1TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**CHUYÊN ĐỀ .NET**

**PHÁT TRIỂN ỨNG DỤNG QUẢN LÝ QUÁN ĂN**

*Người hướng dẫn*: **Thầy MAI VĂN MẠNH**

*Người thực hiện*: **Phan Tuấn Trọng – 51203152**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2018**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**CHUYÊN ĐỀ .NET**

**PHÁT TRIỂN ỨNG DỤNG QUẢN LÝ QUÁN ĂN**

*Người hướng dẫn*: **Thầy MAI VĂN MẠNH**

*Người thực hiện*: **Phan Tuấn Trọng – 51203152**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2018**

LỜI CẢM ƠN

Trước tiên, em xin gửi lời cám ơn chân thành sâu sắc tới các thầy cô giáo trong trường đại học Tôn Đức Thắng nói chung và các thầy cô giáo trong khoa Công nghệ Thông tin nói riêng đã tận tình giảng dạy, truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt thời gian qua.

Đặc biệt em xin gửi lời cám ơn đến thầy Mai Văn Mạnh, thầy đã tận tình giúp đỡ, trực tiếp chỉ bảo, hướng dẫn em trong suốt quá trình học. Trong thời gian làm việc với thầy, em không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu nghiêm túc, hiệu quả. Đây thực sự là những điều cần thiết cho em trong quá trình học tập và công tác sau này.

Sau cùng xin gửi lời cám ơn chân thành đến bạn bè đã đóng góp ý kiến và giúp đỡ trong quá trình học tập, nghiên cứu và hoàn thành đồ án.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của Thầy Mai Văn Mạnh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính nhóm tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Phan Tuấn Trọng*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Ngày nay trong quá trình phát triển của moại loại lĩnh vực đều cần có sự góp mặt của công nghệ thông tin, đặc biệt là đối với lĩnh vực bán hàng với số lượng dữ liệu lớn và cần phải thay đổi cách thức quản lý thủ công hiện tại.

Cùng với sự phát triển của phần cứng thì cũng kéo theo sự phát triển vượt bậc của ngành phần mềm hỗ trợ cho con người trong quá trình tính toán cũng như xử lý nghiệp vụ mang lại hiệu quả sử dụng cùng sự tự động hóa cao.

Trong phát triển phần mềm việc đòi hỏi không chỉ là sự chính xác trong việc tính toán mà nó cần phải mang lại tính bảo mật cao cùng với giao diện thân thiện với người dùng dễ sử dụng để mang lại hiệu suất sử dụng cáo tiết kiệm thời gian công sức của người sử dụng mà vẫn đảm bảo được hiệu suất sử dụng cao nhất.

Trước sự phát triển vượt bậc của ngành dịch vụ hiện nay mà đặc biệc là loại hình dịch vụ nhưng cùng với sự phát triển đó thì nó cũng có rất nhiều khó khăn trong nghiệp vụ quản lý cũng như tính toán.

Từ nhu cầu ở trên sẽ cần một phần mềm quản lý quán để giải quyết các vấn đề nghiệp vụ quán trong mà đề tài này sẽ đi vào phân tích và thiết kết .

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc515665988)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc515665989)

[TÓM TẮT iv](#_Toc515665990)

[MỤC LỤC 1](#_Toc515665991)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 5](#_Toc515665992)

[CHƯƠNG 1 – MỞ ĐẦU 11](#_Toc515665993)

[1.1. “Internet of Things” – Vạn vật kết nối 11](#_Toc515665994)

[1.1.1. IoT – Xu hướng cho công nghệ tự động hóa: 12](#_Toc515665995)

[1.1.2. Đặc tính cơ bản của IoT: 13](#_Toc515665996)

[1.1.3. Các giao thức kết nối đang được sử dụng trong IoT 15](#_Toc515665997)

[1.1.4. Các giao thức truyền tải dữ liệu trong IoT 15](#_Toc515665998)

[1.1.4.1. MQTT (Message Queue Telemetry Transport) 15](#_Toc515665999)

[1.1.4.2. CoAP (Constrained Applications Protocol) 16](#_Toc515666000)

[1.1.4.3. DDS (Data Distribution Service) 17](#_Toc515666001)

[1.1.4.4. AMQP (Advanced Message Queue Protocol) 17](#_Toc515666002)

[1.1.4.5. XMPP (Extensible Messaging và Presence Protocol) 18](#_Toc515666003)

[Chương 2 – TỔNG QUAN 19](#_Toc515666004)

[2.1. Mục tiêu và hướng tiếp cận đề tài mạng cảm biến nhiệt độ sử dụng IoT 19](#_Toc515666005)

[2.1.1. Mục tiêu: 19](#_Toc515666006)

[2.1.2. Hướng tiếp cận: 19](#_Toc515666007)

[2.1.2.1. Giới thiệu IBM IoT: 19](#_Toc515666008)

[2.1.2.2. Chuẩn bị: 20](#_Toc515666009)

[2.2. Raspberry Pi: 20](#_Toc515666010)

[2.2.1. Cấu trúc Raspberry Pi 3 Model B 21](#_Toc515666011)

[2.3. Arduino 23](#_Toc515666012)

[2.3.1. Hướng dẫn nạp chương trình đơn giản cho Arduino 23](#_Toc515666013)

[2.3.2. Arduino Uno 29](#_Toc515666014)

[2.3.3. Arduino ESP 8266 MCU 29](#_Toc515666015)

[2.4. Lắp đặt các module vào Raspberry Pi và Arduino 39](#_Toc515666016)

[2.4.1. Đo nhiệt độ và độ ẩm với cảm biến DHT11 39](#_Toc515666017)

[2.4.1.1. Cảm biến DHT11 39](#_Toc515666018)

[2.4.1.2. Lập trình đọc cảm biến 41](#_Toc515666019)

[2.4.2. Đo nhiệt độ với cảm biến LM35(Analog) 43](#_Toc515666020)

[2.4.2.1. Cảm biến LM35 43](#_Toc515666021)

[2.4.2.2. Lập trình đọc cảm biến trên Arduino 45](#_Toc515666022)

[2.4.3. Thu phát tín hiệu với RF loại Zigbee UART CC2530+PA V2 46](#_Toc515666023)

[2.4.3.1. Mạch thu phát RF loại Zigbee UART CC2530+PA V2 46](#_Toc515666024)

[2.4.3.2. Lập trình đọc gửi thông tin Zigbee 49](#_Toc515666025)

[2.4.4. Điều khiển đèn Led đơn màu và còi 49](#_Toc515666026)

[2.4.4.1. Đèn led đơn màu và còi Buzzer 49](#_Toc515666027)

[2.4.4.2. Lập trình điều khiển đèn led/ còi buzzer 50](#_Toc515666028)

[2.4.5. Gửi và nhận giữa Arduino và Raspberry 51](#_Toc515666029)

[2.4.5.1. Gửi và nhận dữ liệu giữa Arduino(ESP8266) và Raspberry (Node-red) thông qua Wifi sử dụng giao thức UDP 51](#_Toc515666030)

[2.4.5.2. Gửi và nhận dữ liệu giữa Arduino Uno và Raspberry (Node-red) thông qua RF sử dụng 2 mạch Zigbee 55](#_Toc515666031)

[Chương 3 - Nghiên cứu thực nghiệm 60](#_Toc515666032)

[3.1. Giới thiệu về Node-red 60](#_Toc515666033)

[3.2. Xử lý giữa Node-red trên Raspberry và IBM Watson 61](#_Toc515666034)

[3.2.1. Đăng ký IBM Waston 61](#_Toc515666035)

[3.2.2. Cài đặt Node-red trên IBM 72](#_Toc515666036)

[3.2.3. Xử lý gửi và nhận dữ liệu giữa Raspberry và IBM Watson 75](#_Toc515666037)

[3.2.3.1. Gửi dữ liệu 75](#_Toc515666038)

[3.2.3.2. Nhận dữ liệu 82](#_Toc515666039)

[3.2.4. Lưu trữ dữ liệu trên Cloudant IBM 83](#_Toc515666040)

[3.3. Các vấn đề gặp phải khi thực hiện dự án 86](#_Toc515666041)

[3.3.1. Truyền nhận thông tin qua Wifi với Arduino ESP 8266 MCU 86](#_Toc515666042)

[3.3.2. Truyền nhận thông tin qua RF loại Zigbee UART CC2530+PA V2 86](#_Toc515666043)

[3.3.3. Lý do sử dụng Raspberry pi 87](#_Toc515666044)

[3.4. Xây dựng hệ thống thực nghiệm 87](#_Toc515666045)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

*f Tần số của dòng điện và điện áp (Hz)*

**CÁC CHỮ VIẾT TẮT**

IoT Internet of Things

Fw Firmware

DB Database

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1 IOT 11](#_Toc515665267)

[Hình 2 Mô hình IoT ứng dụng trong nông nghiệp 12](#_Toc515665268)

[Hình 3 Mô hình IoT ứng dụng trong công nghiệp 13](#_Toc515665269)

[Hình 4 Giao thức MQTT 16](#_Toc515665270)

[Hình 5 Giao thức CoAP và HTTP 17](#_Toc515665271)

[Hình 6 Mô hình IBM IoT 20](#_Toc515665272)

[Hình 7 Cấu hình Pi 3 Model B 21](#_Toc515665273)

[Hình 8 Chân pin của Raspberry Pi 3 Model B 22](#_Toc515665274)

[Hình 9 Kết nối Arduino UNO R3 vào máy tính 23](#_Toc515665275)

[Hình 10 mở Device Manager 24](#_Toc515665276)

[Hình 11 cửa sổ Device Manager 24](#_Toc515665277)

[Hình 12 Cổng kết nối ở đây là COM3. 25](#_Toc515665278)

[Hình 13 Chọn loại Arduino muốn nạp 25](#_Toc515665279)

[Hình 14 Chọn cổng Arduino đang kết nối 26](#_Toc515665280)

[Hình 15 Xác nhận cổng COM của Arduino IDE ở góc dưới cùng bên phải cửa sổ làm việc 26](#_Toc515665281)

[Hình 16 chọn AVR ISP 27](#_Toc515665282)

[Hình 17 Nạp một chương trình mẫu 27](#_Toc515665283)

[Hình 18 Cửa sổ mới chứa mã nguồn Blink 28](#_Toc515665284)

[Hình 19 IDE xác nhận đã lập trình thành công 28](#_Toc515665285)

[Hình 20 Sơ đồ chân của Arduino Uno R3 29](#_Toc515665286)

[Hình 21 Arduino ESP 8266 MCU 29](#_Toc515665287)

[Hình 22 Sơ đồ chân của Arduino ESP 8266 MCU 30](#_Toc515665288)

[Hình 23 Tạo custom fw cho NodeMCU ESP8266 - Điền địa chỉ email nhận file fw 31](#_Toc515665289)

[Hình 24 Tạo custom fw cho NodeMCU ESP8266 - Chọn các module cần thiết 31](#_Toc515665290)

[Hình 25 Tạo custom fw cho NodeMCU ESP8266 - Tạo thành công fw 32](#_Toc515665291)

[Hình 26 Tạo custom fw cho NodeMCU ESP8266 - Email nhận file custom fw 32](#_Toc515665292)

[Hình 27 Cài đặt python 33](#_Toc515665293)

[Hình 28 Bắt đầu nạp fw cho ESP8266 34](#_Toc515665294)

[Hình 29 Nạp fw cho ESP8266 thành công 34](#_Toc515665295)

[Hình 30 Cài đặt Board Manager cho Arduino IDE 35](#_Toc515665296)

[Hình 31 Tiến hành cài đặt Board Manager cho Esp8266 36](#_Toc515665297)

[Hình 32 Chọn phiên bản Board Manager cho Esp8266 36](#_Toc515665298)

[Hình 33 chọn Board NodeMCU 1.0 37](#_Toc515665299)

[Hình 34 chọn COM để tiến hành nạp code 38](#_Toc515665300)

[Hình 35 0D0HT11 (trái) và DHT22 (phải) 39](#_Toc515665301)

[Hình 36 Cảm biến chưa hàn mạch và cảm biến đã hàn mạch ra chân 39](#_Toc515665302)

