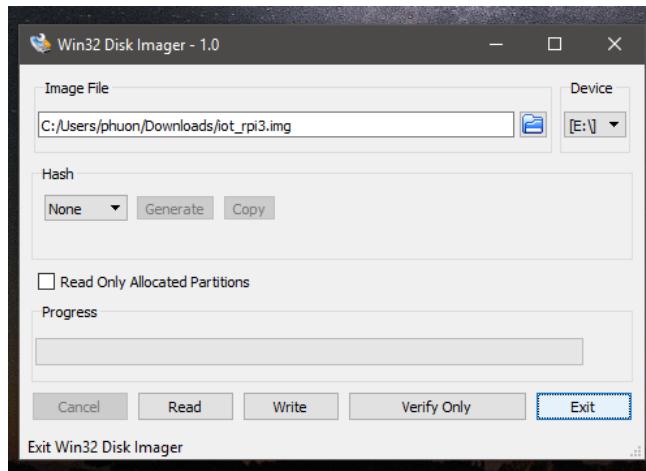
14. Thông báo của việc ghi hệ điều hành lên thẻ nhớ được thực hiện thành công, nhấn "OK".



Hình 3.52: Nhấn OK để tắt thông báo.

15. Vậy là đã cài đặt xong hệ điều hành Android Things lên thẻ nhớ. Việc tiếp theo là gắn thẻ nhớ vào mạch Rpi3.

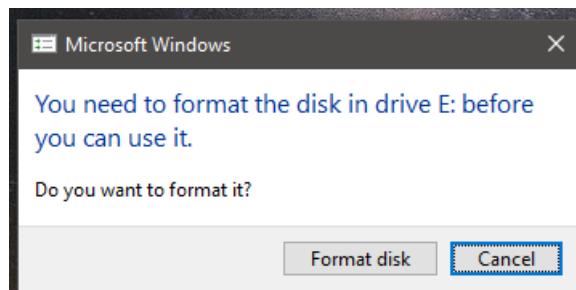
Đề cương luận văn 2018



Hình 3.53: Nhấn Exit để thoát chương trình.

Cách 2: sử dụng phần mềm android-things-setup-utility.

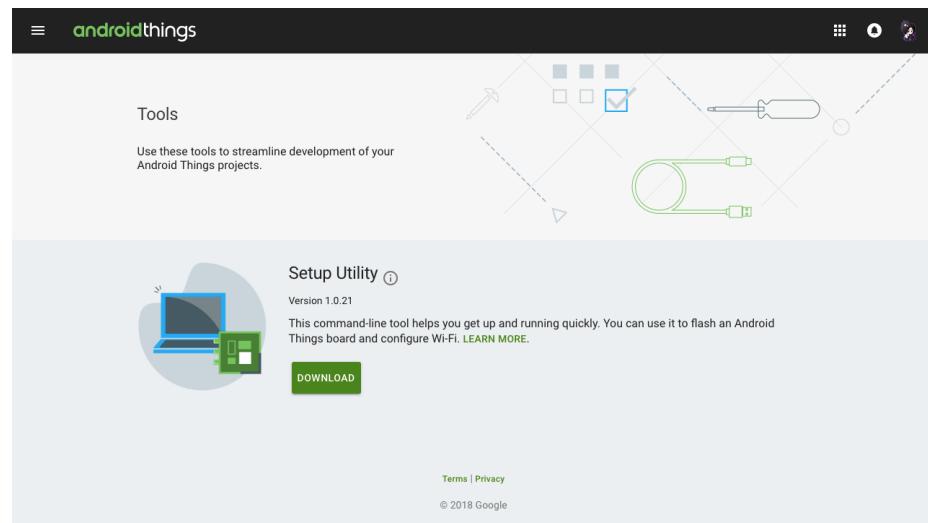
1. Cắm thẻ nhớ MicroSD vào đầu đọc thẻ, sau đó cắm vào máy tính.



Hình 3.54: Nhấn Cancel sau khi cắm thẻ nhớ vô máy.

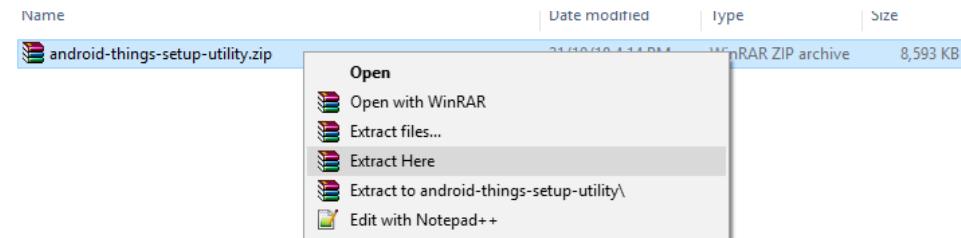
2. Tải phần mềm từ trang <https://partner.android.com/things/console/#/tools>.

Đề cương luận văn 2018



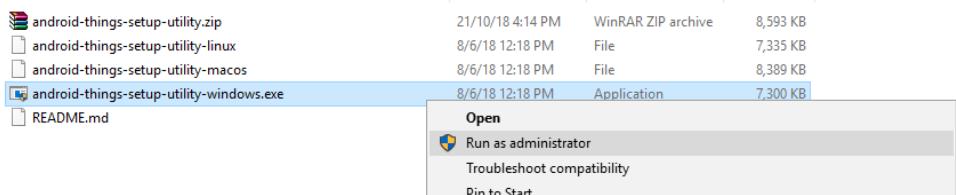
Hình 3.55: Nhấn DOWNLOAD để tải phần mềm.

3. Giải nén phần mềm vừa tải được, chọn tập tin tương ứng với hệ điều hành mà máy tính đang sử dụng.



Hình 3.56: Giải nén phần mềm.

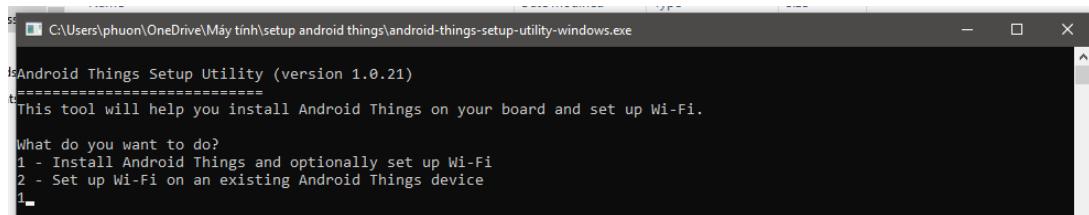
4. Chạy tập tin "android-things-setup-utility-windows.exe" (đối với Windows) bằng quyền Administrator. Ấn "YES" nếu có hiện thông báo.



Hình 3.57: Chuột phải, chọn Run as administrator.

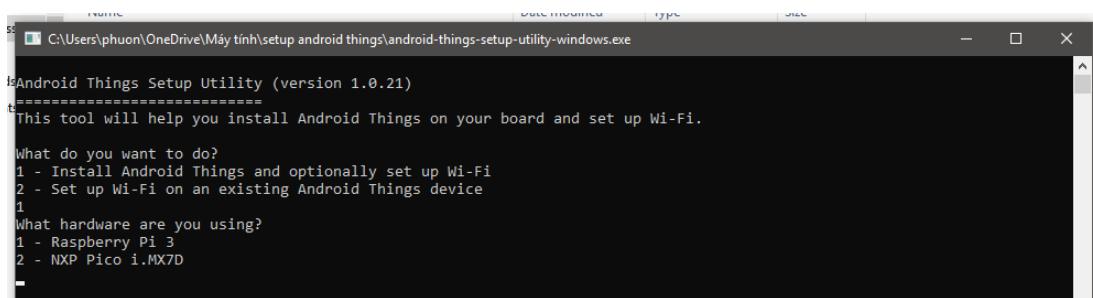
5. Chọn "1" để tiến hành cài đặt hệ điều hành Android Things.

Đề cương luận văn 2018



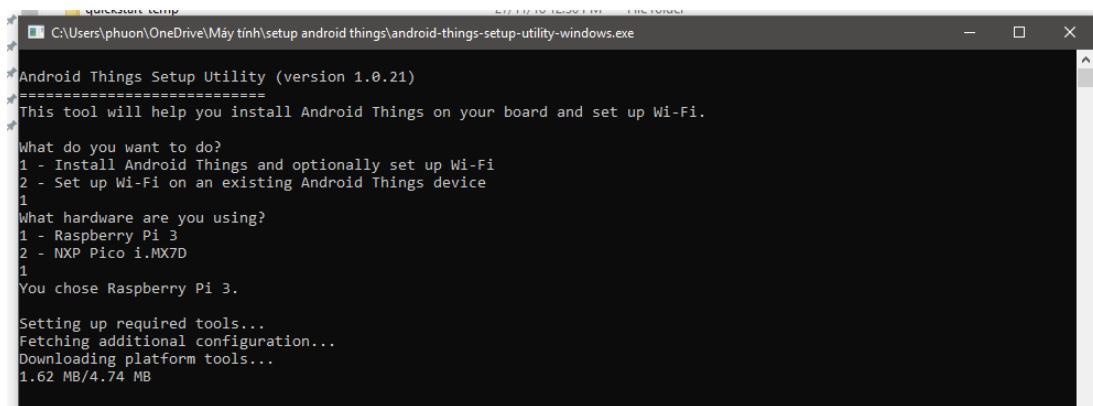
Hình 3.58: Nhấn 1.

6. Chọn "1" để chọn thiết bị phần cứng, ở đây là "Raspberry Pi 3".



Hình 3.59: Nhấn 1 để tiếp tục.

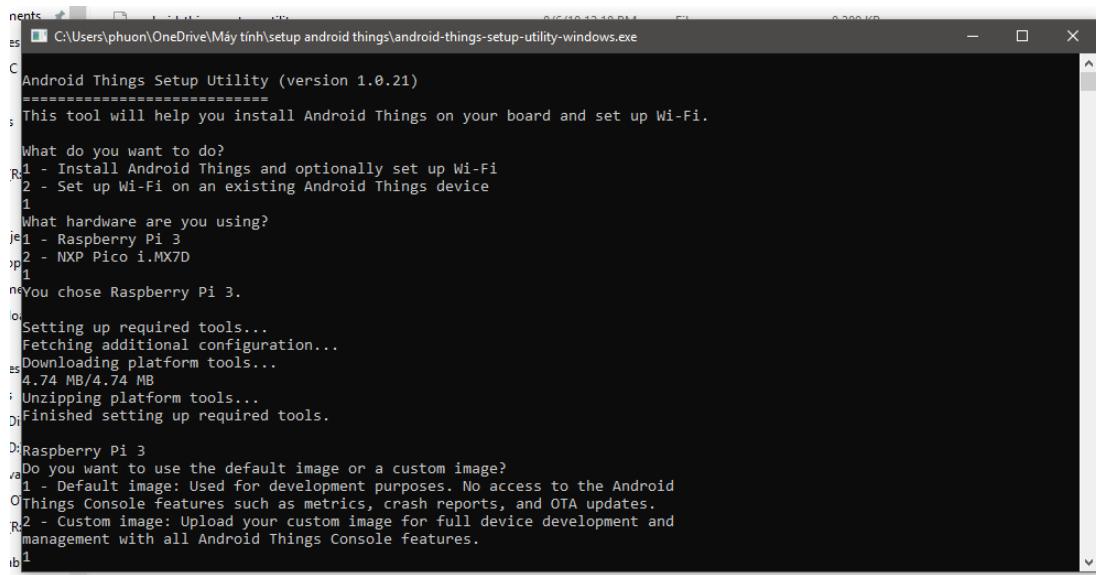
7. Sau vài giây để chương trình tiến hành tải công cụ hỗ trợ. Tại đây, có sự 2 lựa chọn phiên bản hệ điều hành để ghi lên thẻ nhớ.



Hình 3.60: Lựa chọn phiên bản để ghi.

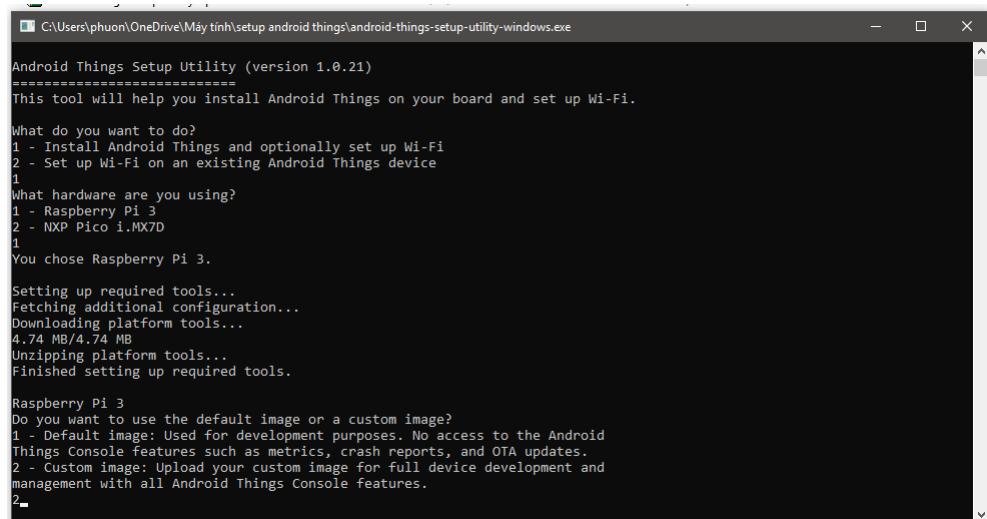
- Nếu chưa tạo bản Image Boot và muốn sử dụng bản mặc định thì ấn "1". Đợi từ 10 - 15 phút để tải bản Image Boot.

Đề cương luận văn 2018



Hình 3.61: Nhấn 1 nếu cài bản mặc định.

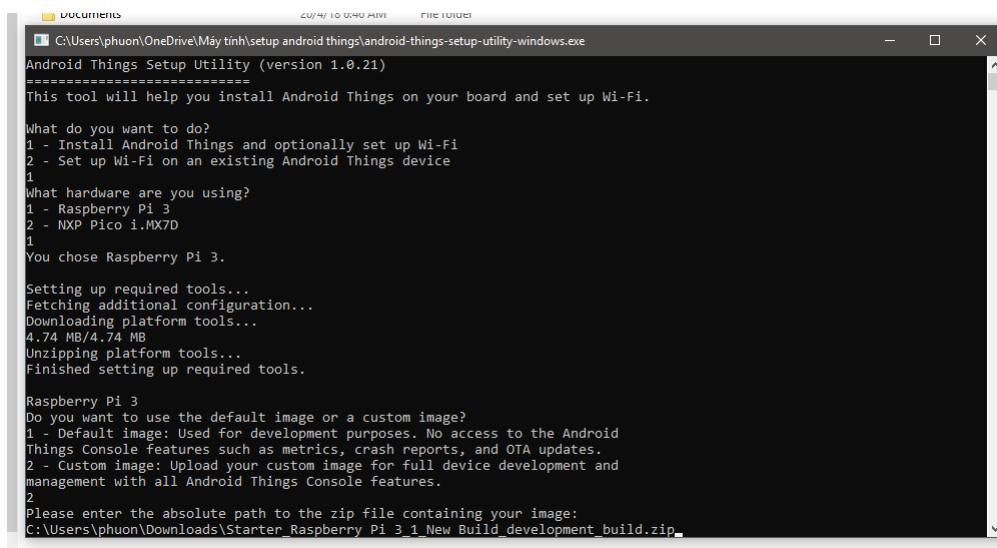
- Nếu đã tạo (Bước 1) thì nhấn "2" để chọn đường dẫn tới tập tin ".zip". Dán địa chỉ của tập tin ".zip" và nhấn ENTER.



Hình 3.62: Nhấn 2 để cài bản đã tải tại Bước 1.

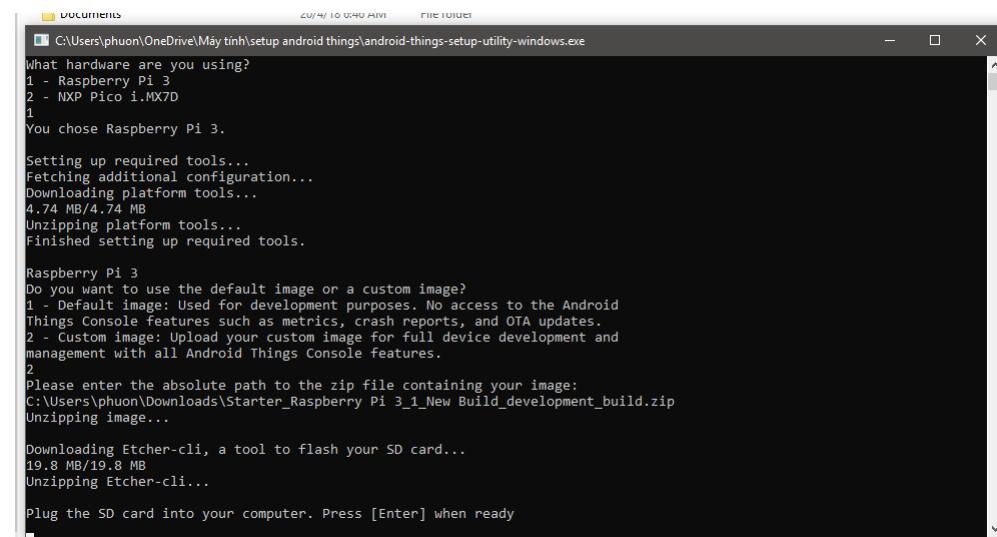
- Dán địa chỉ của tập tin ".zip" và nhấn ENTER. Ở đây, địa chỉ là: C:/Users/phuon/Downloads/Starter_Raspberry Pi 3_1_New-Build_development_build.zip rồi nhấn "ENTER".

Đề cương luận văn 2018



Hình 3.63: Dán địa chỉ chứa tập tin Image Boot .zip.

8. Sau tải xong công cụ Etcher-cli, ấn ENTER để nhận thẻ nhớ.



Hình 3.64: Nhấn ENTER để chọn thẻ nhớ.

9. Nhấn phím mũi tên để chọn thẻ nhớ cần ghi hệ điều hành. Nhấn "ENTER" để xác nhận thẻ nhớ.

Đề cương luận văn 2018

```
C:\Users\phuong\OneDrive\Máy tính\setup android things\android-things-setup-utility-windows.exe
1
You chose Raspberry Pi 3.

Setting up required tools...
Fetching additional configuration...
Downloading platform tools...
4.74 MB/4.74 MB
Unzipping platform tools...
Finished setting up required tools.

Raspberry Pi 3
Do you want to use the default image or a custom image?
1 - Default image: Used for development purposes. No access to the Android Things Console features such as metrics, crash reports, and OTA updates.
2 - Custom image: Upload your custom image for full device development and management with all Android Things Console features.
2
Please enter the absolute path to the zip file containing your image:
C:\Users\phuong\Downloads\Starter_Raspberry Pi 3_1_New Build_development_build.zip
Unzipping image...
Downloading Etcher-cli, a tool to flash your SD card...
19.8 MB/19.8 MB
Unzipping Etcher-cli...
Plug the SD card into your computer. Press [Enter] when ready
Running Etcher-cli...
? Select drive (Use arrow keys)
> \\.\PHYSICALDRIVE1 (15.9 GB) - SDHC Card
```

Hình 3.65: Chọn thẻ nhớ rồi nhấn ENTER.