[Hình 37 Mắc dây với cảm biến DHT11/22, nguồn VCC của Arduino/Pi sẻ kết nối với một điện trở 10k 40](#_Toc515665303)

[Hình 38 Mắc dây với cảm biến DHT11/22 đã được hàn mạch ra chân 41](#_Toc515665304)

[Hình 39 Nội dung file python 41](#_Toc515665305)

[Hình 40 Kết quả đọc cảm biến DHT trên raspberry Py 42](#_Toc515665306)

[Hình 41 Thêm thư viện DHT cho Arduino IDE 42](#_Toc515665307)

[Hình 42 Chọn đường dẫn file thư viện DHT 43](#_Toc515665308)

[Hình 43 Thông số kỹ thuật cảm biến LM35 44](#_Toc515665309)

[Hình 44 Mắc dây với cảm biến LM35 44](#_Toc515665310)

[Hình 45 Kết quả đọc cảm biến LM35 45](#_Toc515665311)

[Hình 46 Chọn baudrate cho Zigbee UART CC2530+PA V2 47](#_Toc515665312)

[Hình 47 Chọn channel cho Zigbee UART CC2530+PA V2 48](#_Toc515665313)

[Hình 48 Chân của Zigbee UART CC2530+PA V2 49](#_Toc515665314)

[Hình 49 Chân cắm của đèn Led và còi 50](#_Toc515665315)

[Hình 50 Lấy code mẫu nhấp nháy đèn led/còi kêu 51](#_Toc515665316)

[Hình 51 Flow thử bật tắt và còi 51](#_Toc515665317)

[Hình 52 Cấu hình node UDP in 52](#_Toc515665318)

[Hình 53 Node Function toString để ép kiểu dữ liệu buffer sang String 53](#_Toc515665319)

[Hình 54 Node Function toFloatToCompare để chuyển kiểu dữ liệu cho so sánh 53](#_Toc515665320)

[Hình 55 Cấu hình node Switch 53](#_Toc515665321)

[Hình 56 Cấu hình node tạo lệnh 54](#_Toc515665322)

[Hình 57 Cấu hình node UDP out 54](#_Toc515665323)

[Hình 58 Flow gửi và nhận dữ liệu bằng UDP trên node-red 55](#_Toc515665324)

[Hình 59 Cấu hình node Serial in 56](#_Toc515665325)

[Hình 60 Kiểm tra cổng port cắm Zigbee trên Raspberry 56](#_Toc515665326)

[Hình 61 Cấu hình node getTempToCompare 57](#_Toc515665327)

[Hình 62 Cấu hình node Switch 57](#_Toc515665328)

[Hình 63 Cấu hình node tạo lệnh - Serial 58](#_Toc515665329)

[Hình 64 Cấu hình node Serial out 58](#_Toc515665330)

[Hình 65 Flow gửi và nhận dữ liệu bằng Serial qua RF trên node-red 59](#_Toc515665331)

[Hình 66 Tương tác giữa các node khác nhau 60](#_Toc515665332)

[Hình 67 Raspberry tương tác với các cảm biến và thiết bị 60](#_Toc515665333)

[Hình 68 Node-red chạy trên Raspberry hay OrangePi 61](#_Toc515665334)

[Hình 69 IBM Watson nằm trung gian giữa Client và các cảm biến, thiết bị 61](#_Toc515665335)

[Hình 70 Nhấn vào Sign Up 62](#_Toc515665336)

[Hình 71 Form đăng ký tài khoản IBM 62](#_Toc515665337)

[Hình 72 Yêu cầu xác thực email 63](#_Toc515665338)

[Hình 73 Truy cập vào địa chỉ email xác thực 63](#_Toc515665339)

[Hình 74 Mở email được IBM gửi 63](#_Toc515665340)

[Hình 75 Đăng ký thành công 64](#_Toc515665341)

[Hình 76 Form Login tài khoản IBM-Nhập địa chỉ email 64](#_Toc515665342)

[Hình 77 Form Login tài khoản IBM-Nhập mật khẩu 65](#_Toc515665343)

[Hình 78 Giới thiệu các điều khoản IBM 65](#_Toc515665344)

[Hình 79 Giao diện Dashboard khi mới tạo tài khoản IBM 66](#_Toc515665345)

[Hình 80 Đăng ký dịch vụ Internet of Things Platform 66](#_Toc515665346)

[Hình 81 Điền tên App trên IBM 67](#_Toc515665347)

[Hình 82 IBM kích hoạt dịch vụ 67](#_Toc515665348)

[Hình 83 Chọn Cloud Foundry Applications 67](#_Toc515665349)

[Hình 84 Chọn IoT Platform 68](#_Toc515665350)

[Hình 85 Giao diện Watson IoT Dashboard 68](#_Toc515665351)

[Hình 86 Sơ đồ kết nối tổng quan giữ các thiết bị và IBM Watson 69](#_Toc515665352)

[Hình 87 Chọn tab Devices 69](#_Toc515665353)

[Hình 88 Thêm thiết bị trên IBM Watson 69](#_Toc515665354)

[Hình 89 Thêm loại thiết bị 70](#_Toc515665355)

[Hình 90 Tạo thiết bị trên Watson IoT 70](#_Toc515665356)

[Hình 91 Tạo mật khẩu liên kết thiết bị 71](#_Toc515665357)

[Hình 92 Hoàn tất khởi tạo thiết bị trên Watson IoT 71](#_Toc515665358)

[Hình 93 Thông tin thiết bị đã đăng ký với Watson 71](#_Toc515665359)

[Hình 94 Giao diện Dashboard trên IBM Cloud 72](#_Toc515665360)

[Hình 95 Trang quản lý dịch vụ Node-red trên IBM 72](#_Toc515665361)

[Hình 96 Truy cập Node-red 73](#_Toc515665362)

[Hình 97 Tạo tài khoản quản lý Node-red trên IBM 73](#_Toc515665363)

[Hình 98 Hoàn tất cài đặt Node-red trên IBM 74](#_Toc515665364)

[Hình 99 Giao diện trang Node-red trên IBM 74](#_Toc515665365)

[Hình 100 Giao diện Flow trên Node-red IBM 74](#_Toc515665366)

[Hình 101 Đăng nhập quản lý trên Node-red IBM 75](#_Toc515665367)

[Hình 102 Cập nhật node-red trên Raspberry 76](#_Toc515665368)

[Hình 103 Truy cập vào Manage palette - Quản lý node 77](#_Toc515665369)

[Hình 104 Cài đặt node kết nối IBM cho node-red 77](#_Toc515665370)

[Hình 105 Chọn APPS để tạo khóa API 78](#_Toc515665371)

[Hình 106 Chọn Generate API Key 78](#_Toc515665372)

[Hình 107 Mô tả API Key 78](#_Toc515665373)

[Hình 108 Chọn Role cho API key 79](#_Toc515665374)

[Hình 109 Thông tin API key 79](#_Toc515665375)

[Hình 110 API key ta vừa tạo 79](#_Toc515665376)

[Hình 111 Cấu hình node ibmiot out 80](#_Toc515665377)

[Hình 112 Thiết lập thông số API key trên node-red 81](#_Toc515665378)

[Hình 113 node toJSON 81](#_Toc515665379)

[Hình 114 node-red Pi gửi dữ liệu lên IBM Watson 82](#_Toc515665380)

[Hình 115 Kiểm tra dữ liệu gửi lên IBM Watson 82](#_Toc515665381)

[Hình 116 Cấu hình node ibmiot in trên Node-red on IBM 83](#_Toc515665382)

[Hình 117 Đăng nhập dịch vụ Cloudant trên IBM 83](#_Toc515665383)

[Hình 118 Giao diện Cloudant 84](#_Toc515665384)

[Hình 119 Tạo DB trên Cloudant 84](#_Toc515665385)

[Hình 120 Cấu hình node Cloudant out 85](#_Toc515665386)

[Hình 121 Nối dây để hoàn tất lưu vào Cloudant 85](#_Toc515665387)

[Hình 122 Kết quả lưu trữ trên Cloudant 86](#_Toc515665388)

[Hình 123 Giao diện hiển thị dữ liệu nhận đươc của IBM cung cấp 89](#_Toc515665389)

[Hình 124 Màn hình điều khiển cũng như theo dõi của Node-red trên Raspberry Pi 89](#_Toc515665390)

[Hình 125 Màn hình điều khiển cũng như theo dõi của Node-red trên IBM 90](#_Toc515665391)

**DANH MỤC BẢNG**

[Bảng 1 Các giao thức kết nối 15](#_Toc515665392)

[Bảng 2 Thông số DHT11 và DHT22 40](#_Toc515665393)

[Bảng 3 Mắc dây với cảm biến DHT11/DHT22 chưa được hàn mạch 40](#_Toc515665394)

[Bảng 4 Mắc dây với cảm biến DHT11/DHT22 đã được hàn mạch ra chân 41](#_Toc515665395)

[Bảng 5 Mắc dây với cảm biến LM35 45](#_Toc515665396)

[Bảng 6 Cắm chân của Zigbee UART CC2530+PA V2 49](#_Toc515665397)

[Bảng 7 Cắm của đèn Led và còi 50](#_Toc515665398)

[Bảng 8 Danh sách các node để gửi và nhận dữ liệu bằng UDP 52](#_Toc515665399)

[Bảng 9 Danh sách các node để gửi và nhận dữ liệu bằng Serial sử dụng RF 55](#_Toc515665400)

[Bảng 10 Danh sách các node sử dụng kết nối với IBM Watson trên node-red 80](#_Toc515665401)

CHƯƠNG 1 – KHẢO SÁT HỆ THỐNG

* 1. GIỚI THIỆU VỀ THỰC TRẠNG

Đây là đề tài về thiết kế phần mềm quản lý quán ăn.

Hiện nay thì cùng với quá trình phát triển của công nghệ thông tin 4.0 thì hầu như bất cứ doanh nghiệp nào cũng sẽ cần đến một phần mềm quản lý cho doanh nghiệp mình để giảm thiểu sự sai sót trong quá trình tính toán cũng như đơn giản hóa các nghiệp vụ quản lý trước đây được thực hiện một cách thô sơ qua cách lưu trữ bằng cách viết và lưu giữ trên giấy tờ.

Việc quản lý quán ăn trước đây nếu không có sự hỗ trợ của tin học thì các quy trình nghiệp vụ như quản lý nhân viên, quản lý hóa đơn …. đều được thực hiện một cách thủ công thông qua sổ sách giấy tờ vì vậy nó mang lại rất nhiều khó khăn cho việc kiểm tra và rà soát thông tin cũng như thống kê. Do các thông tin được lưu trữ thủ công qua sổ sách nên cũng rất dễ xảy ra sai sót trong quá trình tính toán, điều này dẫn đến sự thất thoát tốn kém do vậy việc tin học hóa các quy trình nghiệp vụ mà cụ thể ở đây là thông qua phần mềm quản lý quán ăn sẽ cải thiện được các nhược điểm ở trên mang lại hiệu suất sử dụng cao cùng với độ chính xác cao tính bảo mật cũng được đảm bảo.

Tại những quán ăn hiện nay với lượng khách hàng ngày càng tăng thì đòi hỏi độ chính xác cao trong quá trình tính toán và quản lý đặc biệt là trong nghiệp vụ kế toán và quản lý hàng hóa.

Các nghiệp vụ được thực hiện thủ công bộc lộ nhiều hạn chế :

* Tra cứu thông tin không được chính xác và mất nhiều thời gian trong việc tìm kiếm.
* Lưu trữ thông tin về nhập hàng hóa đơn và các thông tin về nhập xuất hàng hóa bằng sổ sách hết sức cồng kềnh và dễ bị lạc mất.
* Vì việc tìm kiếm khó khăn nên việc cập nhật thông tin cũng vậy.
* Ngoài ra còn gây khó khăn cho quá trình tổng kết thống kê báo cáo doanh thu.

Những hạn chế trên có thể được khắc phục với một hệ thống thông tin hay phần mềm quản lý :

* Việc quản lý nhân viên trở nên dễ thực hiện hơn
* In hóa đơn lưu trữ hóa đơn dễ dàng và nhanh chóng.
* Thông tin lưu trữ được lâu dài và không bị mất mát.
  1. MỤC TIÊU VÀ PHẠM VI ĐỀ TÀI
     1. Mục tiêu

Sau khi hoàn thành các bước phân tích, xử lý các thông tin… thì kết quả cuối cùng của đề tài là ta sẽ xây dựng ra được một chương trình thực tế để quản lý hệ thống hoạt động của một quán ăn cho doanh nghiệp vừa và nhỏ.

* + 1. Phạm vi đề tài

Quản lí hệ thống:

* Quản lí thông tin nhân viên.
* Quản lý sản phẩm.
* Quản lý bàn : danh sách bàn, tình trạng bàn.
* Quản lý hoá đơn.

Chương 2 – PHÂN TÍCH HỆ THỐNG

* 1. Các yêu cầu và chức năng
* Quản lý nhân viên: Chức năng quản lý nhân viên sẽ được cấp quyền cho tài khoản có quyền quản lý, bao gồm : xem, chỉnh sửa thông tin nhân viên, thêm nhân viên, cấp tài khoản nhân viên, thay đổi mật khẩu đăng nhập và tạo mới tài khoản.
* Quản lý sản phẩm: Bao gồm thêm sản phẩm, chỉnh sửa thông tin sản phẩm(tên, thể loại, giá).
* Quản lý bàn và Quản lý đặt món:
  + Quản lý bàn: sẽ cho phép nhân viên kiểm tra nhanh tình trạng bàn đã phục vụ khách.
  + Khi nhận được danh sách yêu cầu từ nhân viên phục vụ nhân viên thu ngân sẽ nhấn vào bàn( dựa trên thông tin từ nhân viên phục vụ sẽ có mã bàn đang phục vụ và danh sách các món để tiến hành đặt món). Khi nhấn vào bàn chưa phục vụ sẽ hiện ra giao diện chọn món. Khi nhấn vào bàn đang phục vụ( khi thanh toán hoặc khi khách gọi thêm món) sẽ hiện ra giao diện thanh toán hoặc thêm món.
* Quản lý hóa đơn: đơn cho phép người sử dụng xem lại danh sách hóa đơn nhưng không cho phép xóa hóa đơn.
  1. Các bảng và thuộc tính:

Dựa trên mô tả và khảo sát bài toán em thiết kế được các bảng như sau:

**NhanVien**(MaNhanVien, TenNhanVien, NgaySinh, DiaChi, ChucVu, SoDienThoai).

**TaiKhoan**(TaiKhoan, MatKhau, MaNhanVien, MatKhau2).

**Ban**(MaBan, TrangThai).

**HoaDon**(MaHoaDon, NgayLap, TongTien, MaNhanVien, MaBan, TrangThai).

**ChiTietHoaDon**(MaChiTietHoaDon, MaHoaDon, MaSanPham, SoLuong, ThanhTien).

**SanPham**(MaSanPham, TenSanPham, LoaiSanPham , Gia).

Dựa trên các bảng đã thiết kế em tiến hành xác định thuộc tính của các trường của bảng, khóa chính và tiến hành vẽ sơ đồ Class Diagram và ERD.

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô tả** |
| MaNhanVien | Nvarchar | Mã nhân viên – Khóa chính |
| TenNhanVien | Nvarchar | Tên nhân viên |
| NgaySinh | Nvarchar | Ngày sinh |
| ChucVu | Nvarchar | Chức vụ |
| DiaChi | Nvarchar | Địa chỉ của nhân viên |
| SoDienThoai | Nvarchar | Số điện thoại của nhân viên |

Bảng 1 Bảng Nhân Viên

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô tả** |
| TaiKhoan | Nvarchar | Tài khoản đăng nhập – Khóa chính |
| MatKhau | Nvarchar | Mật khẩu đăng nhập |
| MaNhanVien | Nvarchar | Mã nhân viên(tài khoản thuộc nhân viên) |
| MatKhau2 | Nvarchar | Mật khẩu 2 |

Bảng 2 Bảng Tài khoản

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô tả** |
| MaBan | Nvarchar | Mã bàn – Khóa chính |
| TrangThai | Nvarchar | Trạng thái( đang phục vụ , trống) |

Bảng 3 Bảng bàn

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô tả** |
| MaHoaDon | Nvarchar | Mã hóa đơn – Khóa chính |
| NgayLap | DateTime | Ngày giờ lập hóa đơn |
| TongTien | Double | Tổng tiền hóa đơn |
| MaNhanVien | Nvarchar | Mã nhân viên tạo hóa đơn |
| TrangThai | Nvarchar | Trạng thái hoá đơn |
| MaBan | Nvarchar | Mã bàn |

Bảng 4 Bảng Hoá đơn

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô tả** |
| MaChiTietHoaDon | Nvarchar | Mã chi tiết hoá đơn – Khoá chính |
| MaHoaDon | Nvarchar | Mã hóa đơn -Khóa chính |
| MaSanPham | Nvarchar | Mã sản phẩm |
| SoLuong | Int | Số lượng sản phẩm |
| ThanhTien | Float | Thành tiền của sản phẩm |

Bảng 5 Bảng Chi tiết hoá đơn

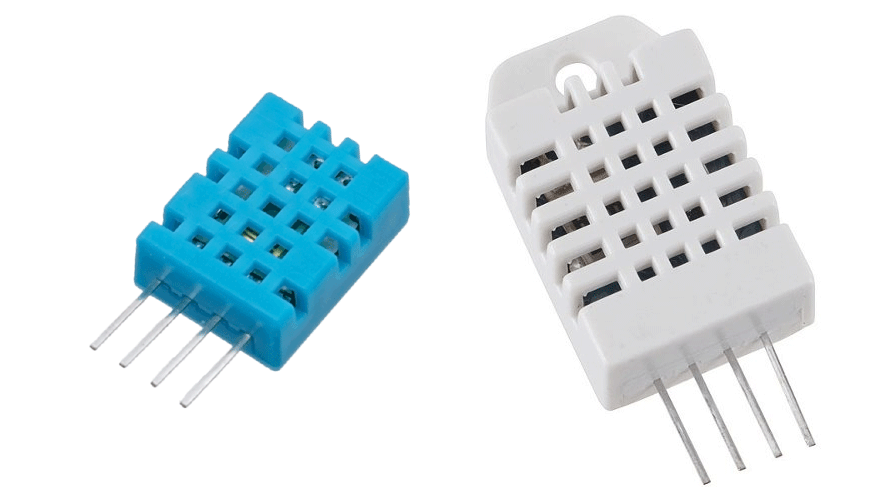
|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô tả** |
| MaSanPham | Nvarchar | Mã sản phẩm – Khóa chính |
| TenSanPham | Nvarchar | Tên sản phẩm |
| Gia | Float | Giá bán |
| LoaiSanPham | Nvarchar | Loại sản phẩm(nước uống,) |

Bảng 6 Bảng Sản phẩm

* 1. Usecase diagram

Từ các yêu cầu chức năng trên cùng với các bảng và thuộc tính ta có mô hình Usecase Diagram sau:

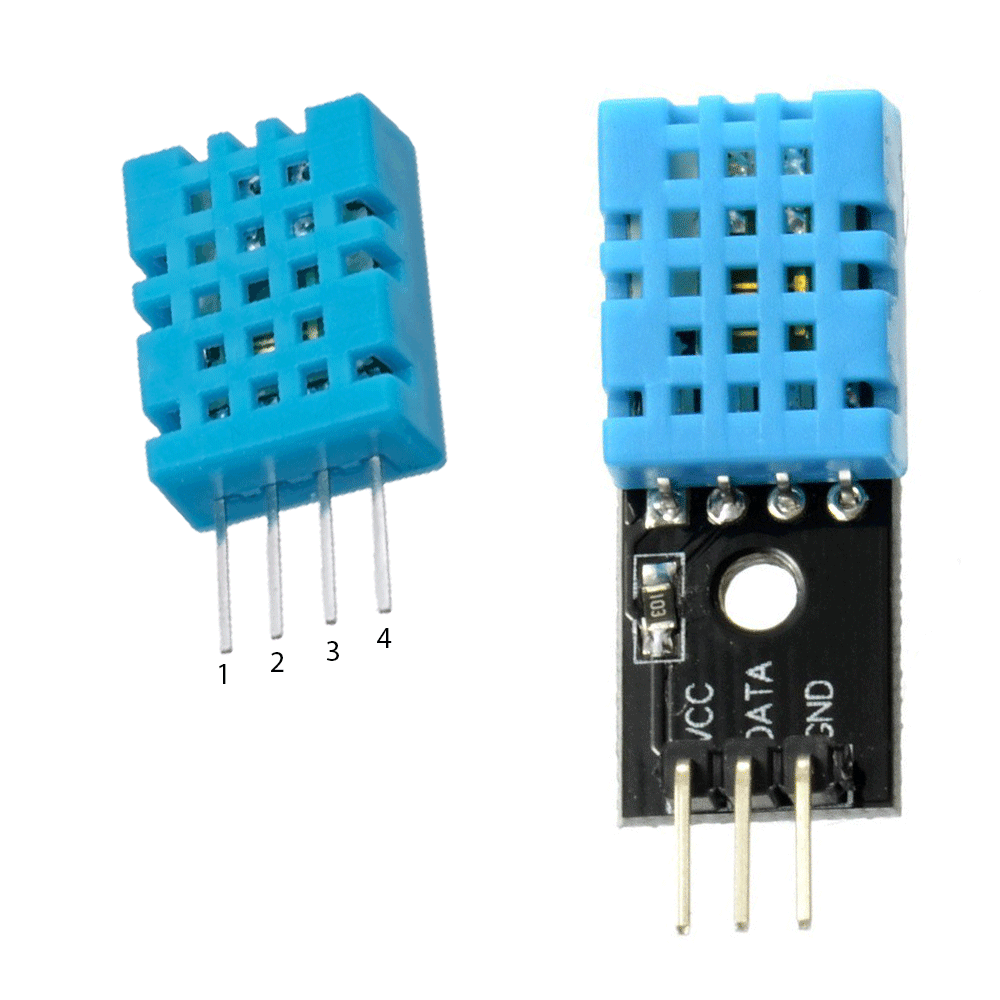
* 1. Lắp đặt các module vào Raspberry Pi và Arduino
     1. Đo nhiệt độ và độ ẩm với cảm biến DHT11
        1. Cảm biến DHT11



Hình 35 0D0HT11 (trái) và DHT22 (phải)

DHT11 và DHT22 là cảm biến nhiệt độ, độ ẩm rất thông dụng hiện nay vì **giá thành rẻ** và dễ dàng lấy được 2 dữ liệu riêng biệt của cảm biến thông qua cổng giao tiếp 1-wire (giao tiếp digital 1-wire truyền dữ liệu duy nhất). Cảm biến được tích hợp bộ tiền xử lý tín hiệu giúp dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ tính toán nào nhưng cần có bộ thư viện.