10. Nhấn "y", rồi nhấn "ENTER" để tiếp tục. *Lưu ý, sau khi cài đặt xong sẽ xuất hiện thông báo yêu cầu định dạng lại thẻ nhớ của Windows, ấn "Cancel" nhé!*

```
C:\Users\phuong\OneDrive\Máy tính\setup android things\android-things-setup-utility-windows.exe
1
You chose Raspberry Pi 3.

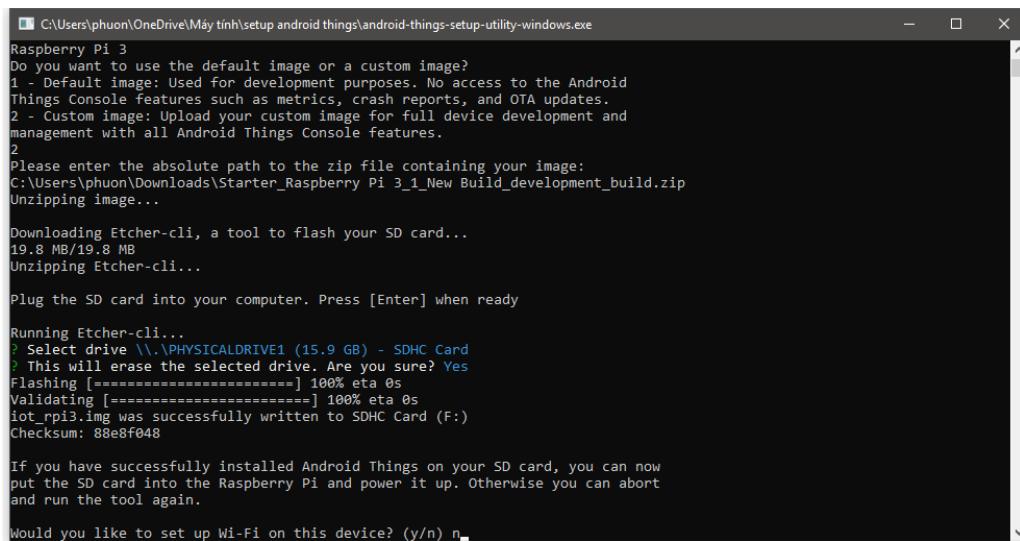
Setting up required tools...
Fetching additional configuration...
Downloading platform tools...
4.74 MB/4.74 MB
Unzipping platform tools...
Finished setting up required tools.

Raspberry Pi 3
Do you want to use the default image or a custom image?
1 - Default image: Used for development purposes. No access to the Android Things Console features such as metrics, crash reports, and OTA updates.
2 - Custom image: Upload your custom image for full device development and management with all Android Things Console features.
2
Please enter the absolute path to the zip file containing your image:
C:\Users\phuong\Downloads\Starter_Raspberry Pi 3_1_New Build_development_build.zip
Unzipping image...
Downloading Etcher-cli, a tool to flash your SD card...
19.8 MB/19.8 MB
Unzipping Etcher-cli...
Plug the SD card into your computer. Press [Enter] when ready
Running Etcher-cli...
? Select drive \\.\PHYSICALDRIVE1 (15.9 GB) - SDHC Card
? This will erase the selected drive. Are you sure? (y/N) y
```

Hình 3.66: Nhấn y, rồi nhấn ENTER.

11. Dợi khoảng 5 phút thì cài đặt xong. Sau đó ấn "n" để thoát thiết lập mạng.

Đề cương luận văn 2018



```
C:\Users\phuong\OneDrive\Máy tính\setup android things\android-things-setup-utility-windows.exe
Raspberry Pi 3
Do you want to use the default image or a custom image?
1 - Default image: Used for development purposes. No access to the Android
Things Console features such as metrics, crash reports, and OTA updates.
2 - Custom image: Upload your custom image for full device development and
management with all Android Things Console features.
2
Please enter the absolute path to the zip file containing your image:
C:\Users\phuong\Downloads\Starter_Raspberry Pi 3_1_New Build_development_build.zip
Unzipping image...

Downloading Etcher-cli, a tool to flash your SD card...
19.8 MB/19.8 MB
Unzipping Etcher-cli...

Plug the SD card into your computer. Press [Enter] when ready

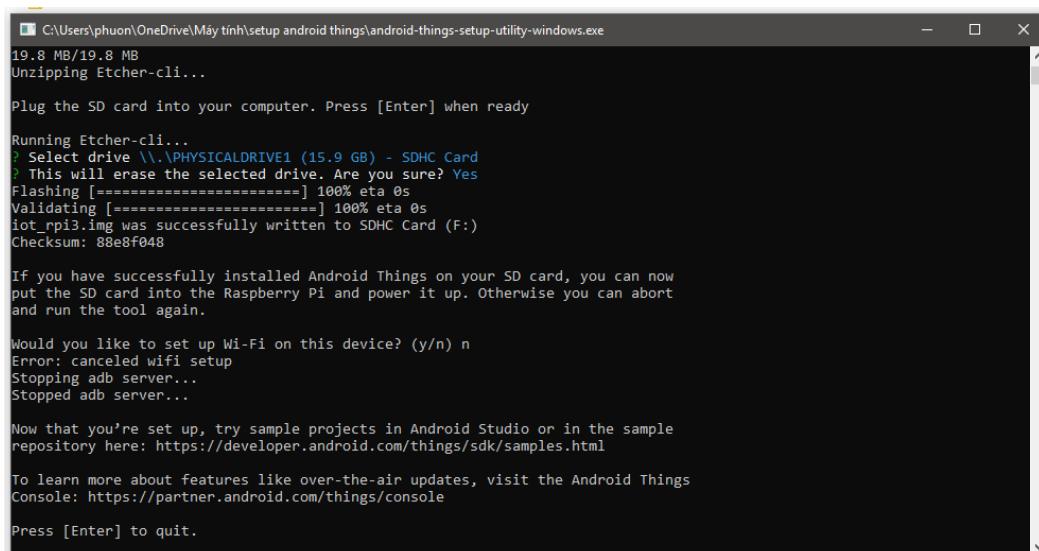
Running Etcher-cli...
? Select drive \\.\PHYSICALDRIVE1 (15.9 GB) - SDHC Card
? This will erase the selected drive. Are you sure? Yes
Flashing [=====] 100% eta 0s
Validating [=====] 100% eta 0s
iot_rpi3.img was successfully written to SDHC Card (F:)
Checksum: 88e8f048

If you have successfully installed Android Things on your SD card, you can now
put the SD card into the Raspberry Pi and power it up. Otherwise you can abort
and run the tool again.

Would you like to set up Wi-Fi on this device? (y/n) n
```

Hình 3.67: Nhấn n rồi nhấn ENTER.

12. Án Enter để thoát chương trình.



```
19.8 MB/19.8 MB
Unzipping Etcher-cli...

Plug the SD card into your computer. Press [Enter] when ready

Running Etcher-cli...
? Select drive \\.\PHYSICALDRIVE1 (15.9 GB) - SDHC Card
? This will erase the selected drive. Are you sure? Yes
Flashing [=====] 100% eta 0s
Validating [=====] 100% eta 0s
iot_rpi3.img was successfully written to SDHC Card (F:)
Checksum: 88e8f048

If you have successfully installed Android Things on your SD card, you can now
put the SD card into the Raspberry Pi and power it up. Otherwise you can abort
and run the tool again.

Would you like to set up Wi-Fi on this device? (y/n) n
Error: canceled wifi setup
Stopping adb server...
Stopped adb server...

Now that you're set up, try sample projects in Android Studio or in the sample
repository here: https://developer.android.com/things/sdk/samples.html

To learn more about features like over-the-air updates, visit the Android Things
Console: https://partner.android.com/things/console

Press [Enter] to quit.
```

Hình 3.68: Nhấn ENTER để thoát.

Lưu ý: Sau khi cài hệ điều hành xong, gắn thẻ nhớ vào Rpi3, rồi mới được cắm nguồn. Tuy nhiên nếu có kết nối màn hình HDMI thì nên cắm cáp HDMI trước rồi mới cắm nguồn để tránh trường hợp hình ảnh hiển thị bị sai độ phân giải.

Đề cương luận văn 2018

Bước 3: Kết nối mạng cho Raspberry Pi 3 để lấy địa chỉ IP.

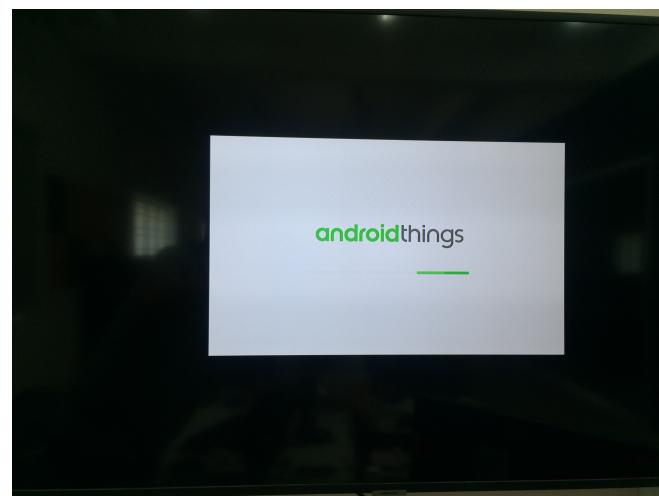
Cách 1: Kết nối wifi và có màn hình.

1. Cắm cáp HDMI với màn hình, gắn bàn phím và chuột vào để điều khiển (nếu màn hình cảm ứng thì không cần thực hiện bước này). Sau đó cắm nguồn cho RPi3.



Hình 3.69: Cắm các kết nối trước rồi mới cắm nguồn.

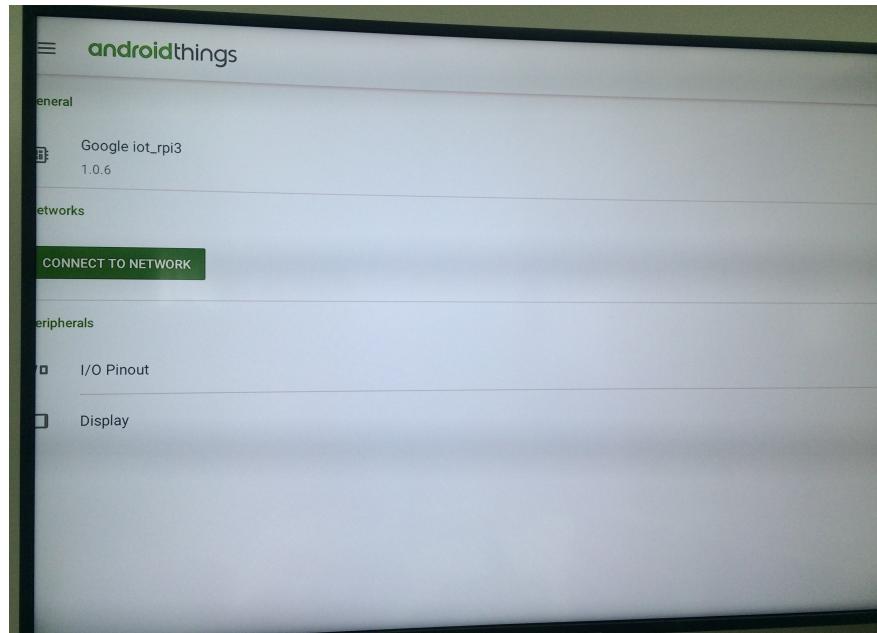
2. Đợi từ 2 - 5 phút cho lần khởi chạy ban đầu, các lần khởi chạy tiếp theo thì thời gian sẽ nhanh hơn.



Hình 3.70: Chờ hệ điều hành khởi động.

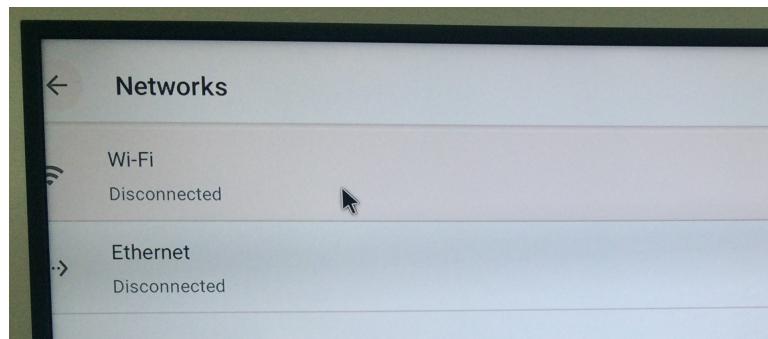
Đề cương luận văn 2018

3. Tại màn hình chính của Android Things, kéo chuột và nhấp vào phần "Networks".



Hình 3.71: Chọn Networks.

4. Chọn phần Wi-Fi để kết nối với mạng wifi.



Hình 3.72: Chọn Wi-Fi.

5. Nếu mới lần đầu kết nối, thì sẽ hiện chữ "Off". Để bật wifi, ấn vào nút cùng hàng với "Off" ở phía bên phải màn hình.

Đề cương luận văn 2018



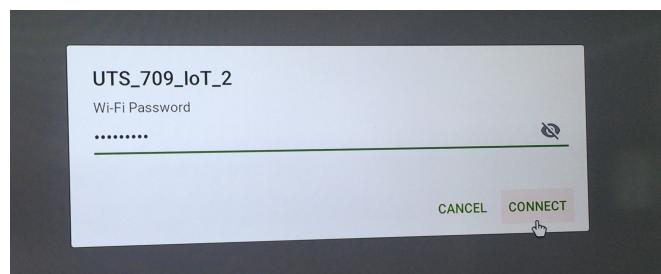
Hình 3.73: Ấn nút bên phải để bật wifi.

6. Sau khi bật wifi, danh sách các wifi khả dụng sẽ được hiện ra.



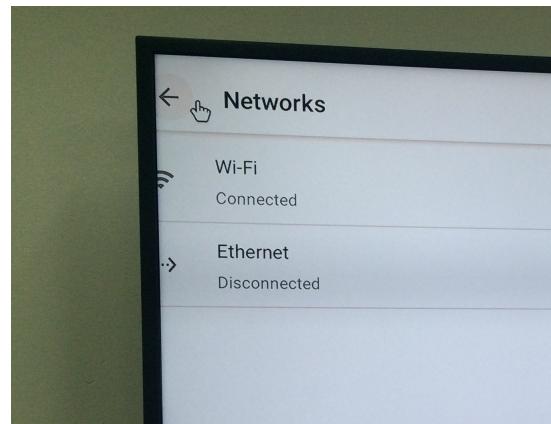
Hình 3.74: Danh sách wifi khả dụng.

7. Chọn wifi, và nhập vào mật khẩu của wifi mà muốn thiết bị kết nối. Nhấn "CONNECT".



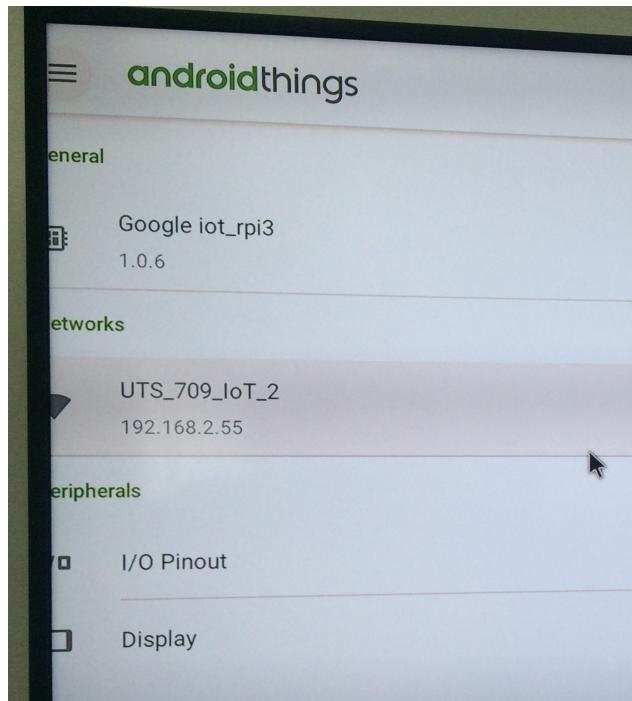
Hình 3.75: Nhập mật khẩu và nhấn CONNECT.

8. Nếu thấy hiện chữ "Connected", tức là đã kết nối wifi thành công. Ấn dấu mũi tên \leftarrow để quay lại màn hình chính, địa chỉ của thiết bị sẽ được hiện ra. Ở đây là: "192.168.2.55".



Hình 3.76: Nhấn < để về màn hình chính.

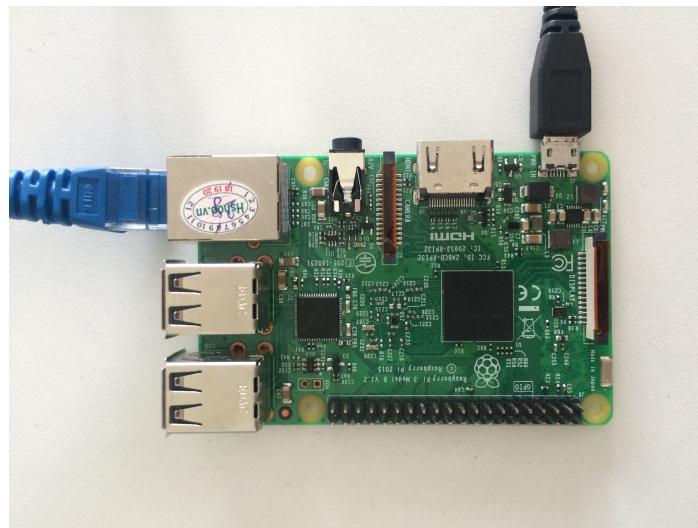
9. Địa chỉ IP của thiết bị sẽ được hiện ra trong phần "Networks". Ở đây là: "192.168.2.55".



Hình 3.77: Networks hiển thị IP của mạch Rpi3.

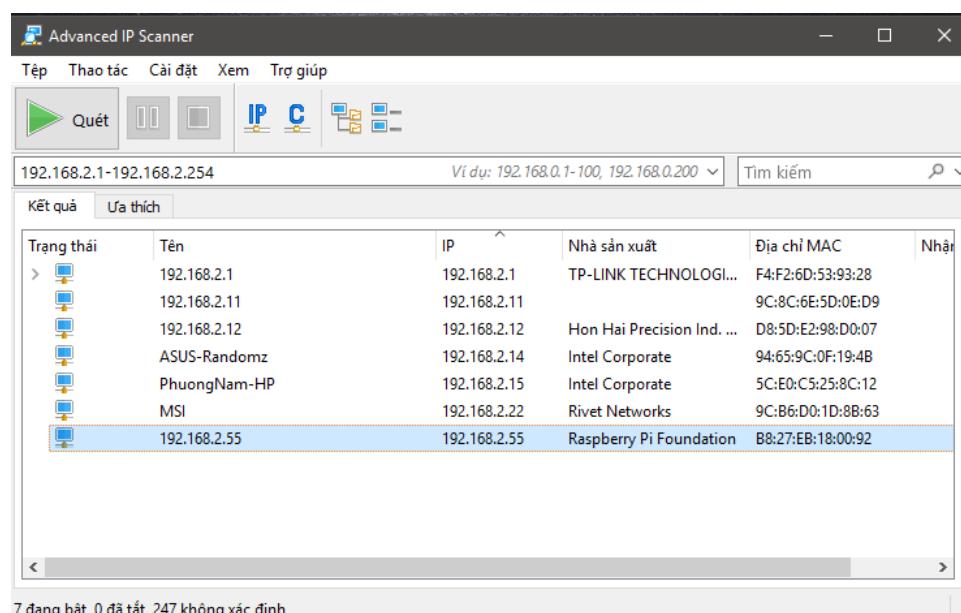
Cách 2: Kết nối mạng dây và không có màn hình.

1. Cắm cáp Ethernet vào mạch Rpi3, sau đó cắm nguồn.



Hình 3.78: Cắm dây Ethernet và cắm nguồn.

2. Đợi từ 2 - 5 phút cho lần khởi chạy ban đầu, các lần khởi chạy tiếp theo thì thời gian sẽ nhanh hơn.
3. Kết nối laptop/desktop cùng với mạng đã kết nối với Rpi3. Sử dụng phần mềm Advanced IP Scanner, tìm với tên tại cột "Nhà sản xuất" là "Raspberry Pi Fundation" để lấy địa chỉ IP (192.168.2.55).