Trên hiện thị trường hiện nay, cảm biến này cũng được bán dưới hai dạng: chưa được hàn mạch (chỉ có cảm biến với 4 chân pin, cảm biến bên trái hình 13) và đã hàn mạch ra chân sẵn (cảm biến được hàn chung với 2 điện trở và chỉ còn 3 chân pin ra)



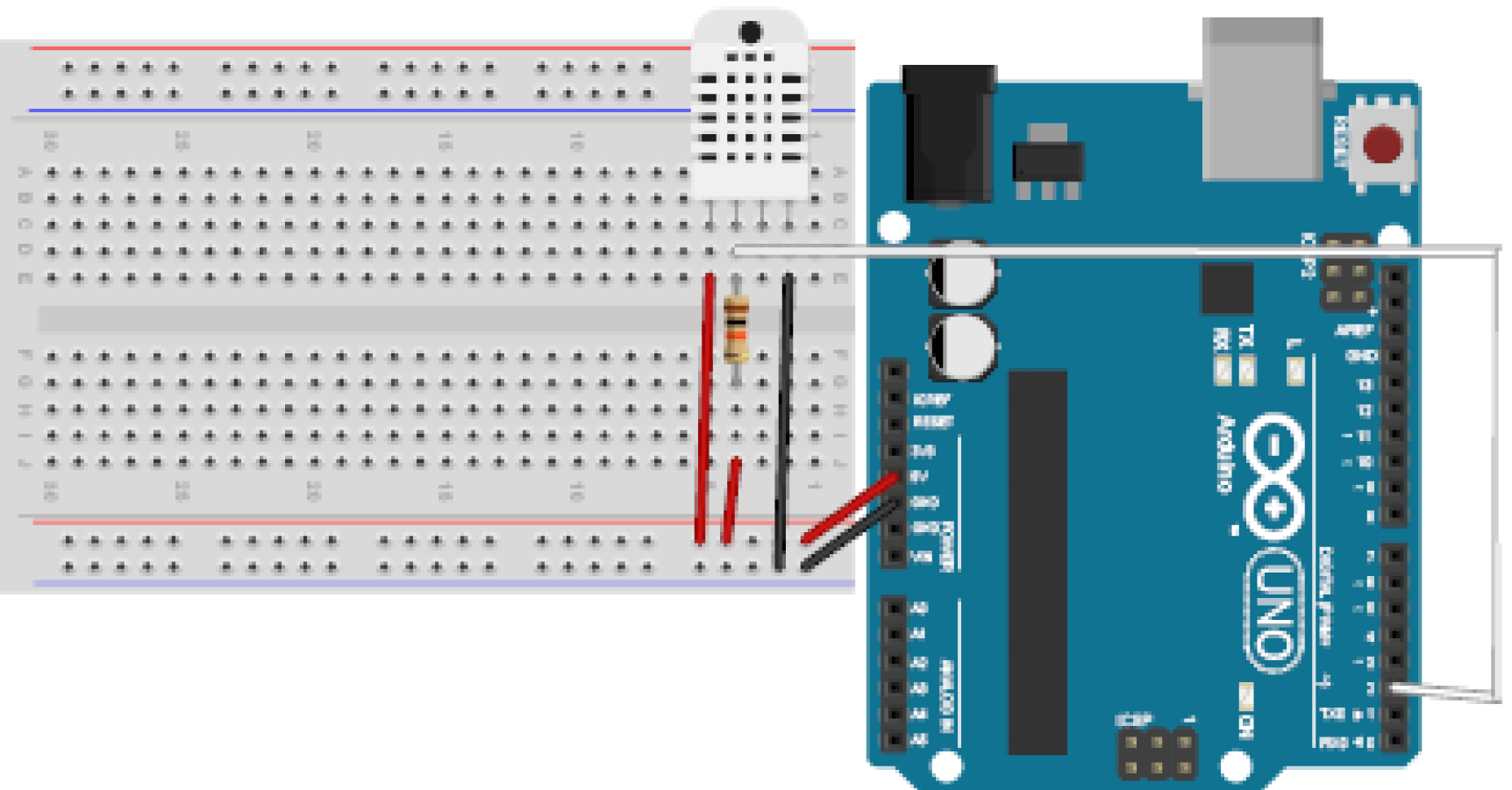
Hình 36 Cảm biến chưa hàn mạch và cảm biến đã hàn mạch ra chân

DHT11 và DHT22 có cách sử dụng giống nhau gần như là hoàn toàn chỉ khác nhau về giá thành và lý do giá thành khác nhau thì thông qua các thông số như bảng dưới đây bạn sẽ biết:

|  |  |  |
| --- | --- | --- |
|  | DHT11 | DHT22 |
| Điện áp hoạt động | 3 ~ 5 VDC | 3 ~ 5 VDC |
| Dòng sử dụng | 2.5mA max (khi truyền dữ liệu) | 2.5mA max (khi truyền dữ liệu) |
| Ngưỡng độ ẩm | 20% - 90% RH | 100%RH |
| Sai số độ ẩm | ±5%RH | 2 ~ 5% |
| Ngưỡng nhiệt độ | 0°C ~ 50°C | -40 to 80°C |
| Sai số nhiệt độ | ±2°C | ±0.5°C |
| Tần số lấy mẫu tối đa | 1 Hz | 0.5Hz |
| Kích thước | 15mm x 12mm x 5.5mm | 27mm x 59mm x 13.5mm |

Bảng 2 Thông số DHT11 và DHT22

Quy ước mắc dây với cảm biến DHT11/DHT22 chưa được hàn mạch:

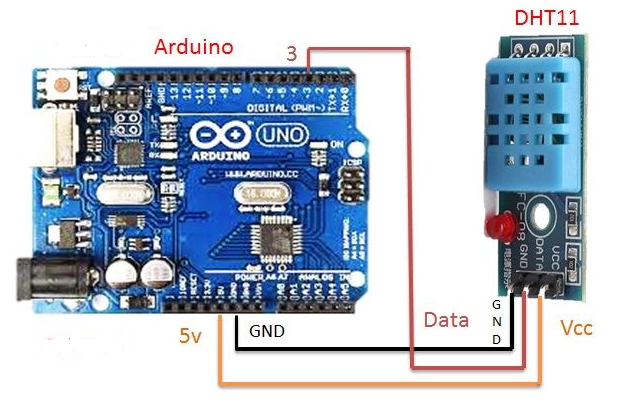


Hình 37 Mắc dây với cảm biến DHT11/22, nguồn VCC của Arduino/Pi sẻ kết nối với một điện trở 10k

|  |  |  |
| --- | --- | --- |
| STT | Chân DHT11/22 | Chân Raspberry Pi/Arduino |
| 1 | 1 (VCC) | 3.3V/5V |
| 2 | 2 (Data) | GPIO Digital |
| 3 | 3 (NC) |  |
| 4 | 4 (GND) | GND |

Bảng 3 Mắc dây với cảm biến DHT11/DHT22 chưa được hàn mạch

Quy ước mắc dây với cảm biến DHT11/DHT22 đã được hàn mạch ra chân:



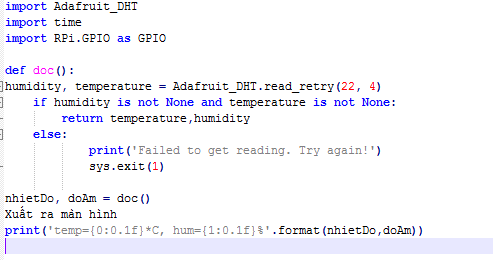
Hình 38 Mắc dây với cảm biến DHT11/22 đã được hàn mạch ra chân

|  |  |  |
| --- | --- | --- |
| STT | Chân DHT11/22 | Chân Raspberry Pi/Arduino |
| 1 | VCC/+ | 5V |
| 2 | Data/Signal | GPIO Digital |
| 3 | GND/- | GND |

Bảng 4 Mắc dây với cảm biến DHT11/DHT22 đã được hàn mạch ra chân

* + - 1. Lập trình đọc cảm biến
* Đọc cảm biến DHT trên raspberry pi:

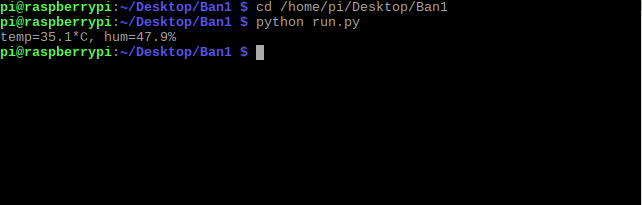
Tạo file có đuôi .py. Các thư viện sử dụng: Adafruit\_DHT, time, Rpi.GPIO as GPIO, có thể pdb để debug nếu cần. Khai báo hàm:



Hình 39 Nội dung file python

Chạy file code:

Mở terminal ở thư mục chứa file python .py ta đã tạo ở trên. Sau đó dùng câu lệnh python tenfile.py để biên dịch và chạy để hiển thị kết quả.

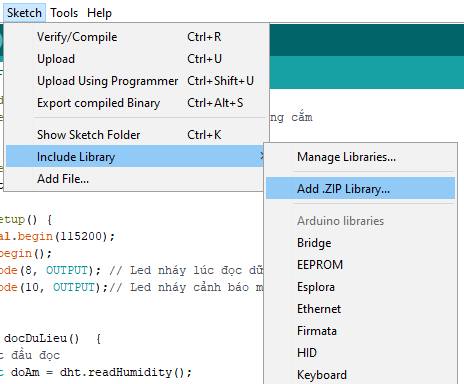


Hình 40 Kết quả đọc cảm biến DHT trên raspberry Py

* Đọc cảm biến DHT trên arduino

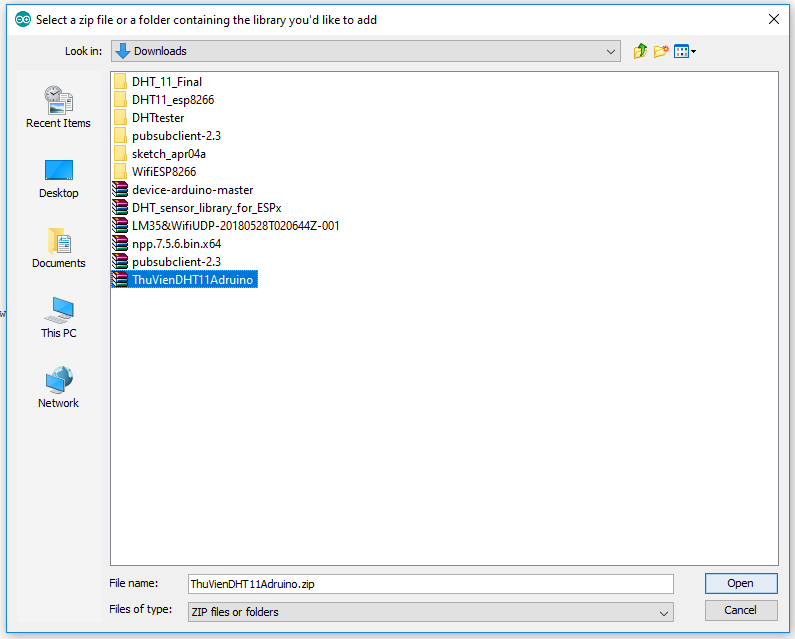
Đối với DHT11 ta cần thêm thư viện DHT để có thể đọc được dữ liệu từ cảm biến, bộ thư viện và code mẫu đã được đính kèm theo CD đồ án, dưới đây là cách thêm thư viện:

Đầu tiên ta tải file thư viện về và mở Arduino IDE lên, sau đó ta nhấn vào Sketch -> Include Library -> Add Zip Library như hình dưới:



Hình 41 Thêm thư viện DHT cho Arduino IDE

Sau đó ta chọn tới file thư viện (.zip) ta đã tải ở trên và chọn open:



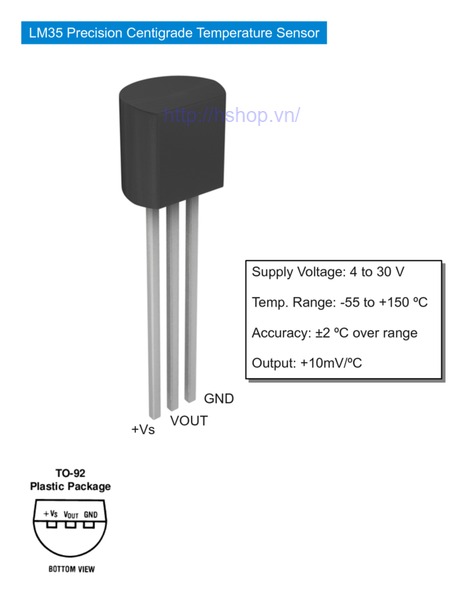
Hình 42 Chọn đường dẫn file thư viện DHT

Như vậy ta đã thêm thư viện DHT xong, bây giờ ta có thể gọi thư viện DHT và tiến hành cài đặt code đọc dữ liệu từ DHT trên Arduino IDE.

* + 1. Đo nhiệt độ với cảm biến LM35(Analog)
       1. Cảm biến LM35

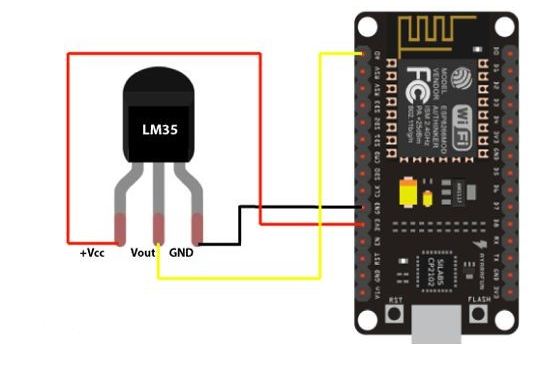
Cảm biến LM35 là bộ cảm biến nhiệt mạch tích hợp chính xác cao mà điện áp đầu ra của nó tỷ lệ tuyến tính với nhiệt độ theo thang độ Celsius. Chúng cũng không yêu cầu cân chỉnh ngoài vì vốn chúng đã được cân chỉnh. Cảm biến LM35 hoạt động bằng cách cho ra một giá trị hiệu điện thế nhất định tại chân Vout (chân giữa) ứng với mỗi mức nhiệt độ.