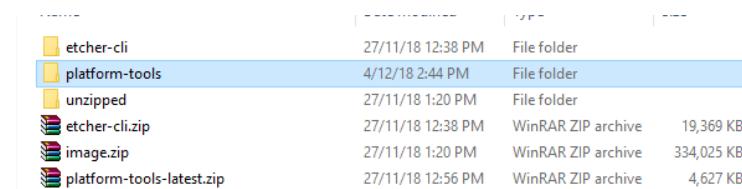


Hình 3.79: Chạy chương trình Advanced IP Scanner để quét địa chỉ IP của Rpi3.

Đề cương luận văn 2018

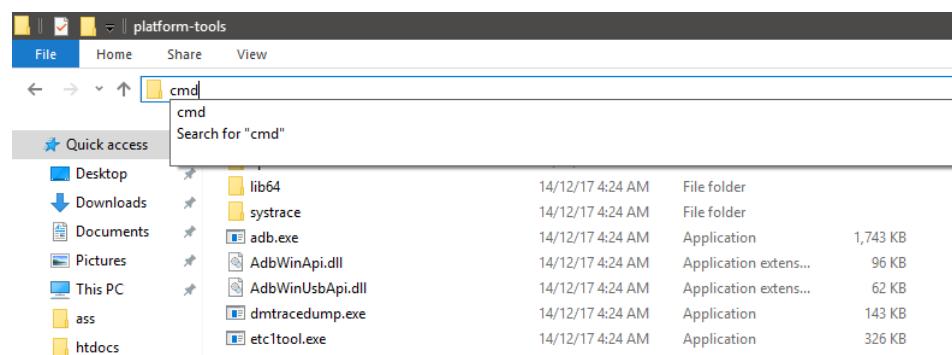
Bước 4: Kết nối với Raspberry Pi 3 bằng địa chỉ IP.

- Sau khi có được địa chỉ của mạch Rpi3. Chạy ứng dụng ADB để thực hiện việc kết nối.



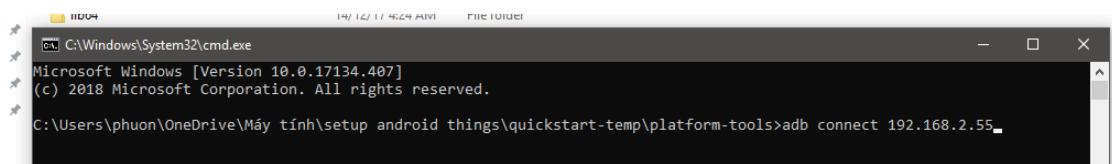
Hình 3.80: Truy cập thư mục platform-tools.

- Chạy "Command Prompt" (Windows) hoặc "Terminal" (Mac/Linux), tại thư mục (platform-tools) chứa tập tin "adb".



Hình 3.81: Chạy Command Prompt tại thư mục platform-tools.

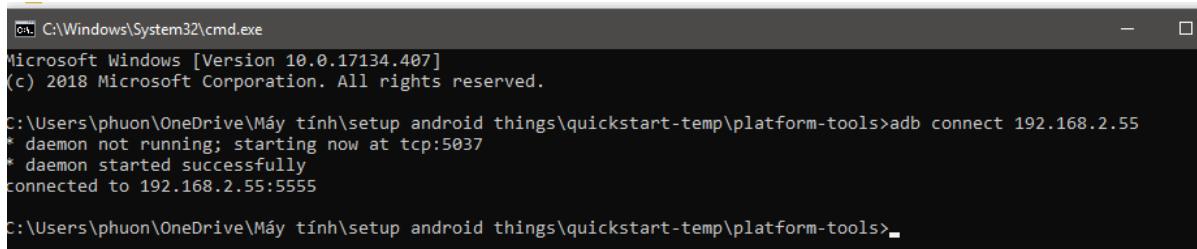
- Nhập lệnh "adb connect <IP>", với <IP> là địa chỉ IP của mạch Rpi3. Ví dụ "adb connect 192.168.2.55".



Hình 3.82: Kết nối với thiết bị "adb connect <IP>".

Đề cương luận văn 2018

4. Kết nối thành công với địa chỉ IP 192.168.2.55 thông qua port 5555.



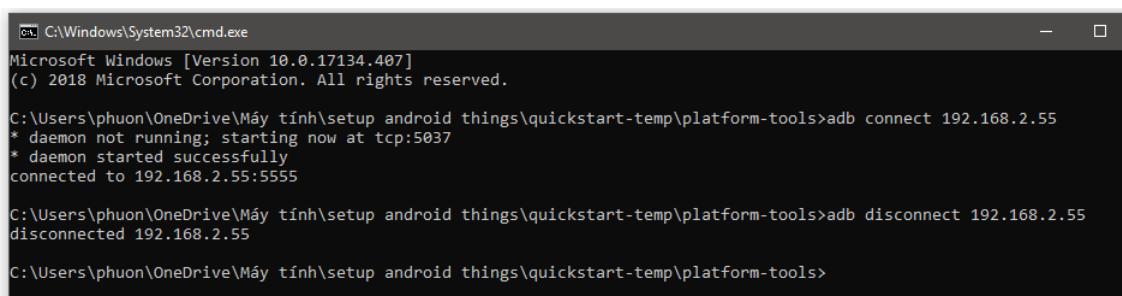
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\phuon\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>adb connect 192.168.2.55
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.2.55:5555

C:\Users\phuon\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>
```

Hình 3.83: Kết nối với mạch Raspberry Pi 3 thành công.

5. Nếu muốn huỷ kết nối với một thiết bị thì dùng lệnh "adb disconnect <IP>".



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

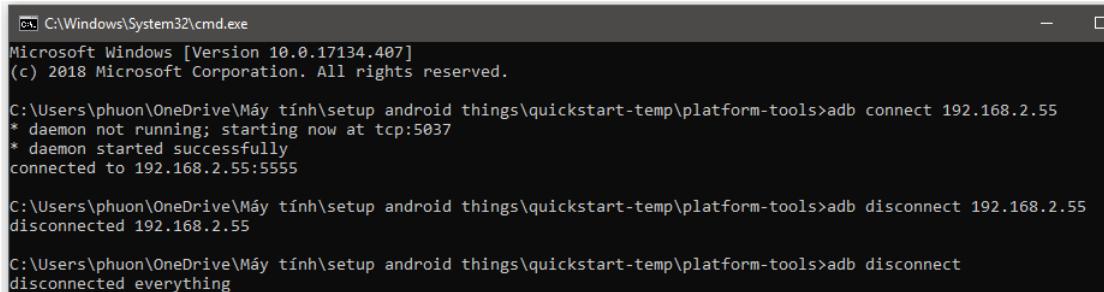
C:\Users\phuon\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>adb connect 192.168.2.55
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.2.55:5555

C:\Users\phuon\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>adb disconnect 192.168.2.55
disconnected 192.168.2.55

C:\Users\phuon\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>
```

Hình 3.84: Hủy kết nối với một thiết bị "adb disconnect <IP>".

6. Còn nếu muốn huỷ hết các thiết bị đang kết nối thì nhập lệnh "adb disconnect".



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\phuon\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>adb connect 192.168.2.55
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.2.55:5555

C:\Users\phuon\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>adb disconnect 192.168.2.55
disconnected 192.168.2.55

C:\Users\phuon\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>adb disconnect
disconnected everything
```

Hình 3.85: Hủy kết nối với tất cả thiết bị "adb disconnect".

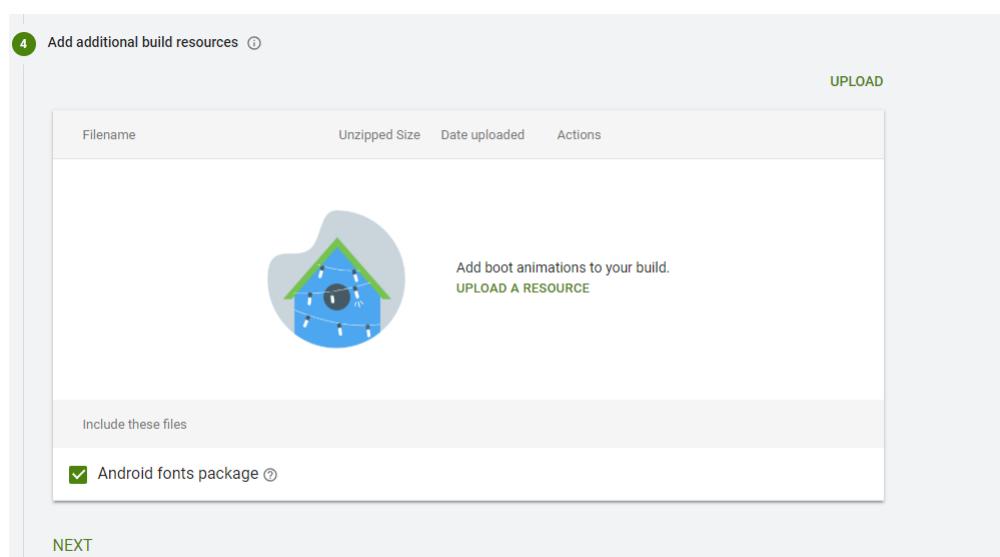
3.5.4 Các công cụ hỗ trợ

3.5.4.1 Thay đổi hình ảnh của Image Boot

Để thay đổi hình ảnh của Image Boot khi chạy hệ điều hành Android Things, thì phải sử dụng tập tin "bootanimation.zip" tại bước tạo Image Boot.

Phần hướng dẫn về cách tạo "bootadnimation.zip" hoặc có thể tải các tập tin bootanimation.zip từ trên mạng:

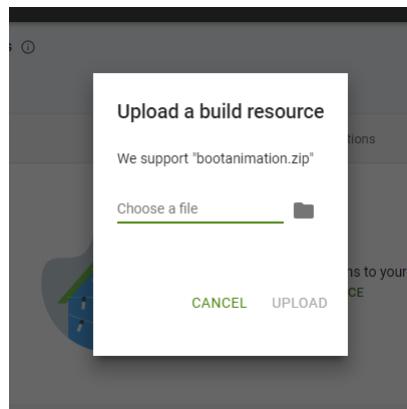
1. Chọn một đoạn clip từ 5 - 10 giây.
2. Sử dụng công cụ DVDDVideoSoft Free Studio để tách đoạn clip thành hình ảnh.
3. Chia ảnh vào các thư mục phù hợp.
4. Dùng công cụ Boot Animation Creator để tạo ra file "bootanimation.zip".
5. Tại trường "4", nhấn "UPLOAD" để tải lên tập tin "bootanimation.zip".



Hình 3.86: Chọn UPLOAD.

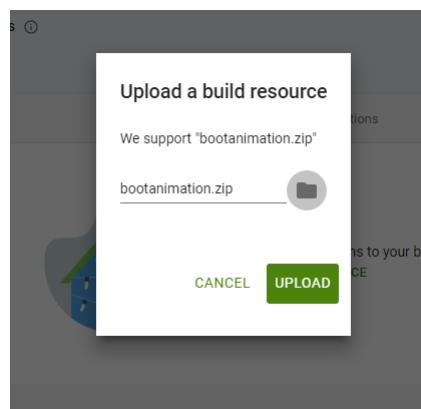
Đề cương luận văn 2018

6. Hiện lên thông báo "Upload a build resource", nhấn chọn biểu tượng thư mục.



Hình 3.87: Chọn biểu tượng thư mục.

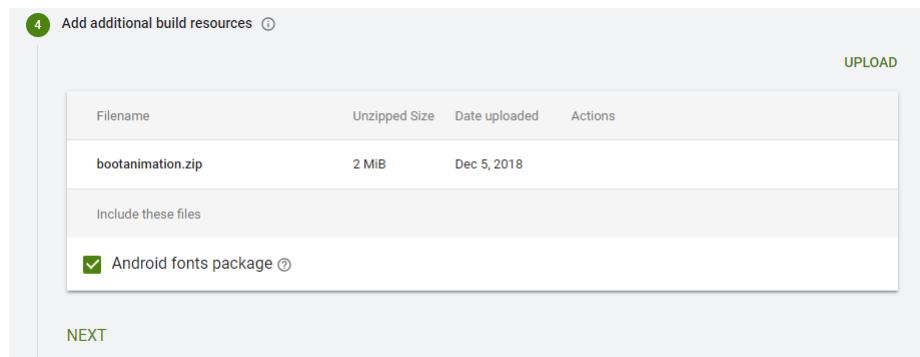
7. Chọn tập tin "bootanimation.zip" cần tải lên.



Hình 3.88: Chọn file "bootanimation.zip", nhấn OK.

8. Tập tin được tải lên thành công, sau đó ấn "NEXT" để tiếp tục trường "5".

Đề cương luận văn 2018



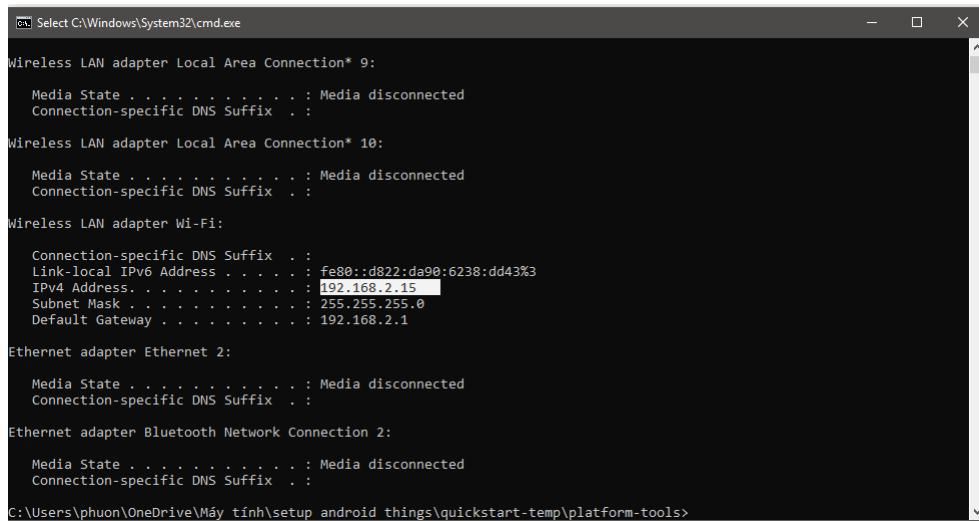
Hình 3.89: Tải tập tin thành công, ấn NEXT để tiếp tục.

3.5.4.2 Advanced IP Scanner (Windows)

Advanced IP Scanner là ứng dụng quét mạng miễn phí, hoạt động nhanh và rất dễ sử dụng dành cho người dùng Windows. Chỉ trong vài giây, công cụ này sẽ tìm ra tất cả các máy trong mạng và cung cấp cách truy cập dễ dàng vào các nguồn tài nguyên của chúng, ví dụ như HTTP, HTTPS, FTP hoặc folder đã chia sẻ. Với Advanced IP Scanner, người dùng có thể bật và tắt máy từ xa.

- Tải tại đây: <https://www.advanced-ip-scanner.com/>
- Hướng dẫn sử dụng:
 1. Chạy "Command Prompt" (Windows) nhập lệnh "ipconfig" hoặc "Terminal" (Mac/Linux) nhập lệnh "ifconfig" để lấy địa chỉ IPv4 hiện tại.

Đề cương luận văn 2018



```
Select C:\Windows\System32\cmd.exe

Wireless LAN adapter Local Area Connection* 9:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 10:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

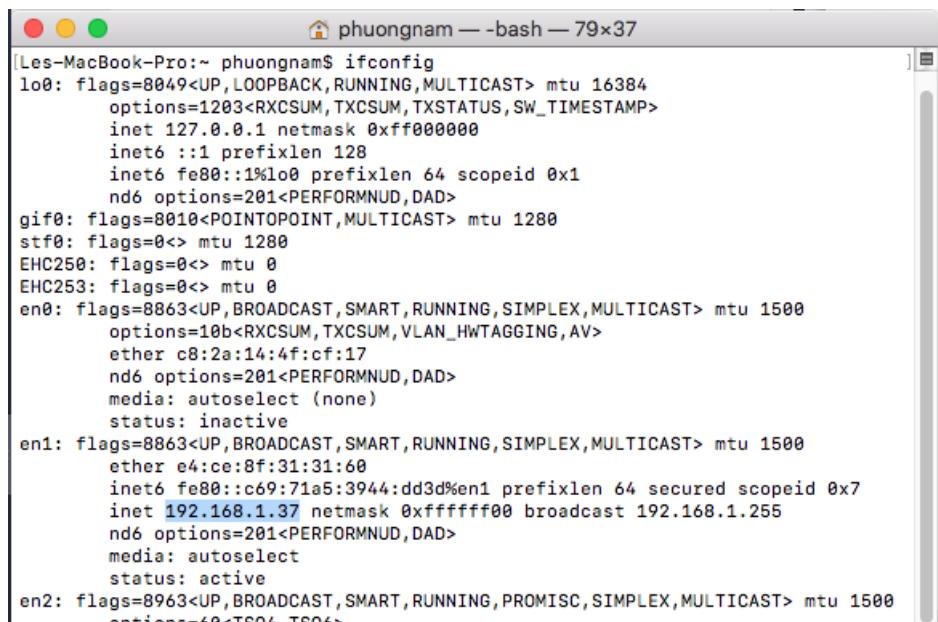
Wireless LAN adapter Wi-Fi:
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::d822:da90:6238:dd43%3
  IPv4 Address . . . . . : 192.168.2.15
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.2.1

Ethernet adapter Ethernet 2:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Ethernet adapter Bluetooth Network Connection 2:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

C:\Users\phuong\OneDrive\Máy tính\setup android things\quickstart-temp\platform-tools>
```

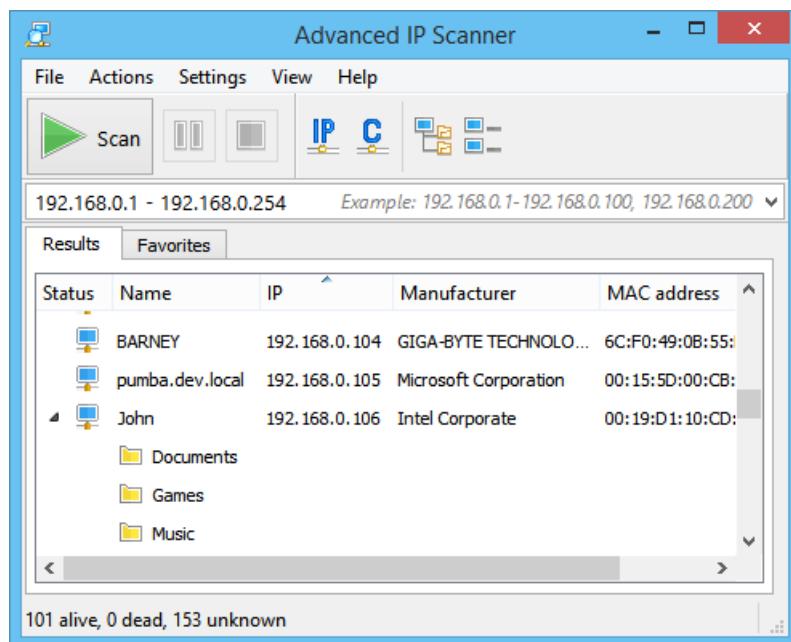
Hình 3.90: Lệnh "ipconfig" trên Windows.



```
[Les-MacBook-Pro:~ phuongnam$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
      options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
      inet 127.0.0.1 netmask 0xff000000
          inet6 ::1 prefixlen 128
              inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
                  nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
EHC250: flags=0<> mtu 0
EHC253: flags=0<> mtu 0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      options=10b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV>
      ether c8:2a:14:4f:cf:17
      nd6 options=201<PERFORMNUD,DAD>
      media: autoselect (none)
      status: inactive
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether e4:ce:8f:31:31:60
      inet6 fe80::c69:71a5:3944:dd3d%en1 prefixlen 64 secured scopeid 0x7
          inet 192.168.1.37 netmask 0xffffffff broadcast 192.168.1.255
          nd6 options=201<PERFORMNUD,DAD>
          media: autoselect
          status: active
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
      options=200<NO_TSOL>
```

Hình 3.91: Lệnh "ifconfig" trên Mac/Linux.

2. Nhập khoảng địa chỉ mà máy tính đang kết nối để quét. Ví dụ: IP máy tính 192.168.0.30, thì sẽ nhập khoảng 192.168.0.1 - 192.168.0.254.
3. Sau đó ấn "Scan". Kết quả sau khi quét sẽ được trả về như hình 3.92.



Hình 3.92: Nhấn Scan để quét.

3.5.4.3 ADB

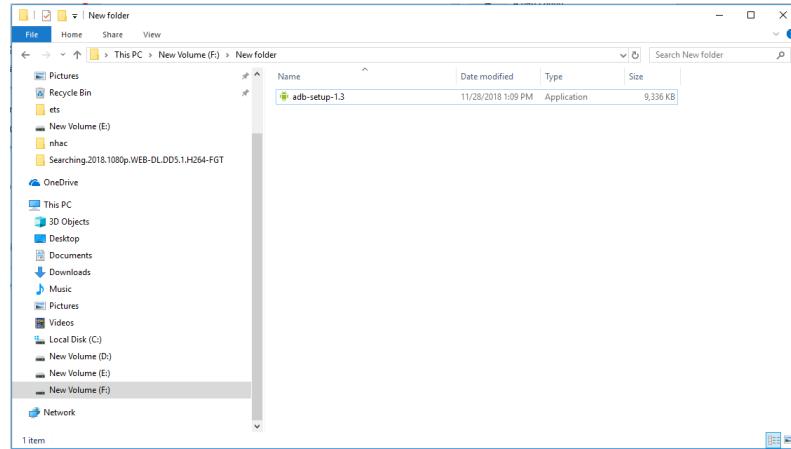
ADB thường được sử dụng khi cố gắng chạy các ứng dụng dành cho điện thoại trên máy tính, do đó bạn có thể debug (gỡ lỗi) các lỗi trên các ứng dụng của mình, ứng dụng mà bạn đang tạo. ADB được sử dụng cho các thiết bị Android root.

Lý do bởi vì ADB cho phép bạn giao tiếp với một điện thoại Android ở mức độ phát triển nào đó, do đó nó rất tiện dụng trong một số trường hợp chẳng hạn như khi chúng ta muốn ra lệnh cho phép mình chuyển các tập tin vào thiết bị và sau đó thực thi tất cả các tập tin trong điện thoại đã root.

Cài đặt ADB:

Đề cương luận văn 2018

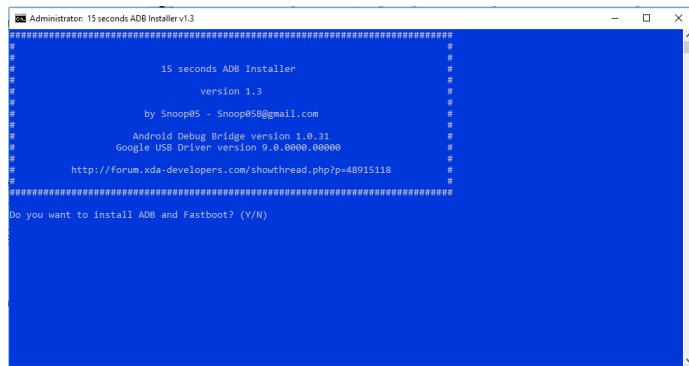
1. vào trang wed này để tải ADB: <https://forum.xda-developers.com/showthread.php?p=48915118#post48915118>



Hình 3.93: Tệp adb được tải về.

2. Chạy file adb vừa tải về. sau đó nhấn Y để cài đặt.

Đề cương luận văn 2018



Hình 3.94: Giao diện adb.

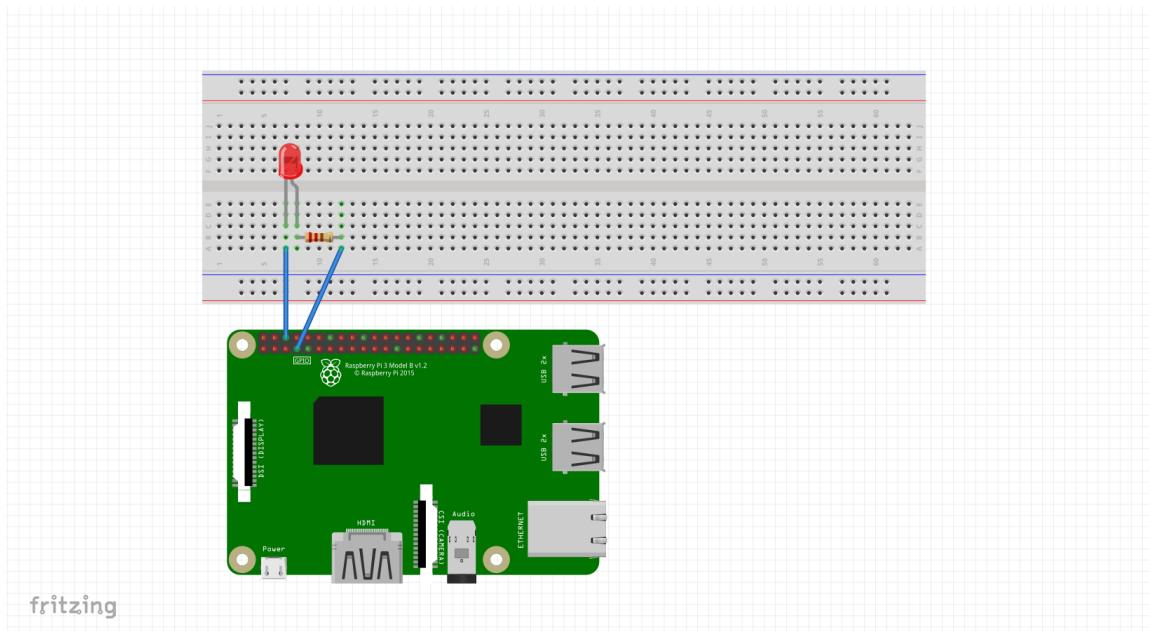
3.5.5 Project LED Blinky

Chuẩn bị:

- 1 board raspberry pi 3.
- 1 điện trở 200 Ohm
- 1 led.
- 1 breadboard.

Thực hiện nối mạch như hình.

Đề cương luận văn 2018



Hình 3.95: Mạch blink led.

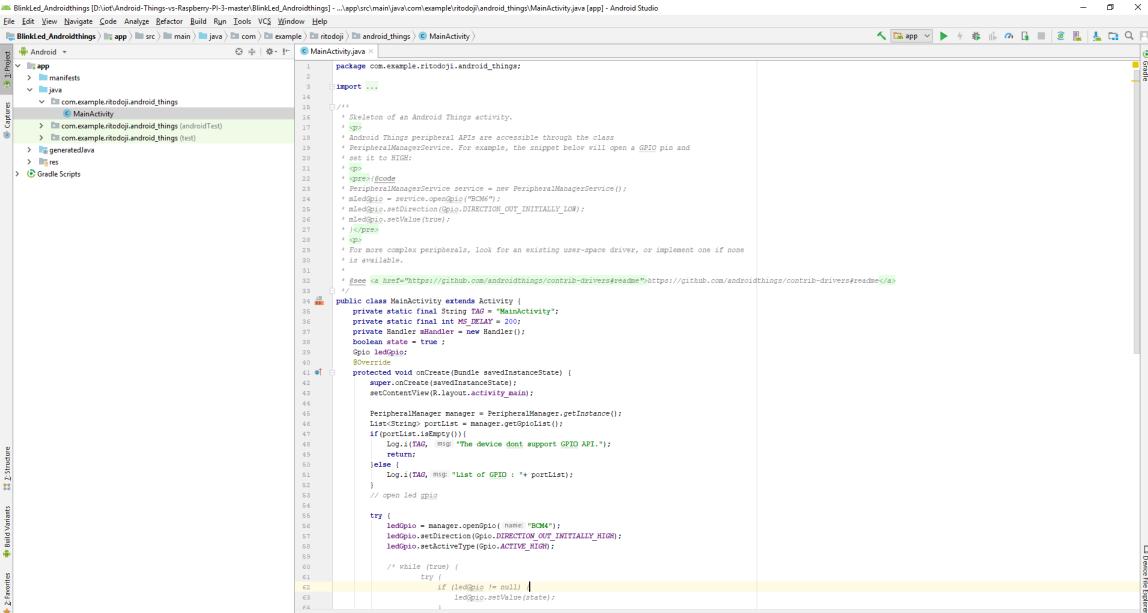
Vào trang web <https://github.com/HungSoma/Android-Things-vs-Raspberry-Pi>.

Mở android studio, mở project blink led, màn hình xuất hiện như sau.

Bạn xem địa chỉ IP của raspberry, ở đây của mình là 192.168.2.18

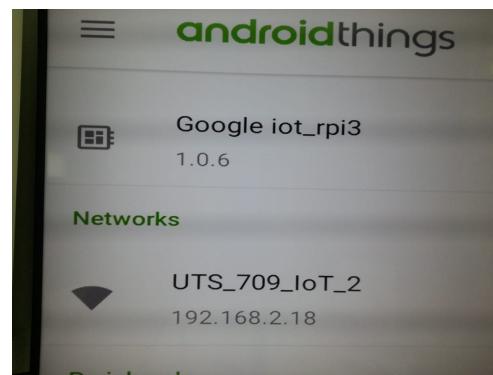
Mở Command Prompt sau đó gõ lệnh adb để khởi động adb.

Đề cương luận văn 2018



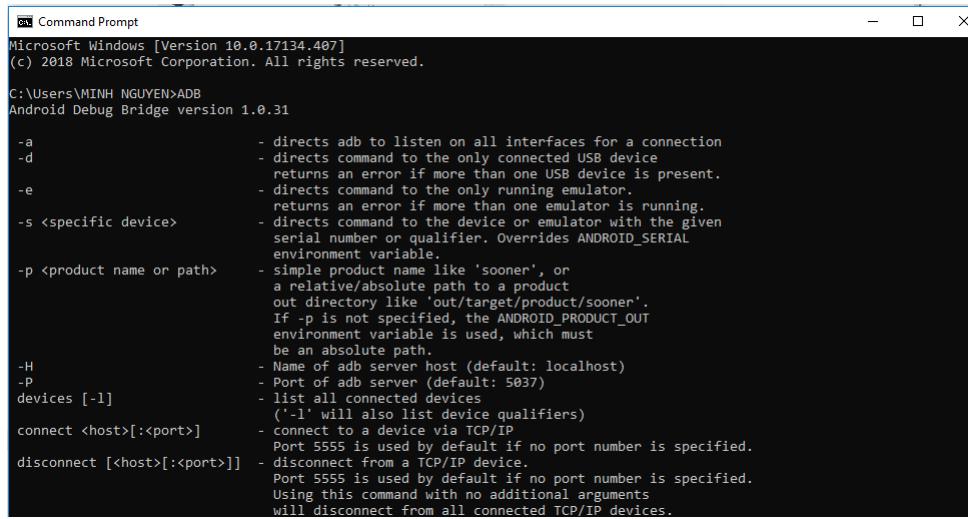
```
package com.example.ritodaji.android_things;
import ...;
...
public class MainActivity extends Activity {
    private static final String TAG = "MainActivity";
    private static final int MC_INTERRUPT = 200;
    private Handler handler = new Handler();
    boolean state = true;
    Gpio ledGpio;
    ...
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        PeripheralManager manager = PeripheralManager.getInstance();
        List<String> portList = manager.getGpioList();
        if(portList.size() == 0) {
            Log.i(TAG, msg("The device dont support GPIO API."));
            return;
        } else {
            Log.i(TAG, msg("List of GPIO : " + portList));
        }
        // open led gpio
        try {
            ledGpio = manager.openGpio("BCM4");
            ledGpio.setDirection(Gpio.DIRECTION_OUT_INITIALLY_HIGH);
            ledGpio.setActiveType(Gpio.ACTIVE_HIGH);
            ...
        } catch (Exception e) {
            if (ledGpio != null) {
                ledGpio.close();
            }
        }
    }
}
```

Hình 3.96: Giao diện android.



Hình 3.97: IP raspberry.

Đề cương luận văn 2018



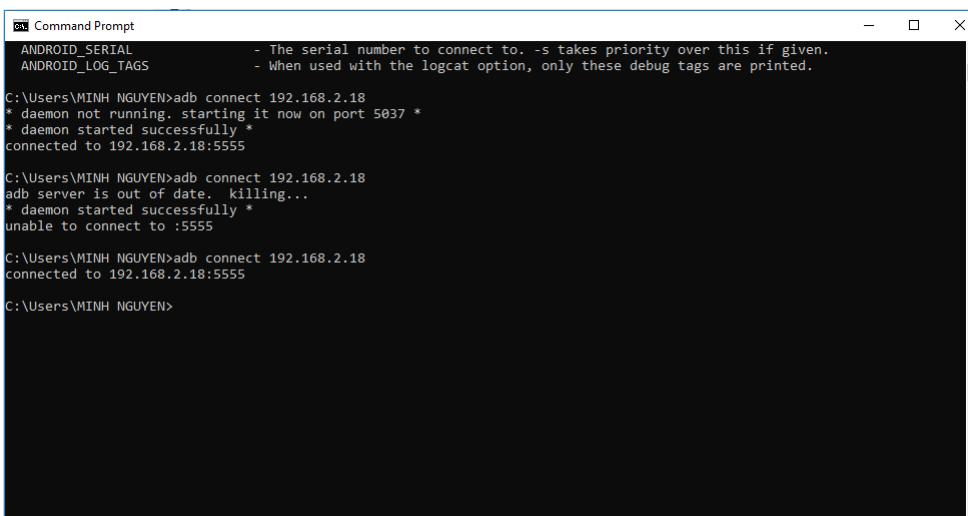
```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\MINH NGUYEN>adb
Android Debug Bridge version 1.0.31

-a           - directs adb to listen on all interfaces for a connection
-d           - directs command to the only connected USB device
-e           - directs command to the only running emulator.
             returns an error if more than one USB device is present.
             returns an error if more than one emulator is running.
-s <specific device> - directs command to the device or emulator with the given
             serial number or qualifier. Overrides ANDROID_SERIAL
             environment variable.
-p <product name or path> - simple product name like 'sooner', or
             a relative/absolute path to a product
             out directory like 'out/target/product/sooner'.
             If -p is not specified, the ANDROID_PRODUCT_OUT
             environment variable is used, which must
             be an absolute path.
-H           - Name of adb server host (default: localhost)
-P           - Port of adb server (default: 5037)
devices [-l] - list all connected devices
              ('-l' will also list device qualifiers)
connect <host>[:<port>] - connect to a device via TCP/IP
              Port 5555 is used by default if no port number is specified.
disconnect [<host>[:<port>]] - disconnect from a TCP/IP device.
              Port 5555 is used by default if no port number is specified.
              Using this command with no additional arguments
              will disconnect from all connected TCP/IP devices.
```

Hình 3.98: Gõ lệnh adb.

Tiếp tục gõ lệnh: abc connect 192.168.2.18 để kết nối, 192.168.2.18 là địa chỉ ip hiện tại của mình, các bạn hay thay địa chỉ ip của các bạn.
Kích vào nút run app trên android studio để xem kết nối.Sau đó nhấn ok để nạp.



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\MINH NGUYEN>adb connect 192.168.2.18
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
connected to 192.168.2.18:5555

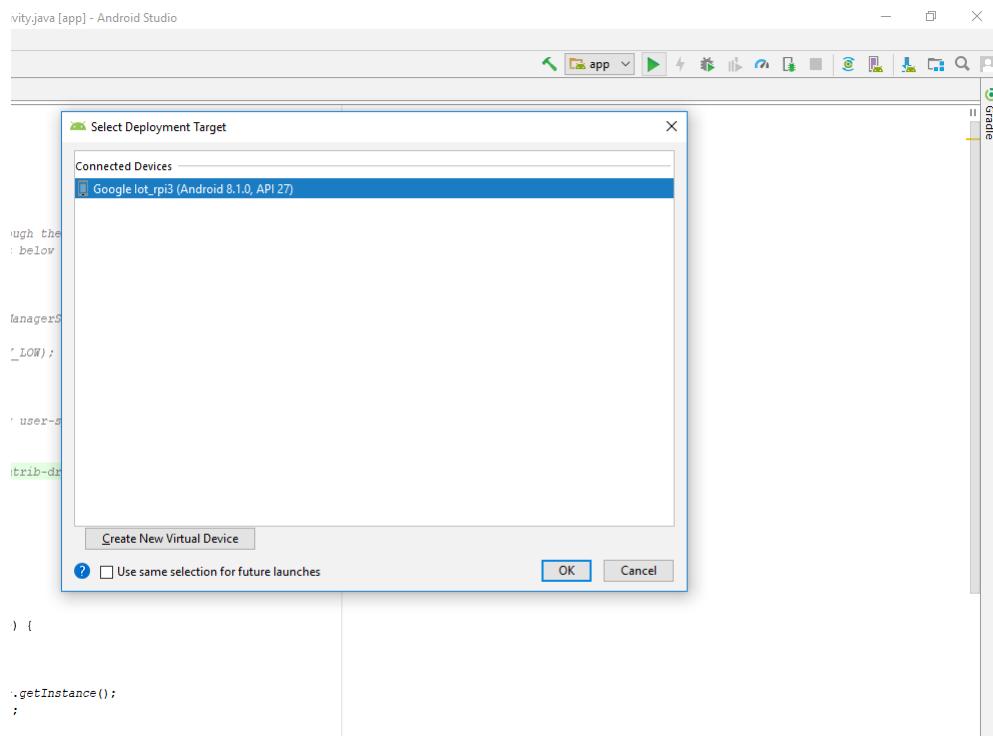
C:\Users\MINH NGUYEN>adb connect 192.168.2.18
adb server is out of date. killing...
* daemon started successfully *
unable to connect to :5555

C:\Users\MINH NGUYEN>adb connect 192.168.2.18
connected to 192.168.2.18:5555

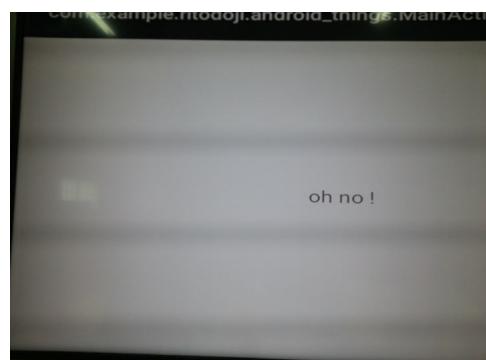
C:\Users\MINH NGUYEN>
```

Hình 3.99: Kết nối PC với raspberry.

Đề cương luận văn 2018



Hình 3.100: Hiển thị kết nối trên androidstudio.



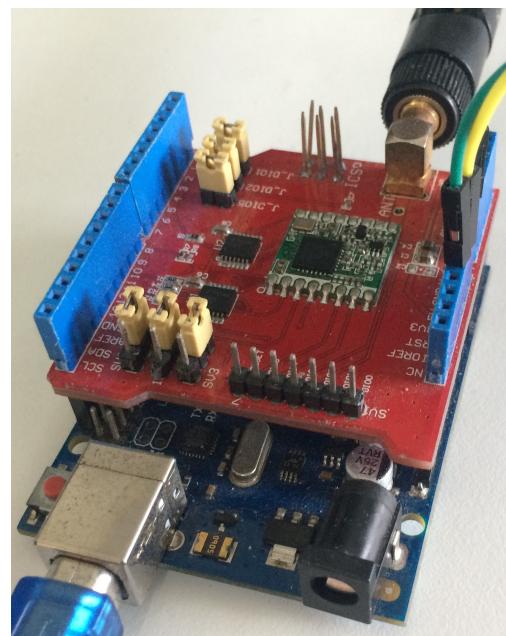
Hình 3.101: Kết quả hiển thị trên raspberry.