Cảm biến LM35 có ưu thế hơn so với DHT11/22 là tốc độ đo nhanh hơn, dãi nhiệt độ đo dài hơn, rẻ hơn so với DHT. Nhưng nó lại có nhược điểm là chỉ dùng cổng analog để đọc(cổng này thường ít trên các mạch), sai số thấp nhất bằng với DHT11và chỉ đo được nhiệt dộ không thể đo độ ẩm



Hình 43 Thông số kỹ thuật cảm biến LM35

Quy ước mắc dây với cảm biến LM35:



Hình 44 Mắc dây với cảm biến LM35

|  |  |  |
| --- | --- | --- |
| STT | Chân LM35 | Arduino |
| 1 | Vs+ | 5V |
| 2 | VOUT | GPIO Analog |
| 3 | GND | GND |

Bảng 5 Mắc dây với cảm biến LM35

* + - 1. Lập trình đọc cảm biến trên Arduino

Thông thường đối với LM35 nhiệt độ đo được sẽ tương ứng với hiệu điện thế mà cảm biến đọc được và theo nhà sản xuất thì 1°C tương ứng với 10mV. Với ESP82266 hiệu điện thế các chân cung cấp trên lý thuyết là 3.3V nên ta sẽ nhân hiệu điện thế đo được cho 3.3 và nhân cho 100 (do 1V = 1000mV) và cần chia cho 1024.

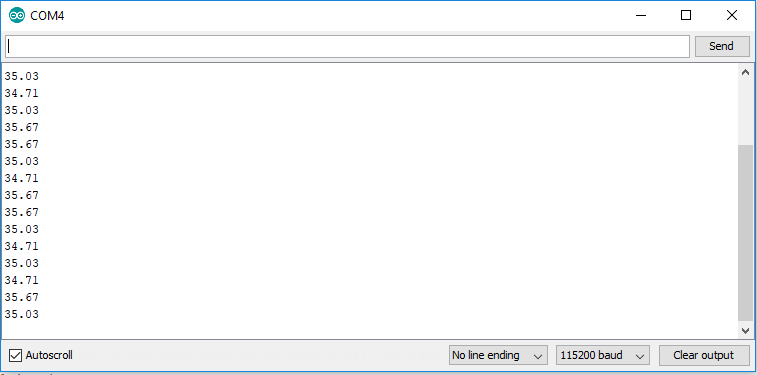
Lý do phải chia cho 1024 vì chân analog của ESP8266 ra tín hiệu là 10 bit và 2^10 = 1024 nên ta cần phải chia cho 1024.

Như vậy ta sẽ có công thức cụ thể cho nhiệt độ ra của LM35 là:

float temp = analogRead(sensorPin) \* 3.3 / 1024.0 \* 100.0 – 3(Do cảm biến bị chênh lệch (so với DHT11) nên cần trừ hao 3 độ).

Với hàm analogRead() là hàm được hỗ trợ sẵn để đọc giá trị từ chân pin (biến sensorPin ở đây là giá trị chân A0-chân analog trên ESP8266). File code cụ thể sẽ được đính kèm trong đĩa đồ án.

Sau khi nạp code ta nhấn tổ hợp phím Ctrl+Shift+M để xem kết quả như hình dưới:



Hình 45 Kết quả đọc cảm biến LM35

* + 1. Thu phát tín hiệu với RF loại Zigbee UART CC2530+PA V2
       1. Mạch thu phát RF loại Zigbee UART CC2530+PA V2

Mạch thu phát RF Zigbee UART CC2530+PA V2 sử dụng IC CC2530 từ TI + với tầng khuếch đại công suất PA CC2591 cho khoảng các truyền nhận rất xa, điều khiện lý tưởng có thể đạt tới 1000m, mạch được lập trình sẵn firmware để có thể dễ dàng sử dụng như một module truyền nhận dữ liệu không dây chuẩn Zigbee với giao tiếp UART rất dễ kết nối với vi điều khiển hoặc máy tính (thông qua cáp chuyển USB-UART) với chỉ một vài bước config bằng nút nhấn.

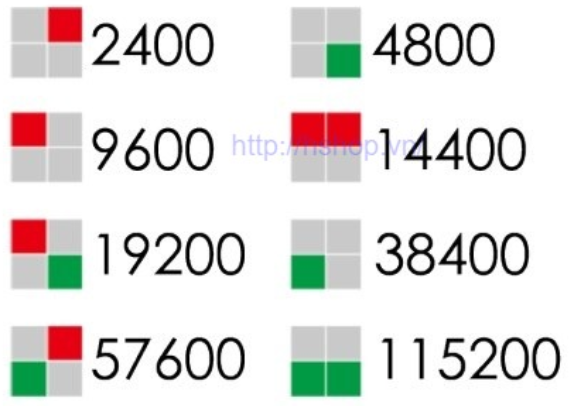
Mạch thu phát RF Zigbee UART CC2530+PA V2 có khoảng cách truyền nhận xa, chuẩn truyền sóng Zigbee 2.4Ghz chuẩn công nghiệp rất ổn định và có khả năng cấu hình tạo thành mạng truyền nhận không dây với nhiều nút, điểm mạng khác khau qua giao thức Zigbee.

**Thông số kỹ thuật:**

* Mạch thu phát RF Zigbee UART CC2530+PA V2.
* IC chính RF Zigbee SoC CC2530 từ TI, khuếch đại công suất PA CC2591.
* Điện áp sử dụng: 2.8 - 3.5VDC
* Dòng tiêu thụ: < 30mA
* Chuẩn truyền sóng Zigbee 2.4Ghz.
* Tốc độ truyền sóng tối đa 3300bps.
* Công suất truyền: 20dbm
* Khoảng cách truyền lý tưởng: 1000m.
* Giao thức kết nối UART TTL 3.3V, Baudrate tối đa 115200.
* Kích thước: 16x34mm.

**Cách sử dụng:**

- **Bước 1:** Không cấp nguồn, nhấn giữ phím key, sau đó cấp nguồn, bốn led trên module sẽ chớp báo hiệu vào chế độ cấu hình, thả nút nhấn ra chúng ta sẽ vào chế độ cấu hình baudrate cho kết nối UART đầu tiên, nhấn phím key tuần tự để chọn baudrate phù hợp (nhìn vào trạng thái 4 led hiển thị) theo bảng sau:



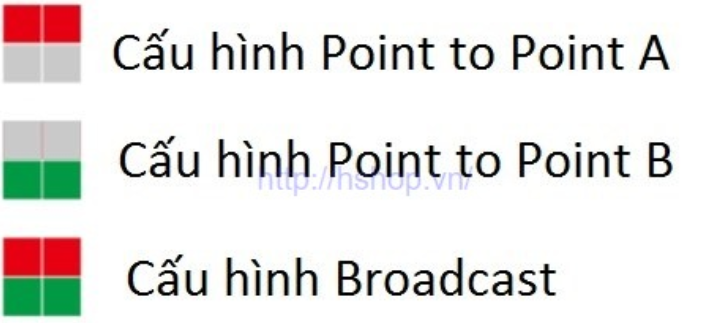
Hình 46 Chọn baudrate cho Zigbee UART CC2530+PA V2

- **Bước 2:** Sau khi chọn Baurate phù hợp cho kết nối UART, nhấn tì phím key để lưu thiết đặt sẽ thấy 4 led cùng chớp báo hiệu chuyển qua chế độ cấu hình tiếp theo: kênh - Channel, có 16 kênh có thể lựa chọn theo trạng thái 4 led hiển thị, lưu ý các module muốn truyền nhận được với nhau phải có cùng channel (không nhất thiết phải cùng baudrate vì baudrate chỉ dùng để giao tiếp với Vi điều khiển).

- **Bước 3:** Sau khi chọn channel, nhấn tì phím key để lưu sẽ thấy 4 led cùng chớp báo hiệu chuyển qua chế độ cấu hình tiếp theo: kiểu truyền nhận, có 2 kiểu truyền là Point to Point và Broadcast.

**Point to Point:** chỉ duy nhất 2 module kết nối với nhau trong 1 Channel, cấu hình 1 Module là A, 1 Module là B trong 1 Channel là đã có thể truyền nhận với nhau.

**Boardcast:** Nhiều Module kết nối với nhau trong 1 channel, lưu ý tất cả các module cần được cấu hình là Broadcast và chung 1 channel, khi 1 module truyền, tất cả các module còn lại sẽ nhận, chức năng của các module là tương đương (Zigbee UART CC2530+PA V2 không thể nạp lại fw được nên không thể chuyển đổi các chế độ router, modem, client)



Hình 47 Chọn channel cho Zigbee UART CC2530+PA V2

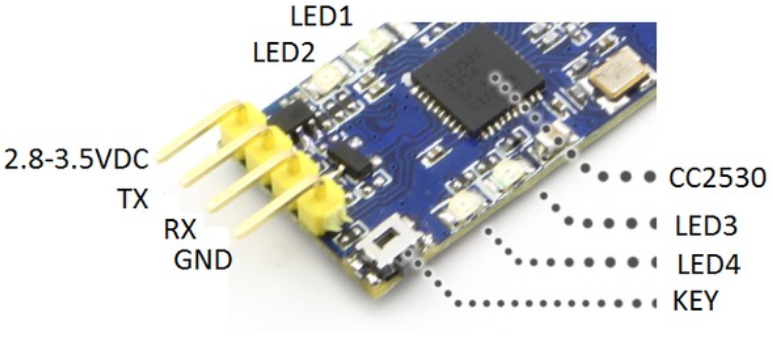
- **Bước 4:** Sau khi cấu hình kiểu truyền nhận, nhấn tì phím key lần cuối cùng sẽ thấy 4 Led cùng chớp báo hiệu hoàn thành cấu hình, tất cả các thông số được lưu và module bắt đầu hoạt động, lưu ý thiết đặt không đến được bước 4 sẽ không có hiệu lực, để thiết đặt lại các bạn quay lại bước 1.

Quy ước mắc dây Zigbee UART CC2530+PA V2:

Chú ý:

* Khi nạp code vào Arduino bạn không được cắm dây Zigbee UART CC2530+PA V2 vào Arduino vì khi cắm nạp code sẻ bị lỗi (Zigbee UART CC2530+PA V2 chiếm dây TX và RX của Arduino).
* Không thể cắm trực tiếp Zigbee UART CC2530+PA V2 vào Raspberry cần phải cắm thông qua bộ giải mã UART sang USB (cáp USB UART PL2303)

Mắc dây Zigbee UART CC2530+PA V2 cho Arduino và cáp UART sang USB:



Hình 48 Chân của Zigbee UART CC2530+PA V2

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Zigbee UART CC2530+PA V2 | Arduino | UART sang USB |
| 1 | VDC | 3.3V | Đỏ |
| 2 | TX | RX | Trắng |
| 3 | RX | TX | Xanh |
| 4 | GND | GND | Đen |

Bảng 6 Cắm chân của Zigbee UART CC2530+PA V2

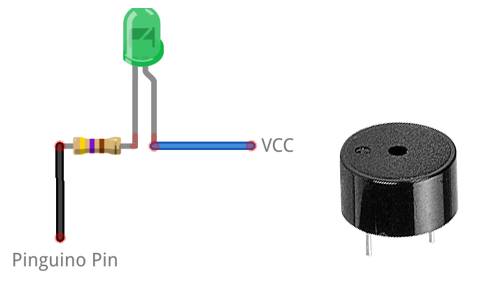
* + - 1. Lập trình đọc gửi thông tin Zigbee

Phần này sẻ được nói đến ở mục 2.4.5

* + 1. Điều khiển đèn Led đơn màu và còi
       1. Đèn led đơn màu và còi Buzzer

Đèn led đơn màu và còi buzzer thường được sử dụng để phát tín hiệu nhanh thông báo tới người dùng một cách đơn giản và nhanh chóng. Chúng ta có thể tùy chỉnh thời gian nhấp nháy, sáng, thời gian cói kêu, thời gian còi lập đi lập lại ứng với mỗi loại thông báo khác nhau tùy vào mình muốn. Quy định thông báo là tùy vào người dùng quy định, nó cũng giống như mã Morse

Quy ước mắc dây đèn led đơn màu và còi:



Hình 49 Chân cắm của đèn Led và còi

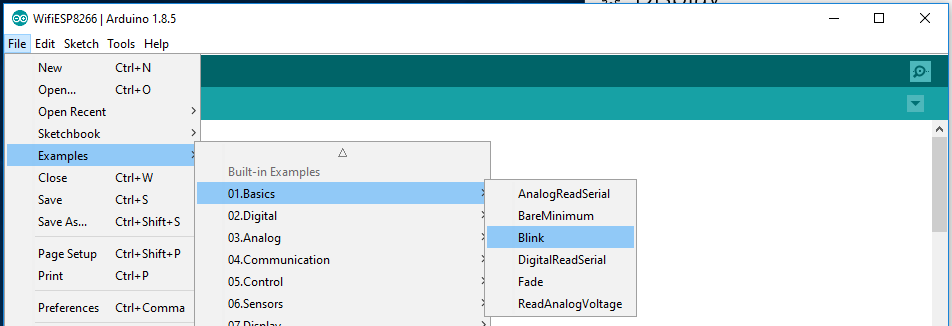
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Led/Buzzer | Arduino | Raspberry |
| 1 | VCC(chân dài, với đèn led cần nối tiếp thêm điện trở 330R) | GPIO Digital | GPIO Digital(với còi thì GND) |
| 2 | GND(Chân ngắn) | GND | GND(với còi thì GPIO Digital) |

Bảng 7 Cắm của đèn Led và còi

* + - 1. Lập trình điều khiển đèn led/ còi buzzer

**Trên arduino:**

Ở phần này đơn giản hơn bản chỉ cần mở thư viện có sẵn của arduino vào chọn mục như hình bên dưới và cần sửa lại biến “LED\_BUILTIN” thành chân dương Led/còi cắm vào



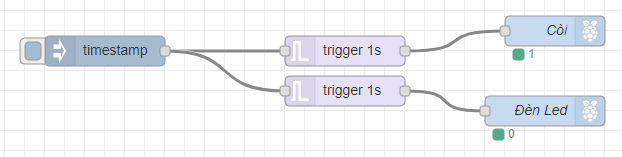
Hình 50 Lấy code mẫu nhấp nháy đèn led/còi kêu

**Trên Raspberry Pi:**

Ta thực hiện điều khiển thử đèn và còi ta sử dụng 3 node: timestamp, trigger và rpi GPIO out

Ở phần này để bật đèn led ta gửi gía trị 1 cho rpi GPIO out chân của đèn, và ngược lại giá trị 0 cho rpi GPIO out chân của đèn để tắt. Còn với còi thì ta lại thực hiện lệnh ngược lại hoàn toàn so với đèn led

Để thêm nhanh ví dụ ta có thể thêm Clipboard (nằm phần source code CD đính kèm) ở Node-red trên Raspberry pi



Hình 51 Flow thử bật tắt và còi

* + 1. Gửi và nhận giữa Arduino và Raspberry
       1. Gửi và nhận dữ liệu giữa Arduino(ESP8266) và Raspberry (Node-red) thông qua Wifi sử dụng giao thức UDP

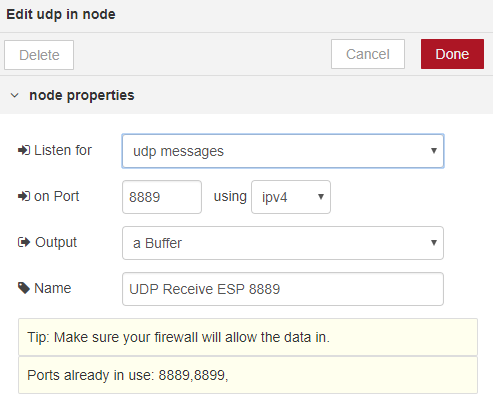
Ở phần này ta sẻ gửi dữ liệu đọc từ cảm biến LM35

Phần code và các thư viện đã được đính kèm trong CD đồ án, bây giờ chúng ta sẽ tiến hành tạo Flow trên Node-red trên Raspberry để nhận cũng như gửi dữ liệu xuống lại cho ESP8266. Các node ta sử dụng trong phần này sẽ được mô tả như bảng dưới:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại node | Tên node | Chức năng |
| 1 | Input | UDP in | Lấy dữ liệu đầu vào thông qua giao thức UDP bằng cổng Port được chỉ định (Dữ liệu ra sẽ là buffer). |
| 2 | Output | UPD out | Gửi dữ liệu tới địa chỉ IP và Port được chỉ định thông qua giao thức UDP (Dữ liệu ra sẽ là String). |
| 3 | Output | Debug | Dùng để xem output của một node có output bất kỳ. Dùng để kiểm tra dữ liệu đầu ra khi chạy. |
| 5 | Function | Function | Dùng để viết hàm sử dụng ngôn ngữ JavaScript. |
| 6 | Fucntion | Switch | Dùng để tách dữ liệu để xử lý hàm với điều kiện tách được chỉ định. |

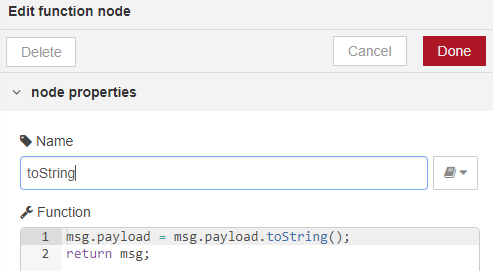
Bảng 8 Danh sách các node để gửi và nhận dữ liệu bằng UDP

Cấu hình từng node:

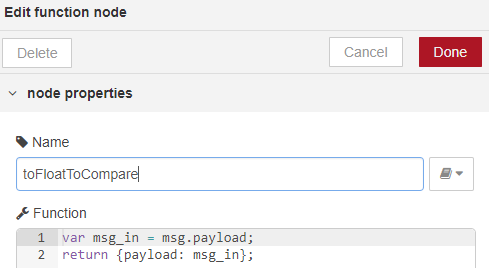


Hình 52 Cấu hình node UDP in

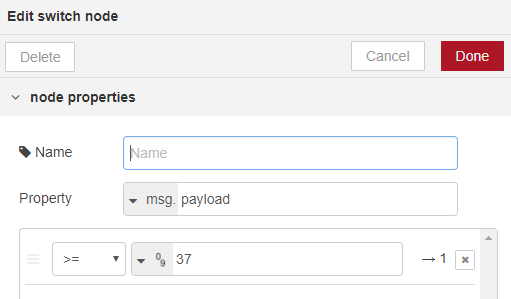
Với phần port ta điền Port đã cấu hình khi nạp code cho ESP8266, Output ta chọn a Buffer (vì phần code trên ESP8266 ta sử dụng giao thức UDP để gửi packet kiểu buffer).



Hình 53 Node Function toString để ép kiểu dữ liệu buffer sang String

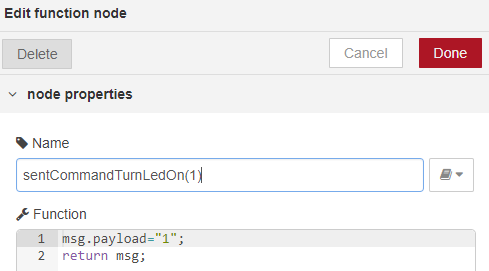


Hình 54 Node Function toFloatToCompare để chuyển kiểu dữ liệu cho so sánh



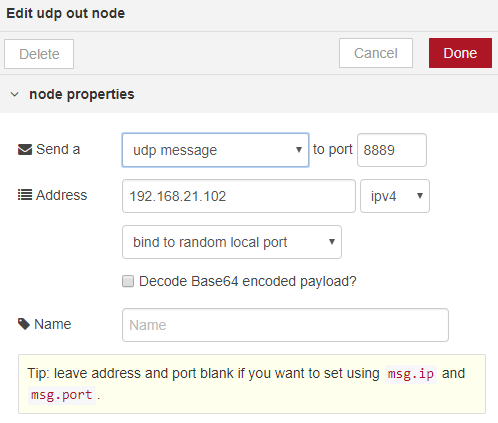
Hình 55 Cấu hình node Switch

Với Node này ta cấu hình điều kiện để thực hiện gửi lệnh xuống ESP8266, ở đây là khi nhiệt độ >= 37 thì ta sẽ thực hiện dòng 1 ở điều kiện ra (gọi node tạo câu lệnh).



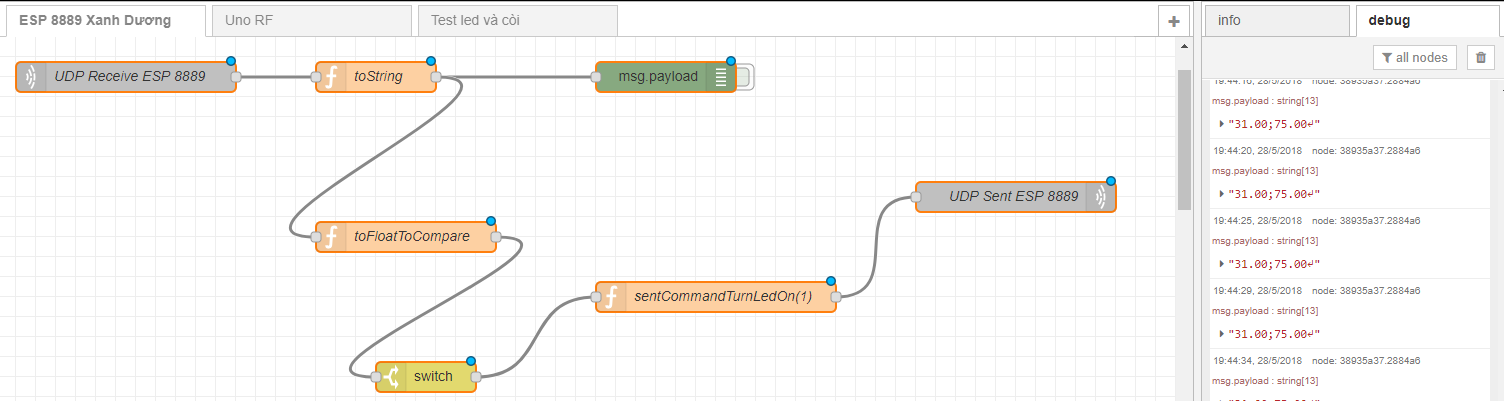
Hình 56 Cấu hình node tạo lệnh

Ta sẽ gửi lệnh (“1”) tương ứng với lệnh nháy đèn led ở chân D5 đang cắm trên ESP8266.