CHƯƠNG 4

HIỆN THỰC GIAO TIẾP KHÔNG DÂY CHO NODE CẢM BIẾN

4.1 Giới thiệu

Sử dụng công nghệ truyền thông không dây LoRa [12] (Long Range Radio).



Hình 4.1: Gateway LoRa

Mạch LoRa Shield v1.3 kết hợp với mạch Arduino Uno R3 và sử dụng các cảm biến (sensor) - phụ thuộc vào nhu cầu mà sử dụng các loại cảm biến phù hợp.

4.1.1 LoRa

Điểm quan trọng của ứng dụng IoT yêu cầu chỉ truyền rất ít bit dữ liệu để theo dõi (monitor) các thiết bị tầm xa. Hệ thống mạng di động thì không phù hợp với vấn đề năng lượng pin (battery) và hiệu quả kinh tế khi gửi ít dữ liệu đi. Vì vậy, Low Power Wide Area Network (LPWAN) được đưa ra cho những ứng dụng này. LPWAN thích hợp cho việc gửi một lượng nhỏ dữ liệu với khoảng cách xa, trong khi thời lượng pin dài.



Hình 4.2: Giới thiệu công nghệ LoRa [15]

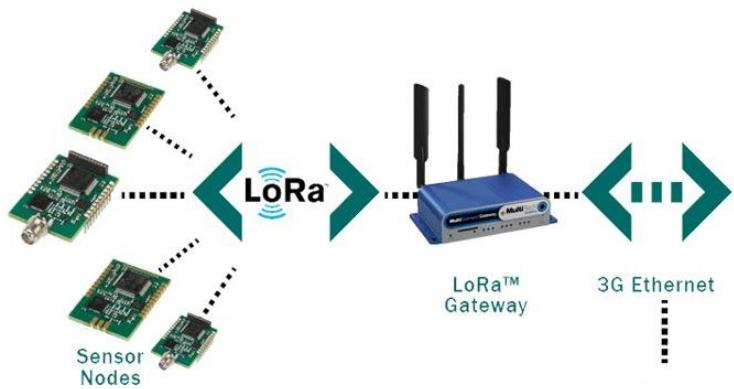
4.1.1.1 LoRa là gì?

LoRaTM [14] (Long Range) là một kỹ thuật điều chế (modulation) dựa trên kỹ thuật Spread-Spectrum và một biến thể của Chirp Spread Spectrum (CSS), nó cho một khoảng cách xa hơn đáng kể cách kỹ thuật khác. Kỹ thuật không dây LoRa được phát triển bởi Cycleo SAS (sau này được mua lại bởi Semtech).

4.1.1.2 LoRaWAN là gì?

Điều chế LoRa nằm ở lớp vật lý (physical layer) của LoRaWAN, nó là một MAC protocol cho high capacity long range và Low Power Wide Area Networks (LPWAN). LoRa Alliance là một tổ chức mở phi lợi nhuận với các thành viên làm việc để chuẩn hóa LoRa Wide Area Network protocol (a.k.a LoRaWAN) cho LPWAN. LoRaWAN chú trọng vào những nhu cầu

cơ bản của IoT như bảo mật giao tiếp 2 chiều (bidirection communication), tính di động và các dịch vụ nội địa.



Hình 4.3: Giới thiệu LoRaWan

Hơn nữa, LoRaWAN network server quản lý data rate và RF output cho mỗi end-device một cách độc lập bằng giá trị trung bình (mean) của biểu đồ data rate thích nghi (adaptive data rate (ADR)), nó có thể mở rộng thời lượng pin và điều khiển thiết bị cuối lên tới 10 năm, và còn gia tăng tổng dung lượng network tới mức độ rất lớn.

4.1.2 Arduino

"Arduino là một bo mạch xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ,... Điểm hấp dẫn ở Arduino với người lập trình là ngôn ngữ cực kì dễ học (giống C/C++), các ngoại vi trên bo mạch đều đã được chuẩn hóa, nên không cần biết nhiều về điện tử, chúng ta cũng có thể lập trình được những ứng dụng thú vị.Thêm nữa, vì Arduino là một platform đã được chuẩn hóa, nên đã có rất nhiều các bo mạch mở rộng (gọi là shield) để cắm chồng lên bo mạch Arduino, có thể hình dung nôm na là “library” của các ngôn ngữ lập trình." [16]



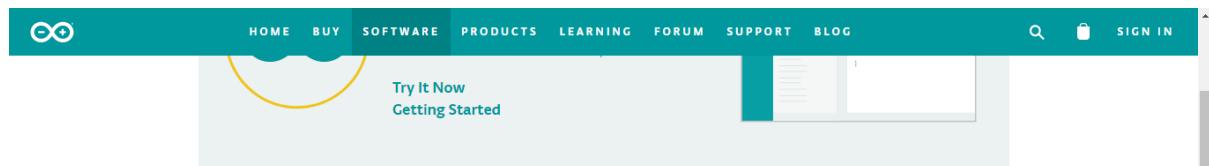
Hình 4.4: Giới thiệu Arduino

Hiện tại, cộng đồng Arduino Việt Nam và trên thế giới là rất lớn mạnh.

4.2 Cài đặt công cụ hỗ trợ

4.2.1 Arduino IDE

1. Truy cập vào địa chỉ <https://www.arduino.cc/en/Main/Software/>.
Bấm vào mục "Windows Installer, for Windows XP and up" để tải trực tiếp hoặc "Windows app" để tải thông qua Store.



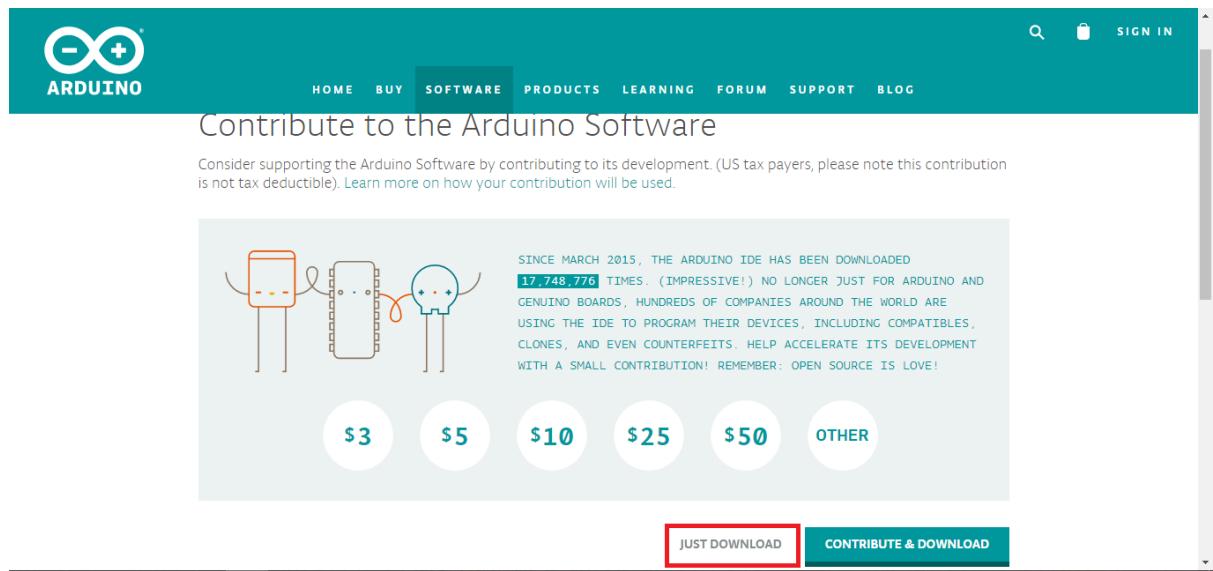
Download the Arduino IDE



Hình 4.5: Truy cập trang tải Arduino

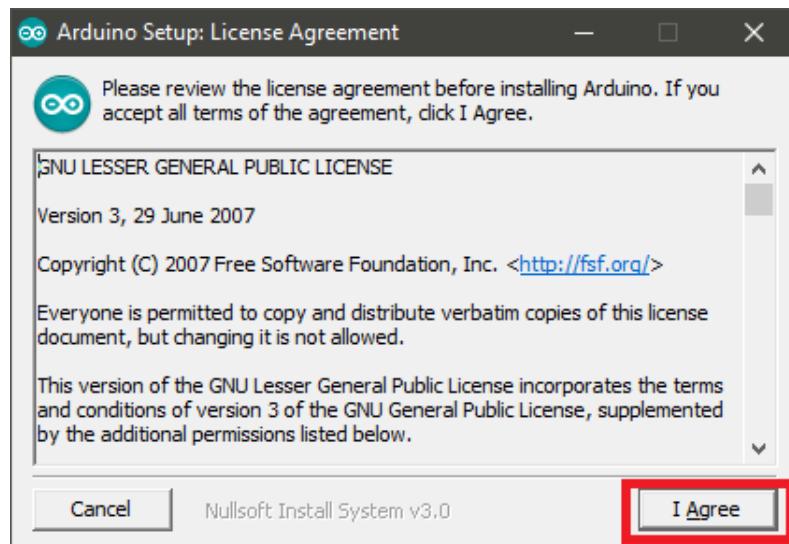
Đề cương luận văn 2018

2. Bấm "JUST DOWNLOAD" để tải bản cài đặt mới nhất cho Windows.



Hình 4.6: Chọn JUST DOWNLOAD

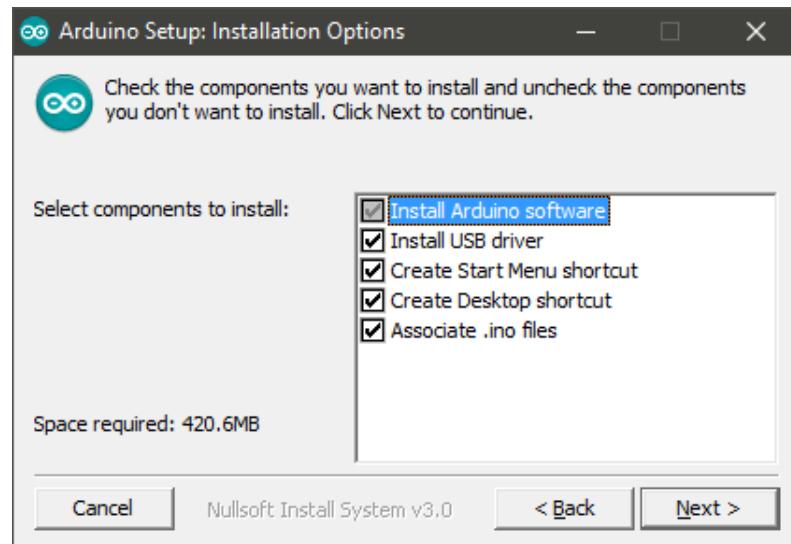
3. Tải về xong và khởi chạy file setup. Click "I Agree" để đồng ý các điều khoản.



Hình 4.7: Chọn I Agree

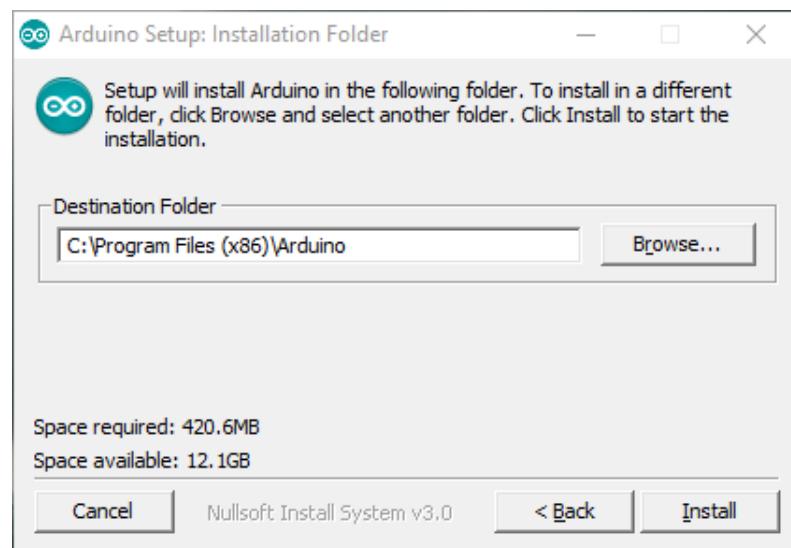
Đề cương luận văn 2018

- Chọn các công cụ cần cài đặt và "Next".



Hình 4.8: Chọn Next

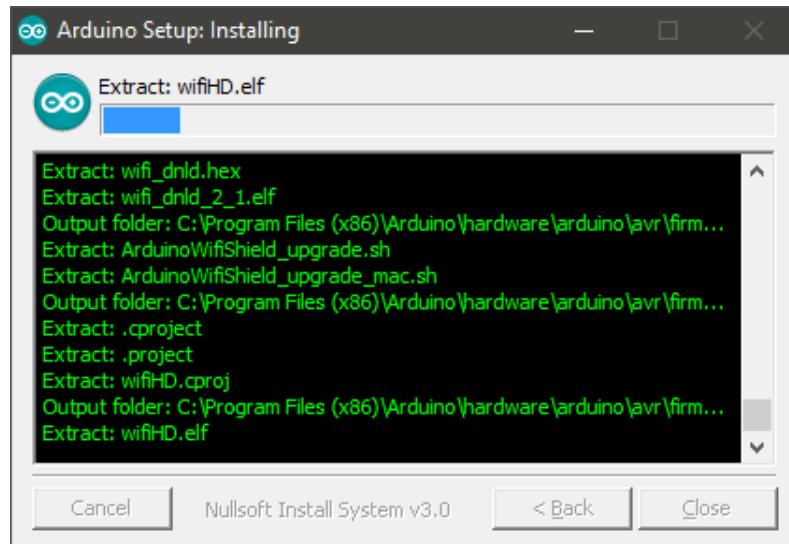
- Chọn thư mục cài đặt Arduino IDE, đề nghị nên giữ mặc định thư mục cài đặt. Chọn "Install".



Hình 4.9: Chọn Install

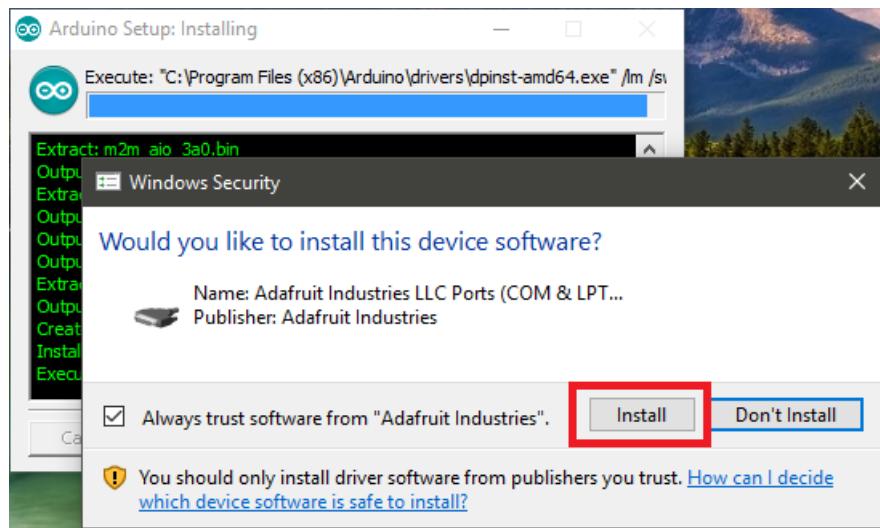
Đề cương luận văn 2018

- Chờ quá trình cài đặt hoàn tất.



Hình 4.10: Chờ đợi cài đặt Arduino IDE

- Khi cài gần xong, sẽ hiện các thông báo cài driver, bạn chỉ cần tích vào "Always trust software from 'Adafruit Industries'" và chọn "Install".

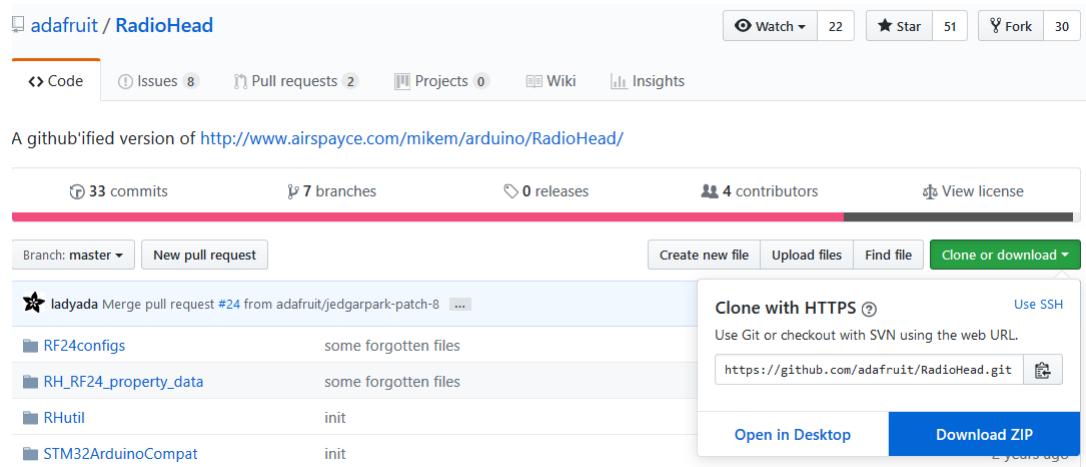


Hình 4.11: Chọn Install để cài đặt driver

- Sau khi việc cài đặt driver kết thúc là đã cài đặt thành công Arduino IDE trên Windows.

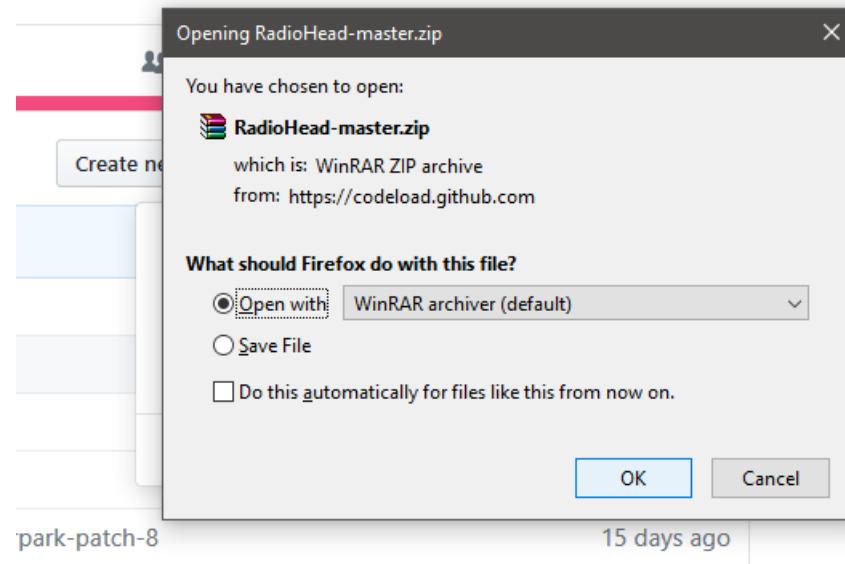
4.2.2 Thêm thư viện RadioHead

- Truy cập vào địa chỉ <https://github.com/adafruit/RadioHead>. Nhấp vào "Clone or download" → chọn "Download ZIP".