Hình 57 Cấu hình node UDP out

Với node này ta điền Port là Port đã nạp code trên ESP8266 (chung port giữa gửi và nhận), địa chỉ IP ta điền địa chỉ IP của ESP8266. Sau khi đã cài đặt các node như trên ta tiến hành nối dây giữa các node như sau để được một Flow hoàn chỉnh (gửi và nhận dữ liệu bằng UDP giữa Node-red và Arduino):



Hình 58 Flow gửi và nhận dữ liệu bằng UDP trên node-red

* + - 1. Gửi và nhận dữ liệu giữa Arduino Uno và Raspberry (Node-red) thông qua RF sử dụng 2 mạch Zigbee

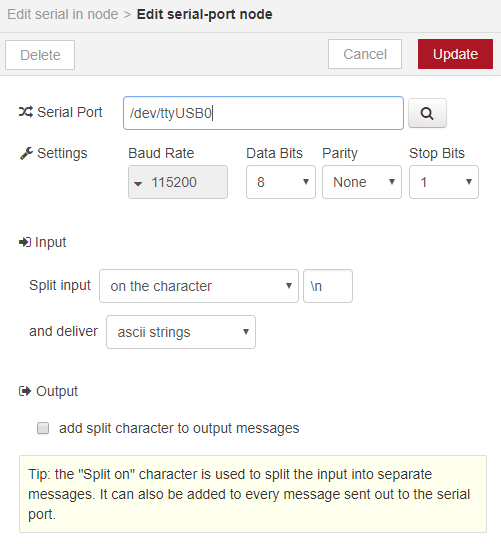
Phần này sử dụng dữ liệu đọc trên cảm biến DHT11

Phần code và các thư viện đã được đính kèm trong CD đồ án, bây giờ chúng ta sẽ tiến hành tạo Flow trên Node-red trên Raspberry để nhận cũng như gửi dữ liệu xuống lại cho Arduino Uno. Các node ta sử dụng trong phần này sẽ được mô tả như bảng dưới:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại node | Tên node | Chức năng |
| 1 | Input | Serial in | Lấy dữ liệu đầu vào thông qua giao thức Serial (Dữ liệu ra sẽ là String). |
| 2 | Output | Serial Out | Gửi dữ liệu xuống Arduino qua giao thức Serial (Dữ liệu ra sẽ là String và \n để kết thúc lệnh). |
| 3 | Function | Function | Dùng để viết hàm sử dụng ngôn ngữ JavaScript. |
| 4 | Function | Switch | Dùng để tách dữ liệu để xử lý hàm với điều kiện tách được chỉ định. |
| 5 | Output | Debug | Dùng để xem output của một node có output bất kỳ. Dùng để kiểm tra dữ liệu đầu ra khi chạy. |

Bảng 9 Danh sách các node để gửi và nhận dữ liệu bằng Serial sử dụng RF

Cấu hình từng node:

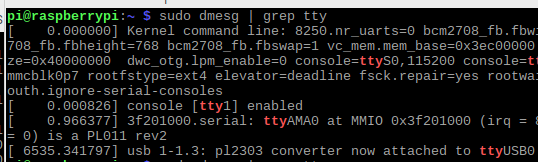


Hình 59 Cấu hình node Serial in

Với node Serial in ta cần điền Serial Port mà ta đã cắm Zigbee vào Raspberry thông qua cổng USB (sử dụng mạch chuyển UART sang USB PL2303). Để biết cổng port ta đang sử dụng trên Raspberry ta mở Terminal và sử dụng câu lệnh dưới đây:

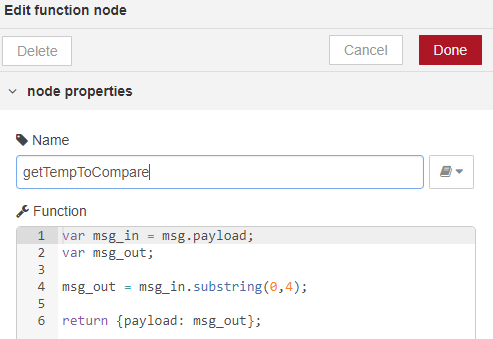
|  |
| --- |
| sudo dmesg | grep tty |

Kết quả sẽ hiển thị như hình dưới:



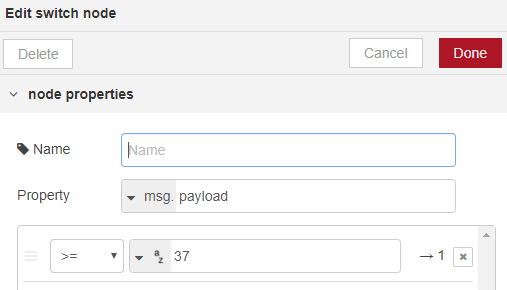
Hình 60 Kiểm tra cổng port cắm Zigbee trên Raspberry

Ta sẽ để ý dòng pl2303 converter now attached to ttyUSB0, với ttyUSB0 là port ta sẽ điền trên node Serial in. Mục baudrate ta điền baudrate đã cài đặt ở trên Zigbee (ở đây là 115200), các tùy chọn khác ta để mặc định.



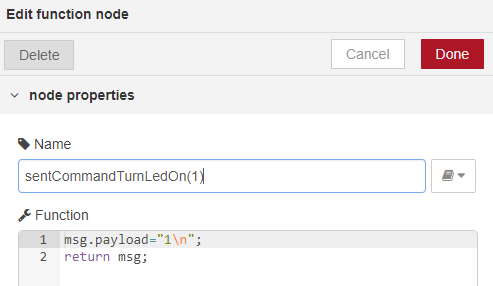
Hình 61 Cấu hình node getTempToCompare

Node này ta sử dụng để tách dữ liệu nhiệt độ ra khỏi chuỗi dữ liệu được gửi lên vì DHT11 đo được cả nhiệt độ và độ ẩm. Định dạng của chuỗi dữ liệu gửi lên từ Arduino sẽ có dạng “xx, xx; yy, yy” với x và y là các số tự nhiên. Sau khi lấy ra nhiệt độ ta chuyển kiểu dữ liệu về dạng số để chuyển cho node Switch.



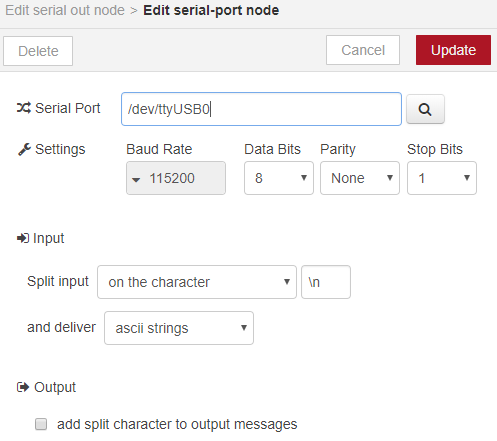
Hình 62 Cấu hình node Switch

Với Node này ta cấu hình điều kiện để thực hiện gửi lệnh xuống Arduino Uno, ở đây là khi nhiệt độ >= 37 thì ta sẽ thực hiện dòng 1 ở điều kiện ra (gọi node tạo câu lệnh).



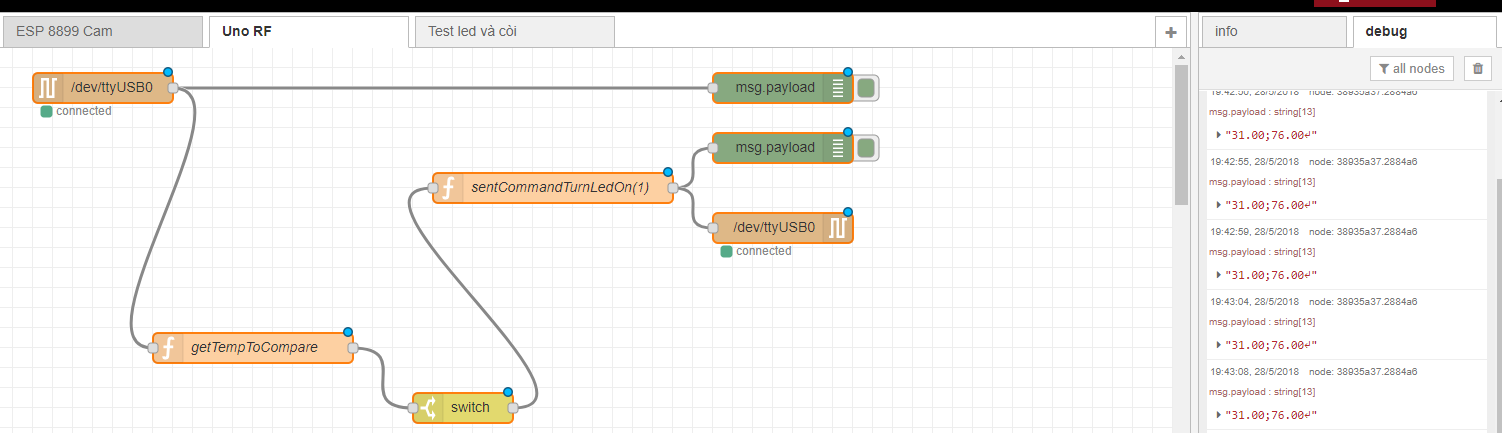
Hình 63 Cấu hình node tạo lệnh - Serial

Sau khi so sánh mà thỏa điều kiện ở node Switch thì node này sẽ được gọi và tạo lệnh sau đó thông qua node Serial out tới Arduino Uno.



Hình 64 Cấu hình node Serial out

Cấu hình node Serial out sẽ tương tự với Serial in (cùng baudrate và cùng Serial Port). Sau khi đã cài đặt các node như trên ta tiến hành nối dây giữa các node như sau để được một Flow hoàn chỉnh (gửi và nhận dữ liệu bằng Serial qua RF giữa Node-red và Arduino Uno):



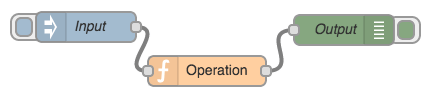
Hình 65 Flow gửi và nhận dữ liệu bằng Serial qua RF trên node-red

Chương 3 - Nghiên cứu thực nghiệm

* 1. Giới thiệu về Node-red

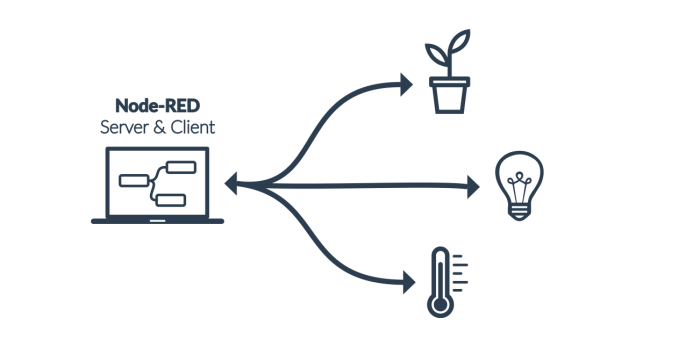
Node-RED được dựa trên Node.js, nó có thể được xem như một web server mà bạn có thể cấu hình tùy chỉnh các chức năng gọi là “flow” từ bất kỳ trình duyệt nào trên máy tính. Mỗi ứng dụng Node-RED bao gồm các node có thể liên kết được với nhau với các dạng là input, output và operation.

Một ví dụ đơn giản để chúng ta có thể hình dung được các node khác nhau sẽ tương tác như thế nào



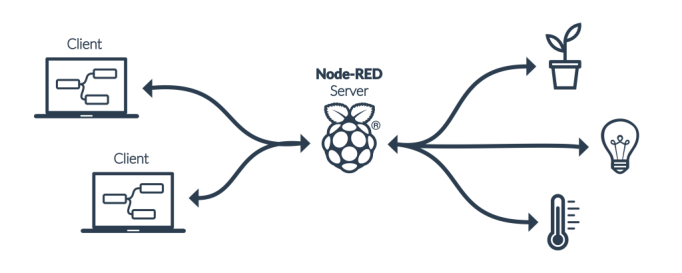
Hình 66 Tương tác giữa các node khác nhau

Với Node-RED ta có thể hình dung cách tương tác và giao tiếp với các thiết bị một cách tổng quan như hình dưới. Ở đây máy tính của mình (raspberry pi) sẽ đóng vai trò là server và client



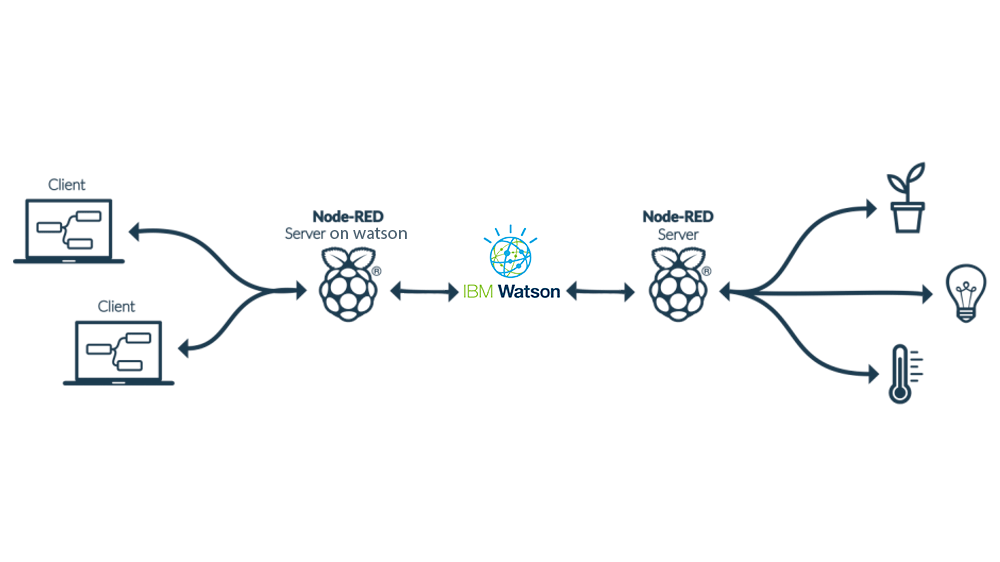
Hình 67 Raspberry tương tác với các cảm biến và thiết bị