Hình 4.12: Tải tập thư viện từ Github

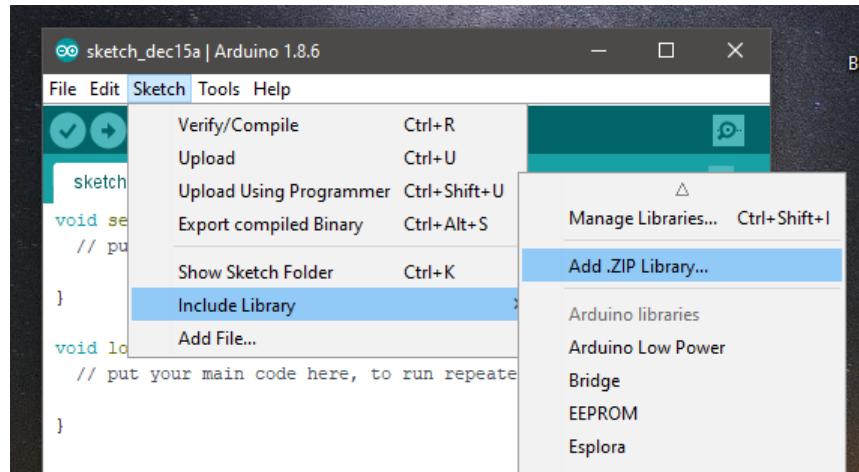
- Tải về tập tin "RadioHead-master.zip".



Hình 4.13: Nhấn OK để tải

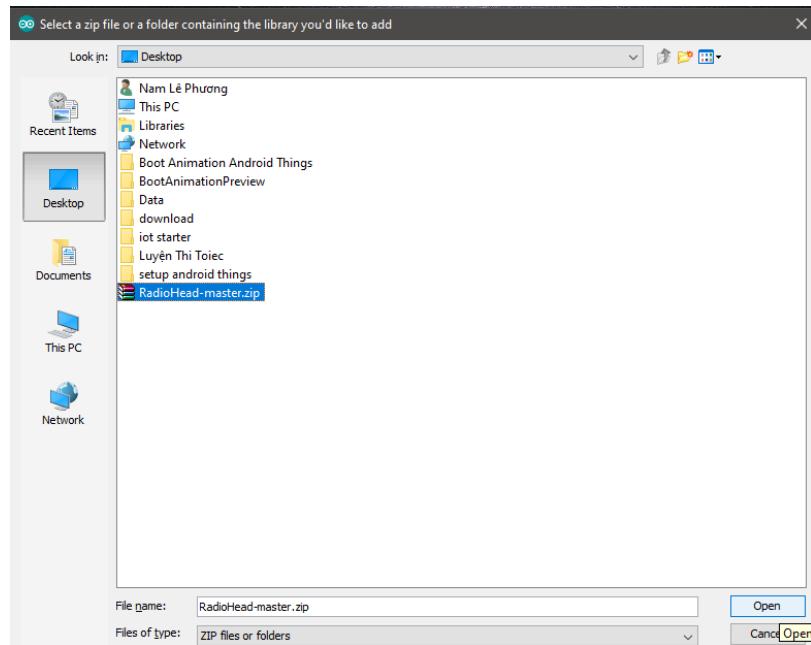
Đề cương luận văn 2018

3. Mở Arduino IDE, vào "Sketch" → "Include Library" → "Add .ZIP Library" để thêm tập tin thư viện.



Hình 4.14: Thêm thư viện cho Arduino IDE

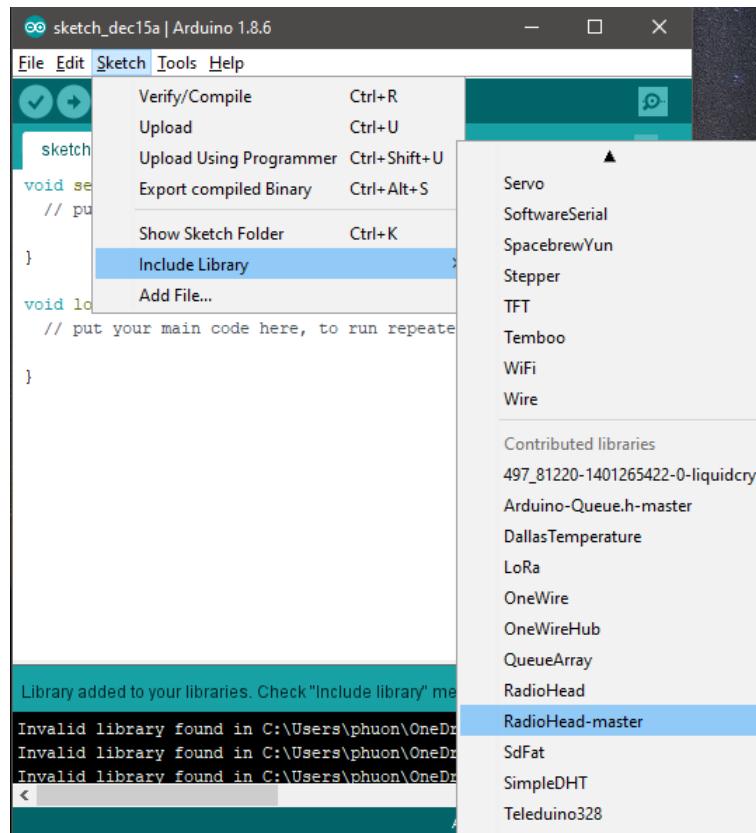
4. Chọn tập tin "RadioHead-master.zip" → nhấn "OK".



Hình 4.15: Chọn OK để thêm

Đề cương luận văn 2018

5. Vào "Sketch" → "Include Library" → tìm "RadioHead-master". Nếu đã có thì việc thêm thư viện đã hoàn tất.

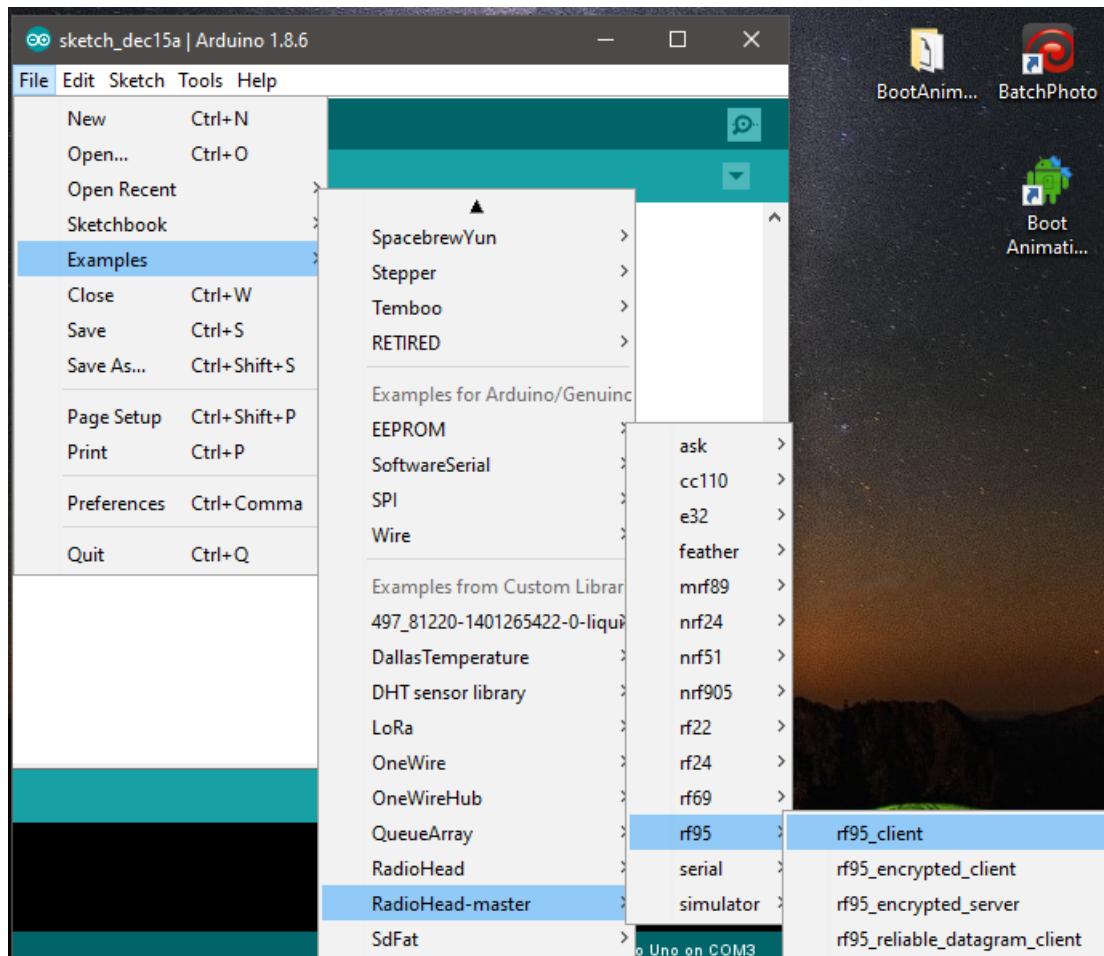


Hình 4.16: Kiểm tra thư viện đã được thêm

4.3 Hướng dẫn chi tiết

Nạp chương trình mẫu cho bên gửi gói tin (client).

1. Mở Arduino IDE, vào "File" → "Examples" → "RadioHead-master" → "rf95" → chọn "rf95_client" để mở code mẫu của bên gửi gói tin.



Hình 4.17: Mở code bên gửi "rf95_client"

Đề cương luận văn 2018

2. Nhấn dấu mũi tên ⇒ hoặc "CTRL + U" để Upload code lên mạch.

The screenshot shows the Arduino IDE interface. The top bar displays the title "rf95_client | Arduino 1.8.6". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, refresh, and other functions. The main code editor window contains the following sketch:

```
22 void setup()
23 {
24     // Rocket Scream Mini Ultra Pro with the RFM95W only:
25     // Ensure serial flash is not interfering with radio communication
26 //    pinMode(4, OUTPUT);
27 //    digitalWrite(4, HIGH);
28
29     Serial.begin(9600);
30     while (!Serial) ; // Wait for serial port to be available
31     if (!rf95.init())
32         Serial.println("init failed");
33     // Defaults after init are 434.0MHz, 13dBm, Bw = 125 kHz, Cr
34
35     // The default transmitter power is 13dBm, using PA_BOOST.
36     // If you are using RFM95/96/97/98 modules which uses the PA_
37     // you can set transmitter powers from 5 to 23 dBm:
38 //    driver.setTxPower(23, false);
39     // If you are using Modtronix inAir4 or inAir9, or any other radio
40     // transmitter RFO pins and not the PA_BOOST pins
```

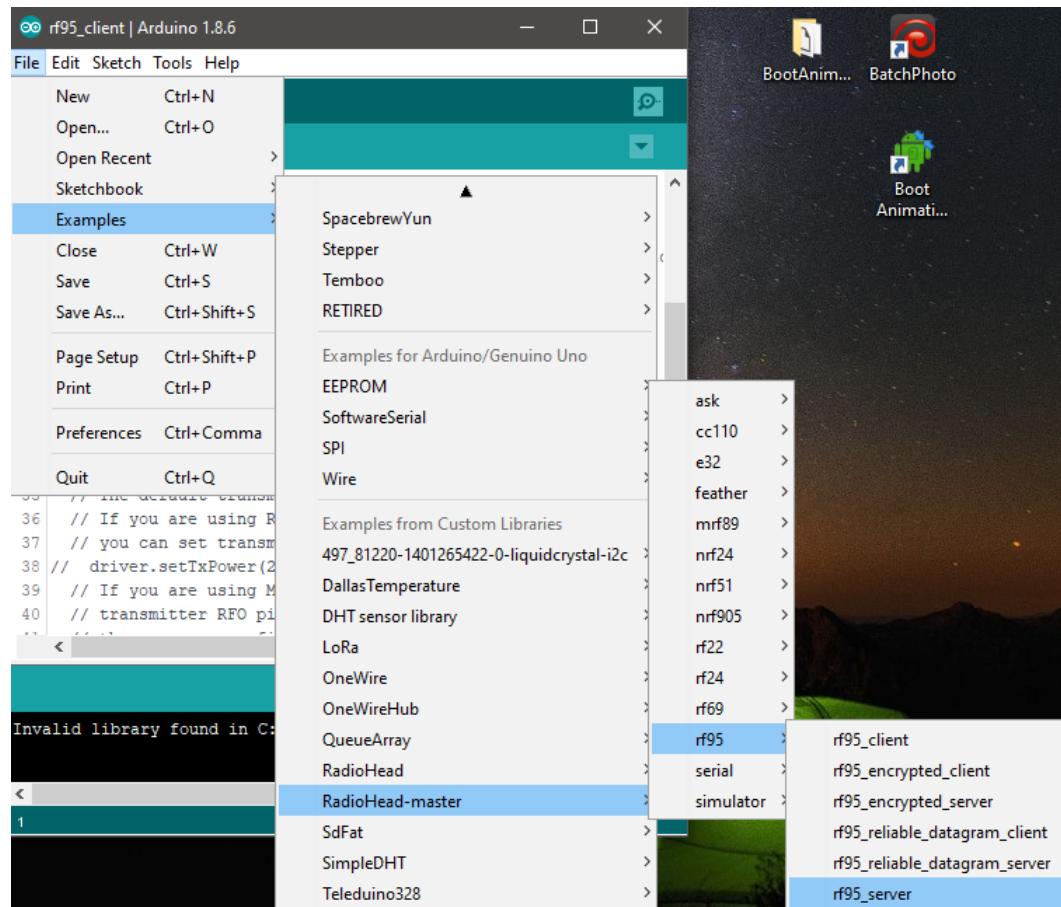
Below the code editor is a serial monitor window with the message: "Invalid library found in C:\Users\phuon\OneDrive\Documents\Arduino\libraries". The bottom status bar indicates "1" and "Arduino/Genuino Uno on COM3".

Hình 4.18: Nhấn dấu mũi tên ⇒ để Upload

Đề cương luận văn 2018

Nạp chương trình mẫu cho bên nhận gói tin (server).

1. Mở Arduino IDE, vào "File" → "Examples" → "RadioHead-master" → "rf95" → chọn "rf95_server" để mở code mẫu của bên nhận gói tin.



Hình 4.19: Mở code bên nhận "rf95_server"

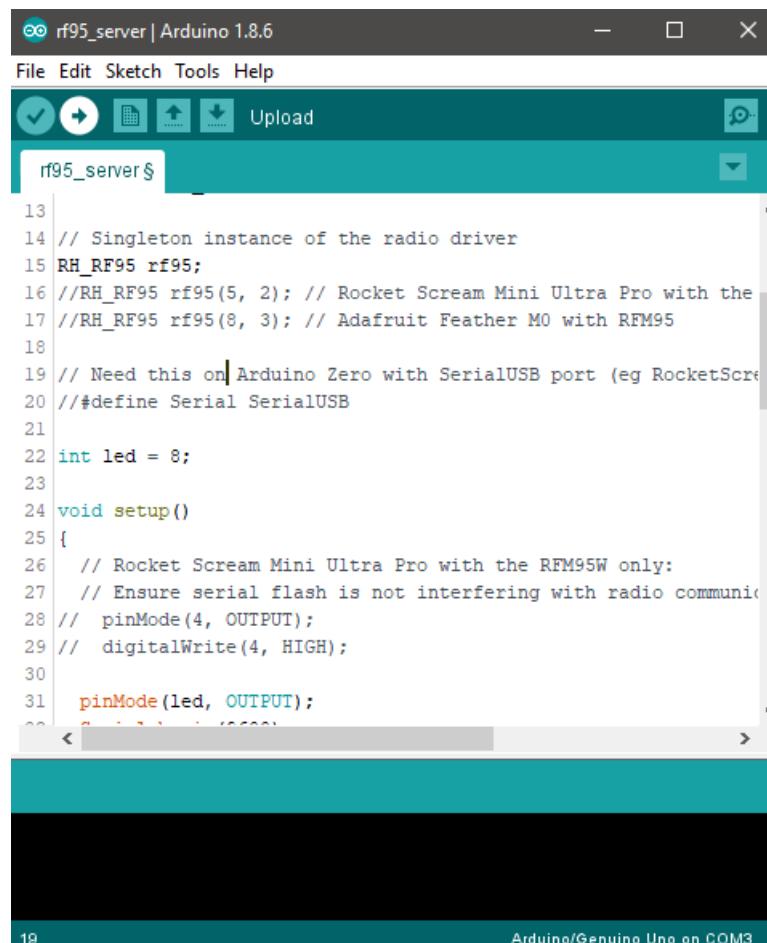
Đề cương luận văn 2018

2. Tại dòng thứ 22, đổi `int led = 9;` thành `int led = 8;` vì chân số 9 của LoRa Shield v1.3 là chân RESET.

```
rf95_server | Arduino 1.8.6
File Edit Sketch Tools Help
rf95_server§
13
14 // Singleton instance of the radio driver
15 RH_RF95 rf95;
16 //RH_RF95 rf95(5, 2); // Rocket Scream Mini Ultra Pro with the
17 //RH_RF95 rf95(8, 3); // Adafruit Feather M0 with RFM95
18
19 // Need this on Arduino Zero with SerialUSB port (eg RocketS
20 // #define Serial SerialUSB
21
22 int led = 8;
23
24 void setup()
25 {
26     // Rocket Scream Mini Ultra Pro with the RFM95W only:
27     // Ensure serial flash is not interfering with radio communic
28     // pinMode(4, OUTPUT);
29     // digitalWrite(4, HIGH);
30
31     pinMode(led, OUTPUT);
32 }
```

Hình 4.20: Thay đổi chân số 9 thành số 8

3. Nhấn dấu mũi tên ⇒ hoặc "CTRL + U" để Upload code lên mạch.



```
rf95_server | Arduino 1.8.6
File Edit Sketch Tools Help
Upload
rf95_server §
13
14 // Singleton instance of the radio driver
15 RH_RF95 rf95;
16 //RH_RF95 rf95(5, 2); // Rocket Scream Mini Ultra Pro with the
17 //RH_RF95 rf95(8, 3); // Adafruit Feather M0 with RFM95
18
19 // Need this on| Arduino Zero with SerialUSB port (eg RocketScre
20 // #define Serial SerialUSB
21
22 int led = 8;
23
24 void setup()
25 {
26     // Rocket Scream Mini Ultra Pro with the RFM95W only:
27     // Ensure serial flash is not interfering with radio communic
28     // pinMode(4, OUTPUT);
29     // digitalWrite(4, HIGH);
30
31     pinMode(led, OUTPUT);
32 }
```

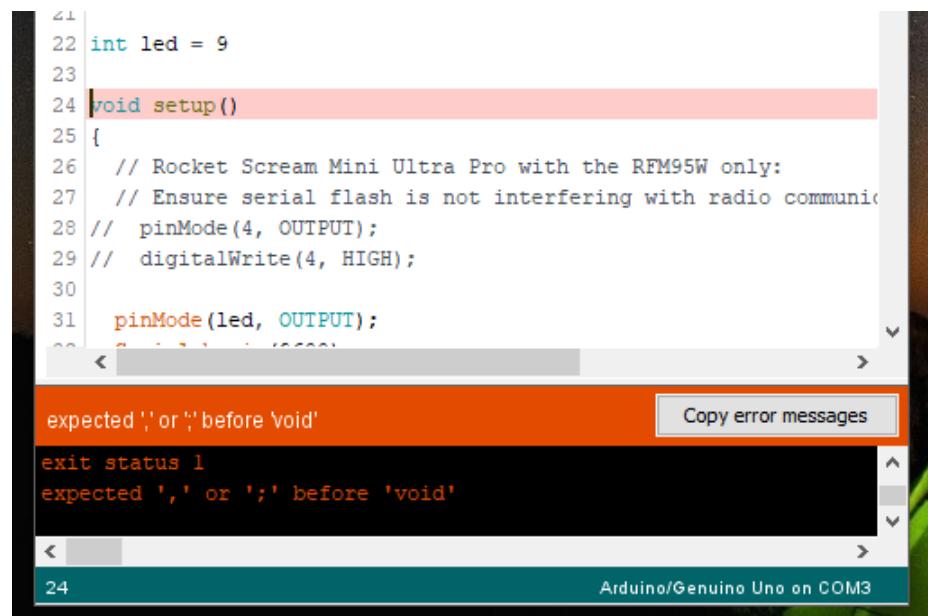
Hình 4.21: Nhấn dấu mũi tên ⇒ để Upload

4.4 Các lỗi thường gặp và cách khắc phục

1. Lỗi cú pháp.

Cách khắc phục: Đọc phần gợi ý màu cam của chương trình để xem và sửa lỗi.

Đề cương luận văn 2018



The screenshot shows the Arduino IDE interface. In the code editor, there is a syntax error highlighted in red. The error message 'expected ',' or ';' before 'void'' is displayed in an orange box. The code editor shows the following snippet:

```
41
42 int led = 9
43
44 void setup()
45 {
46     // Rocket Scream Mini Ultra Pro with the RFM95W only:
47     // Ensure serial flash is not interfering with radio communication
48     // pinMode(4, OUTPUT);
49     // digitalWrite(4, HIGH);
50
51     pinMode(led, OUTPUT);
52 }
```

In the status bar at the bottom, it says 'Arduino/Genuino Uno on COM3'.

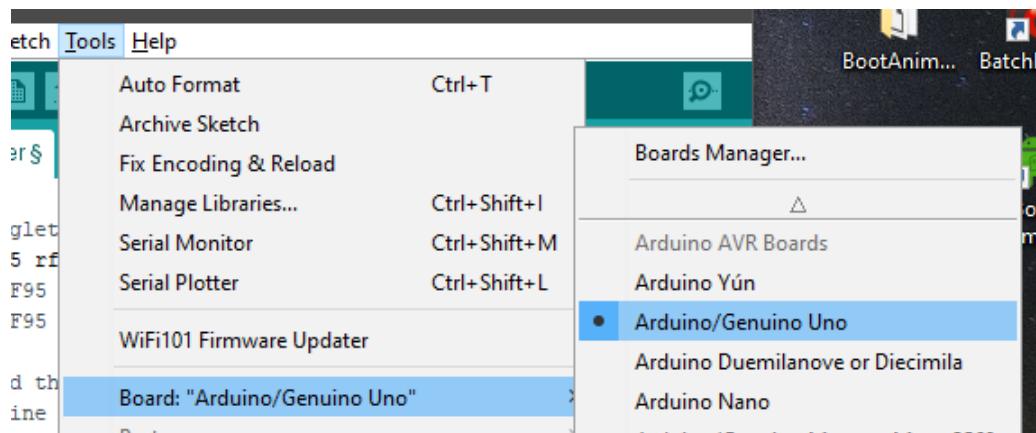
Hình 4.22: Xem phần thông báo lỗi

2. Lỗi logic.

Cách khắc phục: Xem xét lại tuần tự thực hiện lệnh, hướng đi đã hợp lý hay chưa.

3. Chọn sai board.

Cách khắc phục: Vào phần "Tools" → "Board" → kiểm tra board đã chọn là "Arduino/Genuino Uno".

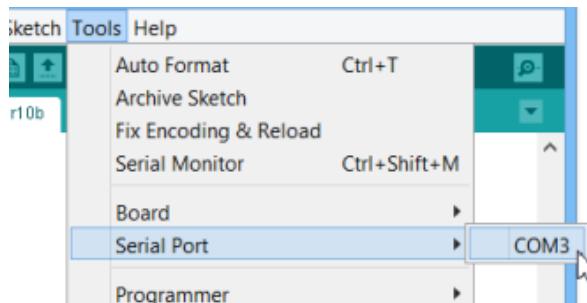


Hình 4.23: Kiểm tra "Tools" → "Board: Arduino/Genuino Uno"

Đề cương luận văn 2018

4. Chưa chọn Port.

Cách khắc phục: Vào phần "Tools" → "Port" hoặc "Serial Port" → kiểm tra đã chọn đúng Port chưa.

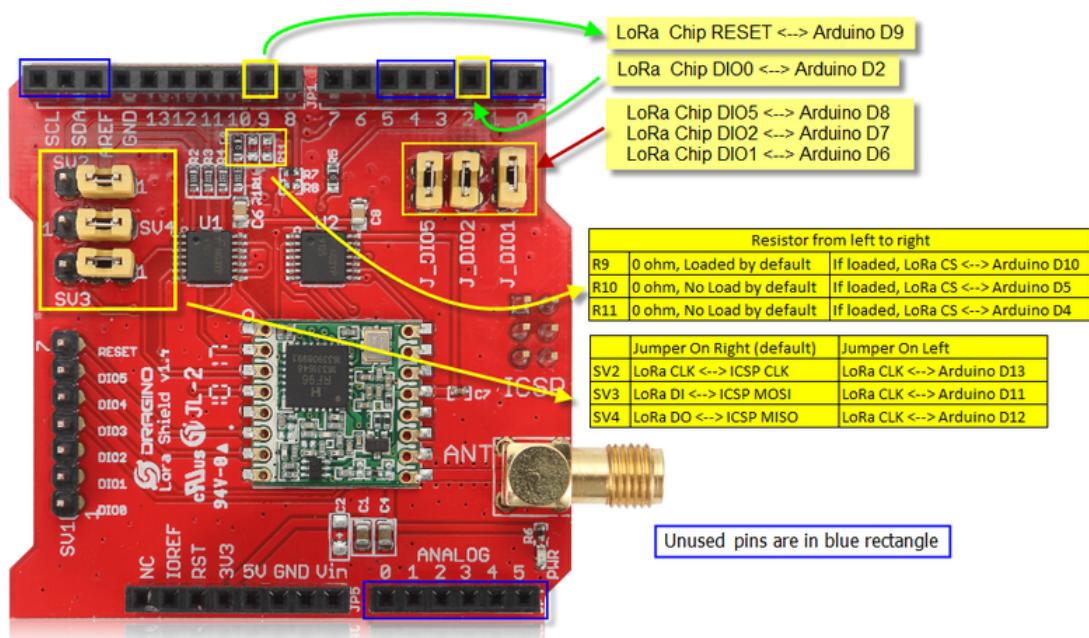


Hình 4.24: Kiểm tra "Tools" → "Port" hoặc "Serial Port"

5. Chưa nối chân jump trên mạch.

Cách khắc phục: Nối tất cả 6 chân jump như 4.25 (các chân được khoanh khung màu vàng) thì LoRa mới có thể hoạt động.

Pin Mapping For LoRa



Hình 4.25: Nối chân jump cho mạch LoRa

6. Sử dụng nhầm chân RESET

Cách khắc phục: Không sử dụng chân số 9 với LoRa Shield v1.3 kết hợp với Arduino Uno R3 vì đó là chân reset của LoRa.

7. Cấu hình sai tần số hoạt động hoặc sai chế độ truyền dữ liệu.

Cách khắc phục: Nếu không thiết lập tần số thì cấu hình mặc định cho tần số và chế độ truyền là `// Defaults after init are 434.0MHz, ← 13dBm, Bw = 125 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC on.`

Tuy nhiên, có thể thay đổi bằng việc sử dụng các lệnh:

Listing 4.1: Các lệnh để thiết lập cấu hình

```
1 setFrequency(434.0); //About 415~453MHz, default 434MHz
2 setTxPower(13); //Powers from 5 to 23 dBm, default 13dBm
3 setModemConfig(0); //Default 0
4 //0 < Bw = 125 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC ←
    on. Medium range
5 //1 < Bw = 500 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC ←
    on. Fast+short range
6 //2 < Bw = 31.25 kHz, Cr = 4/8, Sf = 512chips/symbol, CRC←
    on. Slow+long range
7 //3 < Bw = 125 kHz, Cr = 4/8, Sf = 4096chips/symbol, CRC ←
    on. Slow+long range
```

Tham khảo các thông số cấu hình tại đây [17]

8. Kích thước dữ liệu quá lớn.

Cách khắc phục: Nên lưu ý về kích thước tối đa của một gói tin là 256 bytes, trừ đi 4 bytes đầu (chứa các thông tin khác của gói tin) sẽ còn lại 252 bytes dành cho dữ liệu nội dung.

9. Ký tự kết thúc chuỗi dữ liệu

Cách khắc phục: Thường thì ký tự kết thúc chuỗi sẽ là '\0' hoặc

'\r' hoặc '\n'. Nên để ý khi thêm hoặc phân tích dữ liệu chuỗi trong một gói tin.

10. Kiểu dữ liệu.

Cách khắc phục: Do LoRa hoạt động trên thanh ghi, nên việc quản lý sẽ dựa vào chuỗi bit (từng byte). Vì thế tất cả các dữ liệu muốn gửi đi đều phải chuyển hóa thành từng byte hoặc chuỗi char hoặc kiểu số nguyên int.

11. Trùng gói tin khi có nhiều thiết bị gửi dữ liệu.

Cách khắc phục: Sử dụng các giao thức để quản lý như "Stop-n-wait", hàng đợi "Queue" .. và các giao thức quản lý khác giống trong network.

12. Tham khảo cách khắc phục các lỗi thường gặp khác tại "<http://arduino.vn/reference/cac-loi-thuong-gap-tren-Arduino-va-cach-khac-phuc>".

4.5 Đo đặc hiệu suất của hệ thống

4.5.1 Khoảng cách gửi nhận dữ liệu

Dựa trên kết quả đo đặc thực tế, các thông số cấu hình đều sử dụng mặc định.

Lần	Điểm cố định	Điểm di động	Khoảng cách ước tính (m)
1	105C6-CS1	Solar Cafe-CS1	100
2	105C6-CS1	Sân banh C2-CS1	150
3	420AH1-KTXA	Phía sau KTXXHH	205
4	420AH1-KTXA	Sau nhà AG4-KTXA	234
5	911AH1-KTXA	Tòa nhà H2-CS2	395
6	413AH1-KTXA	Tòa nhà H1-CS2	436
7	611H6-CS2	Tòa nhà AH1-KTXA	348
8	611H6-CS2	Nhà Thi Đấu-CS2	264
9	709H6-CS2	Tòa nhà AH1-KTXA	378
10	709H6-CS2	Nhà Thi Đấu-CS2	260
11	709H6-CS2	Đường A4	430
12	Sảnh tầng 7 H6-CS2	Đại Học Quốc Tế	432

Bảng 4.1: Khoảng cách gửi nhận dữ liệu đo đặc được

Từ bảng 4.1, trừ hao tổn năng lượng do vật cản (nhà, cây cối..), ước lượng khoảng cách trung bình mà LoRa có thể gửi nhận với thông số cấu hình mặc định là **300m**, trong khoảng **250m - 350m**.

Khoảng cách có thể tăng thêm hoặc giảm nếu có sự thay đổi về cấu hình truyền nhận của LoRa.

4.5.2 Ước lượng thời gian hoạt động của node cảm biến

Thời gian hoạt động có thể lên đến 10 năm khi sử dụng pin. [18]

CHƯƠNG 5

HIỆN THỰC NHẬN DỮ LIỆU CHO GATEWAY

5.1 Giới thiệu

Sử dụng công nghệ truyền thông không dây LoRa [12] (Long Range Radio) và nền tảng Android Things cho gateway.



Hình 5.1: Gateway LoRa

5.1.1 LoRa

Xem thêm LoRa - Mục 4.1.1

5.1.2 Android Things

Xem thêm Chương 3

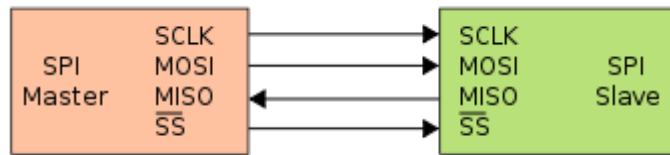
5.2 Phát triển driver cho gateway: SPI

5.2.1 Giới thiệu SPI

SPI viết tắt của Serial Peripheral Interface [10], SPI bus – Giao diện ngoại vi nối tiếp, bus SPI. Chuẩn SPI được phát triển bởi Motorola. Đây là một chuẩn đồng bộ nối tiếp để truyền dữ liệu ở chế độ song công toàn phần (full-duplex) tức trong cùng một thời điểm có thể xảy ra đồng thời quá trình truyền và nhận. Đôi khi SPI còn được gọi là chuẩn giao tiếp 4 dây (Four-wire).

SPI là giao diện đồng bộ, bất cứ quá trình truyền nào cũng được đồng bộ hóa với tín hiệu clock chung. Tín hiệu này sinh ra bởi master.

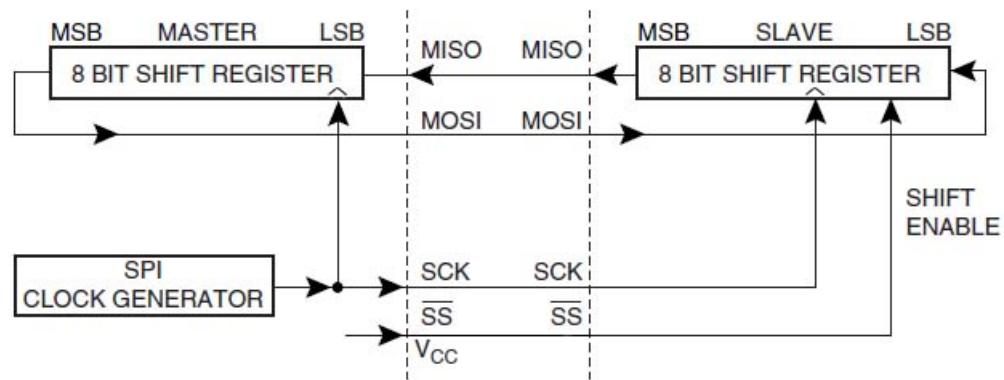
Trong giao diện SPI có bốn tín hiệu số:



Hình 5.2: Khái niệm SPI

- MOSI hay SI – cổng ra của bên Master (Master Out Slave IN). Đây là chân dành cho việc truyền tín hiệu từ thiết bị chủ động đến thiết bị bị động.
- MISO hay SO – Công ra bên Slave (Master IN Slave Out). Đây là chân dành cho việc truyền dữ liệu từ Slave đến Master.
- SCLK hay SCK là tín hiệu clock đồng bộ (Serial Clock). Xung nhịp chỉ được tạo bởi Master.
- CS hay SS là tín hiệu chọn vi mạch (Chip Select hoặc Slave Select). SS sẽ ở mức cao khi không làm việc. Nếu Master kéo SS xuống thấp

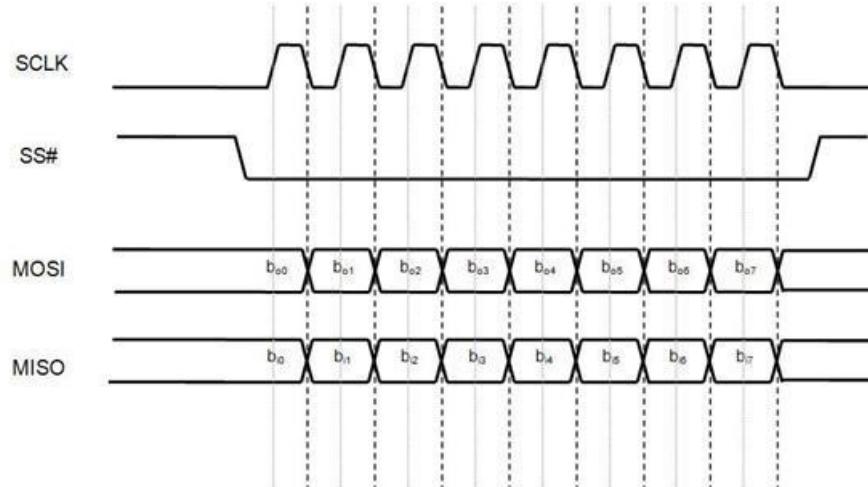
thì sẽ xảy ra quá trình giao tiếp. Chỉ có một đường SS trên mỗi slave nhưng có thể có nhiều đường điều khiển SS trên master, tùy thuộc vào thiết kế của người dùng.



Hình 5.3: Sơ đồ đường đi của SPI

5.2.2 Nguyên lý hoạt động

Để bắt đầu hoạt động thì kéo chân SS xuống thấp và kích hoạt clock ở cả Master và Slave.



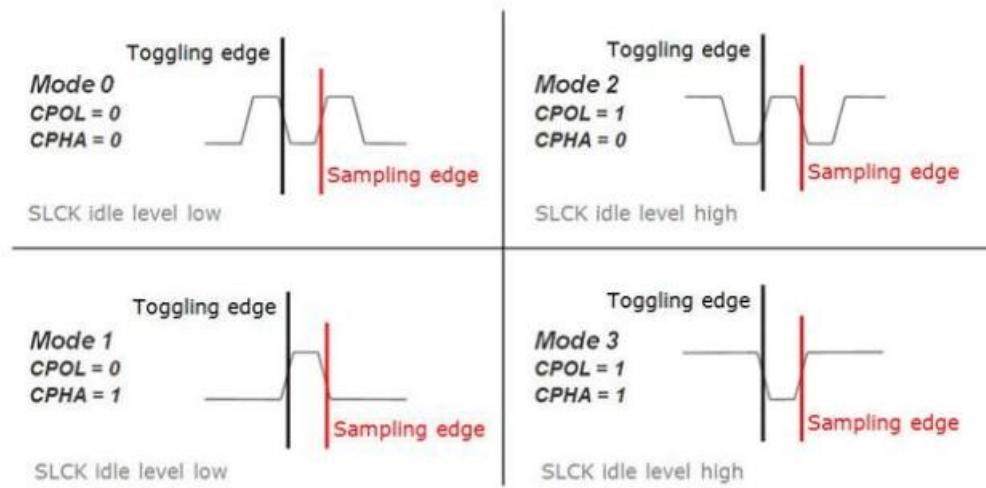
Hình 5.4: Nguyên lý hoạt động của SPI

Mỗi chip Master hay Slave có một thanh ghi dữ liệu 8 bits.

Cứ mỗi cùa xung nhịp do Master tạo ra trên đường giũ nhịp SCK, một bit trong thanh ghi dữ liệu của Master được truyền qua Slave trên đường MOSI, đồng thời một bit trong thanh ghi dữ liệu của chip Slave cũng được truyền qua Master trên đường MISO.

Lưu ý, có thể config tín hiệu đồng bộ clock theo sườn, theo mức

Hiện tại có 4 mode cơ bản (MODE 0, 1, 2, 3) của SPI dựa vào config SCLK như sau:



Hình 5.5: Dạng sóng của 4 mode SPI

Cực của xung giữ nhịp, phase và các chế độ hoạt động: cực của xung giữ nhịp (Clock Polarity) được gọi tắt là CPOL .Đây là khái niệm dùng chỉ trạng thái của chân SCK ở trạng thái nghỉ.

Ở trạng thái nghỉ (Idle), chân SCK có thể được giữ ở mức cao (CPOL=1) hoặc thấp (CPOL=0).

Phase (CPHA) dùng để chỉ cách mà dữ liệu được lấy mẫu (sample) theo xung giữ nhịp.

Dữ liệu có thể được lấy mẫu ở cạnh lên của SCK (CPHA=0) hoặc cạnh xuống (CPHA=1).

SPI Mode	SCK (Cạnh bắt đầu)	SCK (Cạnh kết thúc)
0	Lấy mẫu dữ liệu tại cạnh lên	Truyền dữ liệu mới tại cạnh xuống
1	Truyền dữ liệu mới tại cạnh lên	Lấy mẫu dữ liệu tại cạnh xuống
2	Lấy mẫu dữ liệu tại cạnh xuống	Truyền dữ liệu mới tại cạnh lên
3	Truyền dữ liệu mới tại cạnh xuống	Lấy mẫu dữ liệu tại cạnh lên

Bảng 5.1: Các mode trong giao tiếp SPI [11]

5.2.3 Driver SPI của Raspberry Pi 3 và LoRa HAT

Sử dụng ngôn ngữ JAVA để thiết kế Driver SPI trên nền tảng Andoird Things.

5.2.3.1 Cấu hình kết nối SPI

Dựa vào Datasheet LoRa của SEMTECH, có thể xác định được các thông số sau:

- SPI Mode: MODE 1.
- SPI Frequency: 32,000,000Hz (32MHz).
- SPI First Bit: Bit MSB - bit có trọng số cao.
- SPI Bits Per Word: 8 bits.

Listing 5.1: Cấu hình SPI cho việc giao tiếp

```
1 void configSPIDevice(SpiDevice device) throws IOException {  
2     device.setMode(SpiDevice.MODE1);  
3     device.setFrequency(32000000); // 32MHz  
4     device.setBitJustification(SpiDevice.←  
5         BIT_JUSTIFICATION_MSB_FIRST);  
6     device.setBitsPerWord(8);  
7     Log.d(TAG, "SPI OK now ....");  
}
```

5.2.3.2 Khởi động giao tiếp giữa LoRa và Rpi3

Đầu tiên, cần chạy lệnh `begin()`:

1. Tín hiệu tại chân RESET tích cực thấp để thiết lập giá trị của các thanh ghi về 0x00.

2. Sau đó sẽ đưa LoRa vào trạng thái ngủ (Sleep mode: 0x00) để tiến hành thiết lập các thông số cần thiết.

Việc thiết lập trạng thái hoạt động của LoRa phải thực hiện lệnh GHI (`writeRegister()`) lên thanh ghi REG_OP_MODE, địa chỉ 0x01.

3. Thiết lập tần số hoạt động (thường dùng 434MHz), giá trị thanh ghi TX (gửi), RX (nhận) về 0x00, năng lượng truyền (thường dùng 13dBm),...
4. Sau khi thiết lập xong, chuyển LoRa vào trạng thái chờ (chuẩn bị hoạt động - Standby mode: 0x01).

Listing 5.2: Khởi động việc điều khiển board LoRa

```
1 public int begin(long frequency) throws IOException {
2
3     // perform reset
4     digitalWrite(pinReset, LOW);
5     delay(10);
6     digitalWrite(pinReset, HIGH);
7     delay(10);
8
9     // start SPI
10    configSPIDevice(mDevice);
11
12    // check version
13    byte version = readRegister(REG_VERSION);
14    if (version != 0x12) {
15        return 0;
16    }
17
18    // put in sleep mode to set LoRa mode
19    sleep();
```

```
20
21     // set frequency, usually 434MHz
22     setFrequency(frequency);
23
24     // set base addresses
25     writeRegister(REG_FIFO_TX_BASE_ADDR, (byte) 0);
26     writeRegister(REG_FIFO_RX_BASE_ADDR, (byte) 0);
27
28     // set LNA boost
29     writeRegister(REG_LNA, (byte) (readRegister(REG_LNA) | 0←
30                           x03));
31
32     // set auto AGC
33     writeRegister(REG_MODEM_CONFIG_3, (byte) 0x04);
34
35     // set output power to 17 dBm
36     setTxPower(13, PA_OUTPUT_PA_BOOST_PIN);
37
38     // put in standby mode to start working
39     idle();
40
41 }
```

Để gửi gói tin, dùng lệnh `LoRaSender(string);`.

1. Đầu tiên, LoRa sẽ kiểm tra có đang trong trạng thái gửi gói tin nào không. Nếu không thì một gói tin mới sẽ được tạo ra với kích thước là 0 (REG_PAYLOAD_LENGTH = 0x00).
2. Nội dung cần được gửi đi sẽ thêm vào gói tin vừa được tạo bằng lệnh `write(string)`.
3. LoRa chuyển sang chế độ gửi tin (MODE_TX: 0x03). Gói tin được

Đề cương luận văn 2018

gửi đi. LoRa sẽ chờ cho đến khi gói tin hoàn toàn gửi xong mới kết thúc quá trình gửi.

Listing 5.3: Hàm gửi gói tin

```
1 public void LoRaSender(String string){  
2     beginPacket(0);  
3     write(string.getBytes());  
4     endPacket(false);  
5     delay(5000);  
6 }
```

Để nhận gói tin, dùng lệnh `LoRaReceive()`.

1. Hàm `parsePacket()` sẽ chuyển LoRa về chế độ nhận gói tin (MODE_RX_SINGLE: 0x06).
2. Nếu có tin hiệu của gói tin đang được gửi đến (thường là Header), tiếp tục chuyển sang chế độ MODE_RX_CONTINUOUS: 0x05 để nhận tất cả phần còn lại của gói tin.
3. Toàn bộ dữ liệu của gói tin vừa nhận được lưu ở thanh ghi REG_FIFO (địa chỉ 0x00). Nội dung sẽ được đọc lên bởi lệnh `read()`, dữ liệu được lưu ở dạng "byte" nên cần phải dùng kiểu "char" để chuyển thành ngôn ngữ bình thường.

Listing 5.4: Hàm nhận gói tin

```
1 public void LoRaReceive(){  
2     int packetSize = parsePacket(0);  
3     if (packetSize > 0) {  
4         String dataTemp = "";  
5         // received a packet  
6         dataTemp += "Received packet '";
```

Đề cương luận văn 2018

```
7         // read packet
8         while (available() > 0) {
9             dataTemp += (char)read();
10        }
11        // print string
12        Log.d(TAG, dataTemp);
13    }
14 }
```

Khi không cần sử dụng nữa, chỉ cần dùng lệnh `end()`; để ngắt kết nối giao tiếp SPI giữa mạch Rpi3 và LoRa HAT.

Listing 5.5: Kết thúc việc điều khiển board LoRa

```
1 public void end(){
2     if (mDevice != null) {
3         try {
4             mDevice.close();
5             mDevice = null;
6         } catch (IOException e) {
7             Log.w(TAG, "Unable to close SPI device", e);
8         }
9     }
10 }
```

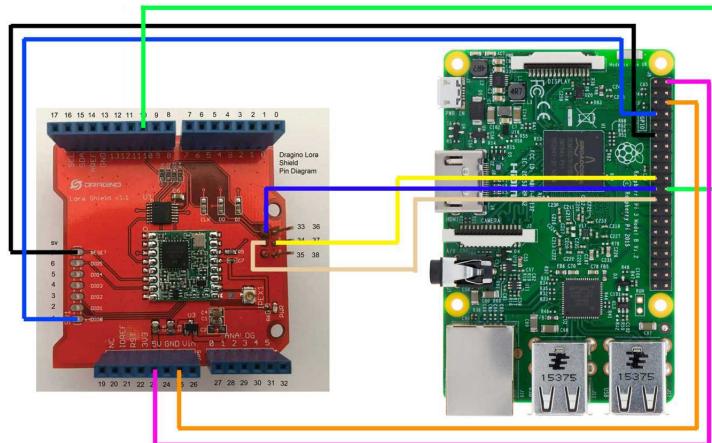
[GITHUB] <https://github.com/phuongnam0907/ConfigLoRa>

CHƯƠNG 6

TỔNG KẾT

6.1 Các kết quả đạt được trong giai đoạn thực tập

6.1.1 Thực hiện kết nối giữa raspberry và Lora

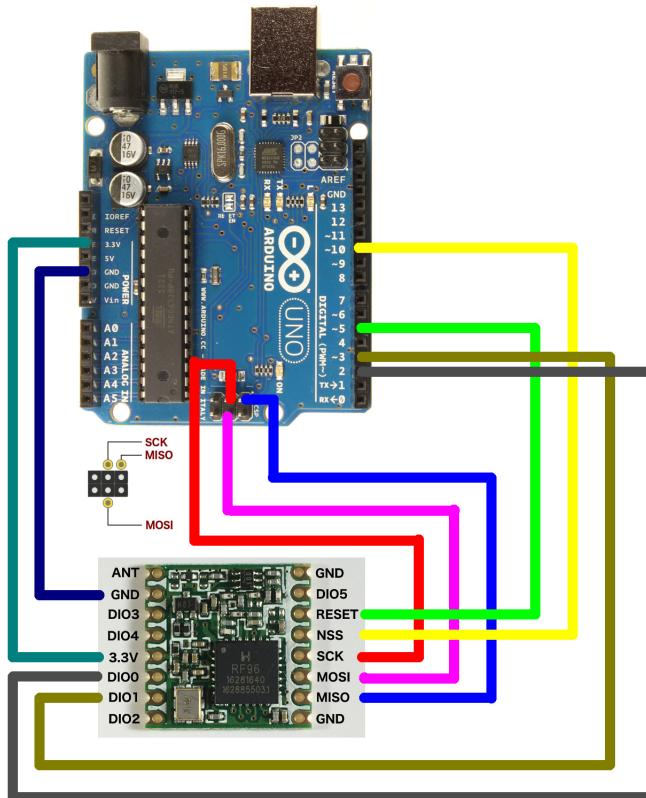


Hình 6.1: Kết nối giữa Lora và raspberry

Gateway bao gồm raspberry kết nối với Lora. Lora có nhiệm vụ kết nối với các node khác bằng sóng radio,Lora sẽ nhận tín hiệu từ các node xung quanh gửi về hoặc nó gửi giữ liệu đến các node xung quanh.Raspberry có nhiệm vụ xử lý các dữ liệu mà lora nhận được hoặc gửi đi.Raspberry kết nối với internet gửi dữ liệu lên server.

Nhóm đã kết nối giữa Lora và Raspberry thông qua giao thức SPI, dữ liệu có gửi và nhận giữa hai thiết bị.

6.1.2 Thực hiện kết nối giữa arduino và Lora



Hình 6.2: Kết nối giữa Lora và arduino

Node bao gồm arduino kết nối với Lora. Lora có nhiệm vụ kết nối với các node khác bằng sóng radio, Lora sẽ nhận tín hiệu từ các node xung quanh gửi về hoặc nó gửi giữ liệu đến các node xung quanh. Arduino có nhiệm vụ xử lý các dữ liệu mà lora nhận được hoặc gửi đi.

Nhóm đã kết nối giữa Lora và arduino thông qua giao thức SPI trong thư viện có sẵn của arduino, dữ liệu có gửi và nhận giữa hai thiết bị. Hiện tại nhóm đang viết lại thư viện cho arduino và Lora

6.1.3 Truyền nhận giữ liệu ở các node Lora

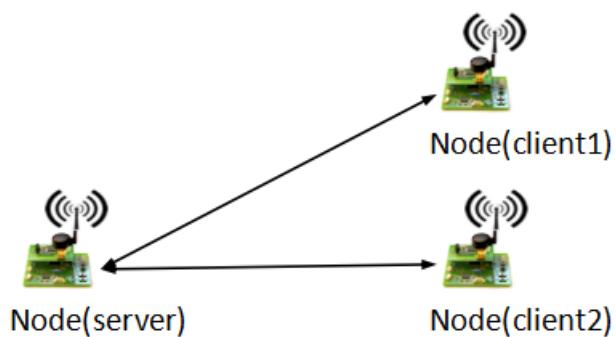
Nhóm thực hiện truyền nhận dữ liệu của các node trong các trường hợp

- Thực hiện truyền nhận giữa một node server và node client.



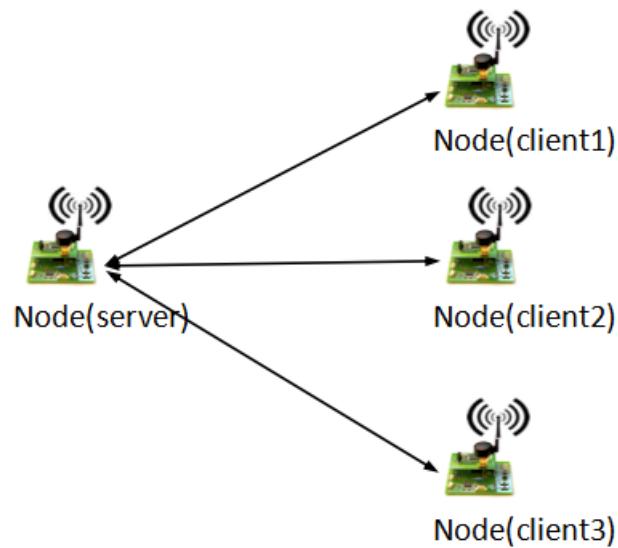
Hình 6.3: Tuyền dữ liệu giữa 2 node

2. Thực hiện kết nối giữa 1 node server và 2 node client



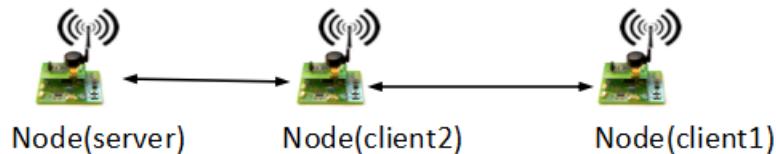
Hình 6.4: Tuyền dữ liệu giữa 3 node

3. Thực hiện kết nối giữa 1 node server và 3 node client Trong trường hợp này gặp vấn đề là dữ liệu bị mất do có quá nhiều dữ liệu gửi đến và nhóm đã khác phục được



Hình 6.5: Tuyễn dữ liệu giữa 3 node

4. Thực hiện truyền qua các node trung gian



Hình 6.6: Tuyễn dữ liệu qua node trung gian

6.2 Hướng phát triển trong giai đoạn luận văn

6.2.1 Hiện thực server

- Trước tiên nhóm sẽ dùng các server co sẵn để thực hiện ví dụ MQTT và Firebase
- Sau đó nhóm sẽ thực hiện server riêng cho mình như my sql...

6.2.2 Hiện thực ứng dụng người dùng bằng ngôn ngữ hai nền tảng

Nhóm sẽ viết phần mềm để người dùng có thể theo dõi được giữ liệu, phần có thể được phát triển ở dạng wed đối với máy tính và app đối với điện thoại.

Đối với điện thoại, nhóm sẽ viết trên hai nền tảng là android và IOS. Vì đây là hai nền tảng phổ biến nhất hiện tại.

Dữ liệu sẽ được máy tính và điện thoại lấy từ server, cấu trúc dữ liệu bao gồm:

1. Nhiệt độ
2. Độ ẩm
3. Thời gian
4. Địa chỉ ID

Dữ liệu sẽ được lưu trữ và được vẽ biểu đồ để người sử dụng có thể dễ quan sát và đưa ra nhận định.

6.2.3 Thực hiện kết nối nhiều node

Nhóm sẽ thực hiện kết nối nhiều node lora trong tự nhiên.

Đo xem giá trị cường độ tín hiệu nào phù hợp để kết nối.

Tìm thêm các thiết bị giúp sóng truyền xa hơn.

tìm hiểu về nguồn năng lượng mà các node sẽ sử dụng.

Thiết kế vỏ hộp cho các node.

TÀI LIỆU THAM KHẢO

- [1] DTT. Internet of things là gì? [online], viewed 25 September 2018, from: <<http://iot.dtt.vn/InternetofThings.html>>.
- [2] Ho Nguyen (May 2018). Định nghĩa khái niệm Internet of Things (IoT) - Internet Vạn Vật [online], viewed 25 September 2018, from: <<https://blog.trginternational.com/vi/dinh-nghia-khai-niem-internet-of-things-iot-internet-van-vat>>.
- [3] Tỷ Phú (April 2018). Android Studio là gì? [online], viewed 1 October 2018, from: <<https://quanzhixiang.com/android-studio-la-gi-149713>>.
- [4] Eric Lamarre, McKinsey & Company (May 2017). Making sense of Internet of Things platforms [online], viewed 1 October 2018, from: <<https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/making-sense-of-internet-of-things-platforms>>.
- [5] Thạch An (August 2016). Những hệ điều hành cho IoT trong tương lai [online], viewed 1 October 2018, from: <<http://www.pcworld.com.vn/articles/cong-nghe/cong-nghe/2016/08/1249319/nhung-he-dieu-hanh-danh-cho-iot-trong-tuong-lai/>>.
- [6] Bùi Danh Nam (May 2018). Giới thiệu về Android Things 1.0 [online], viewed 1 October 2018, from: <<https://viblo.asia/p/gioi-thieu-ve-android-things-10-maGK7M4xlj2>>.

- [7] Lưu Hoàng Trúc (December 2016). [Android Things] Phần 1: Tổng quan về IoT và AT - Cài đặt môi trường cho Android Things với kit Raspberry Pi 3 [online], viewed 2 October 2018, from: <<https://viblo.asia/p/android-things-phan-1-tong-quan-ve-iot-va-at-cai-dat-moi-truong-cho-android-things-voi-kit-raspberry-pi-3-jvlKaqqDKVr>>.
- [8] Paul Trebilcox-Ruiz (January 2017), Vietnamese translation by Dai Phong. Giới thiệu về Android Things [online], viewed 2 October 2018, from: <<https://code.tutsplus.com/vi/articles/introduction-to-android-things--cms-27892>>.
- [9] Duy Luân (February 2018). Discussion in "Thông tin công nghệ": Cách mạng công nghiệp 4.0 là gì, nó ảnh hưởng tới bản thân các bạn ra sao? [online], viewed 5 October 2018, from: <<https://tinhte.vn/threads/cach-mang-cong-nghiep-4-0-la-gi-no-anh-huong-toi-ban-than-cac-ban-ra-sao.2770055/>>.
- [10] Trung Kiên (April 2015). Chuẩn giao tiếp SPI [online], viewed 10 October 2018, from: <<https://kienltb.wordpress.com/2015/04/05/chuan-giao-tiep-spi/>>.
- [11] Ý Tưởng Nhanh (2017). Giao tiếp SPI với vi điều khiển PIC (Phần 1) [online], viewed 12 October 2018, from: <<http://ytuongnhanh.vn/chitiet/giao-tiep-spi-voi-vi-dieu-khien-pic-phan-1-144.html>>.
- [12] AloChip. CÔNG NGHỆ TRUYỀN THÔNG KHÔNG DÂY LORA [online], viewed 15 October 2018, from: <https://alochip.com/blog/cong-nghe-truyen-thong-khong-day-lora_8.html>.
- [13] BKAI. 10 ứng dụng thế giới thực của Internet of Things [online], viewed 1 November 2018, from: <<https://bkaii.com.vn/tin-tuc/222-10-ung-dung-the-gioi-thuc-cua-internet-of-things>>.

Đề cương luận văn 2018

- [14] LoRa Alliance™. About the LoRaWAN™ Specification [online], viewed 20 November 2018, from: <<https://lora-alliance.org/lorawan-for-developers>>.
- [15] MAKER.IO STAFF (October 2016). Introduction to LoRa Technology – The Game Changer [online], viewed 24 November 2018, from: <<https://www.digikey.com/en/maker/blogs/introduction-to-lora-technology>>.
- [16] Hunghv (October 2017). Arduino là gì? Lập trình Arduino bằng C/C++ [online], viewed 18 November 2018, from: <<http://hanhtranglaptrinh.vn6.vn/arduino-la-gi-lap-trinh-arduino-bang-c-c-plus-plus>>.
- [17] Semtech Corporation (July 2013). SX1272/3/6/7/8: LoRa Modem - Designer's Guide [online], viewed 2 November 2018, from: <https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf>.
- [18] Jhon_Control (2017). Introduction LoRa & Module RFM95/RFM95W Hoperf [online], viewed 7 December 2018, from: <<https://www.instructables.com/id/Introduction-LoRa-Module-RFM95-RFM95W-Hoperf/>>.