Nếu dùng Raspberry hay OrangePi thì thiết bị này sẽ đóng vai trò là Server, còn lại sẽ là client như hình



Hình 68 Node-red chạy trên Raspberry hay OrangePi

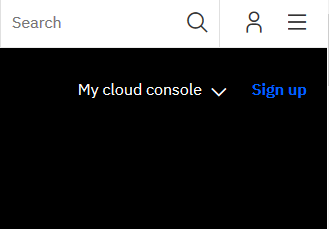
Còn bài với đồ án này chúng ta sẻ sử dụng thêm IBM Watson nằm trung gian giữa Client và các cảm biến, thiết bị như hình bên dưới



Hình 69 IBM Watson nằm trung gian giữa Client và các cảm biến, thiết bị

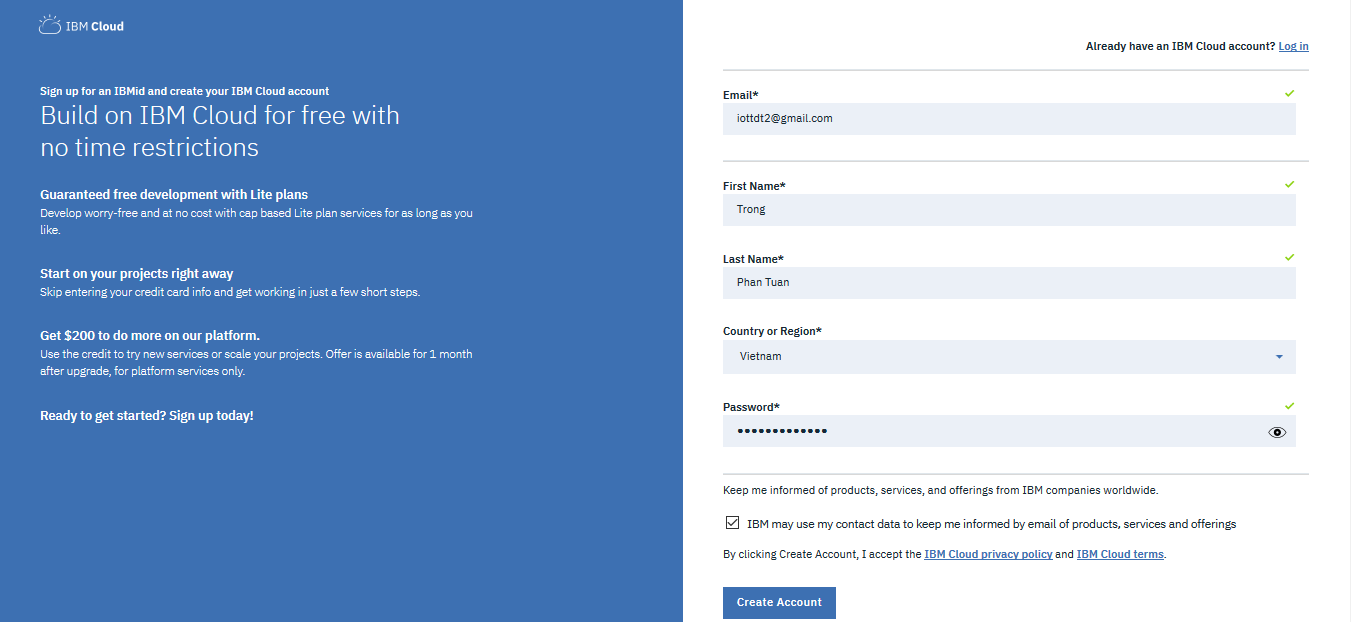
* 1. Xử lý giữa Node-red trên Raspberry và IBM Watson
     1. Đăng ký IBM Waston

Ta tiến hành đăng ký tài khoản IBM Bluemix để sử dụng dịch vụ của IBM. Truy cập vào đường dẫn <https://www.ibm.com/cloud/> để bắt đầu đăng ký. Ta nhấn vào nút Sign Up màu xanh góc bên phải để chuyển vào trang đăng ký tài khoản như hình dưới:



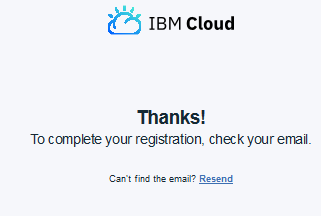
Hình 70 Nhấn vào Sign Up

Ta nhập đầy đủ thông tin vào form đăng ký và nhấn vào nút Create Account như hình minh họa bên dưới:



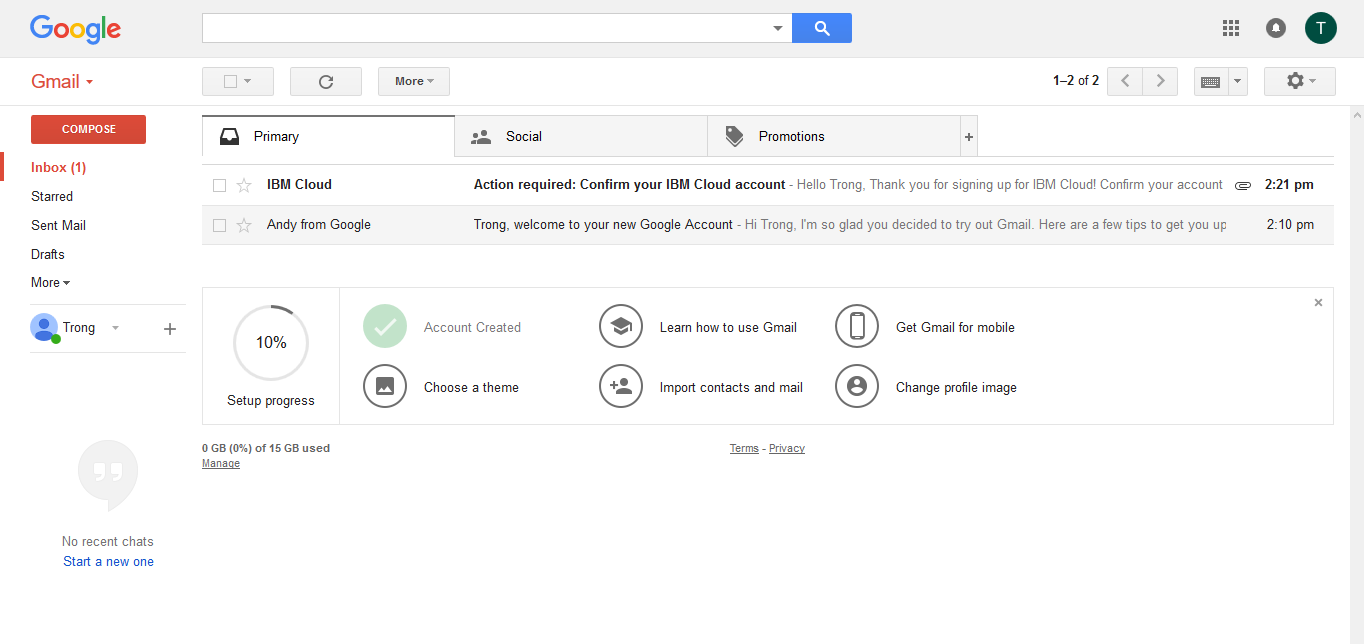
Hình 71 Form đăng ký tài khoản IBM

Lưu ý: Đối với phần mật khẩu cần có ít nhất 1 ký tự in hoa, 1 ký tự đặc biệt và ít nhất một chữ số (độ dài yêu cầu từ 8-31 ký tự). Hiện tại IBM không yêu cầu phải thêm thẻ tín dụng trong quá trình đăng ký nhưng nếu sau khi đăng ký thành công và ta có thẻ tín dụng thì có thể thêm thẻ để được tặng 200 đô sử dụng dịch vụ của IBM trong vòng 1 tháng.

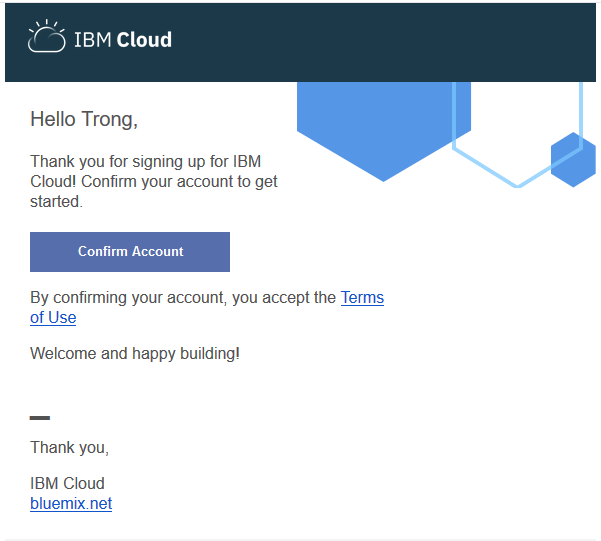


Hình 72 Yêu cầu xác thực email

Nếu thông tin đăng ký hợp lệ ta sẽ chuyển tới trang yêu cầu xác thực địa chỉ email. Các bước xác thực email sẽ được biểu diễn như bên dưới:

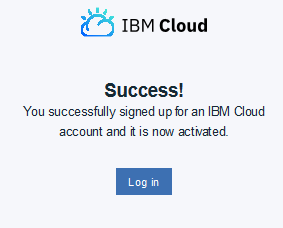


Hình 73 Truy cập vào địa chỉ email xác thực



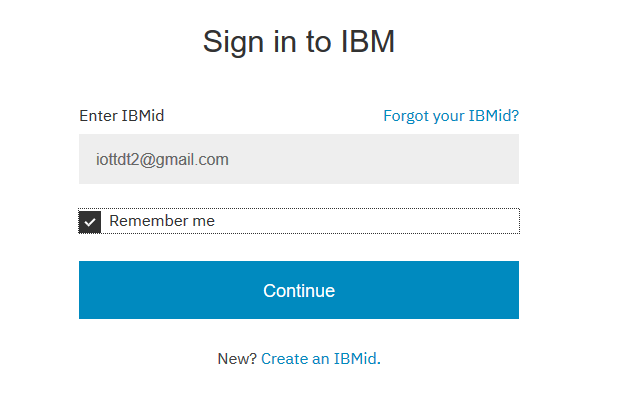
Hình 74 Mở email được IBM gửi

Để xác thực ta nhấn vào nút Confirm Account trong email. Nếu thành công ta sẽ được chuyển tới trang web hiển thị như bên dưới:



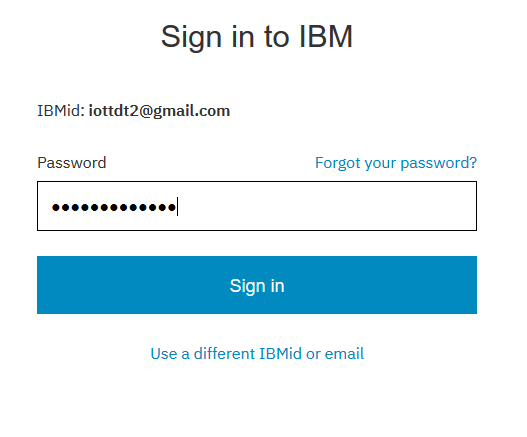
Hình 75 Đăng ký thành công

Như vậy quá trình đăng ký tài khoản đã hoàn tất ta nhấn vào nút Log in để đăng nhập và sử dụng các dịch vụ của IBM.



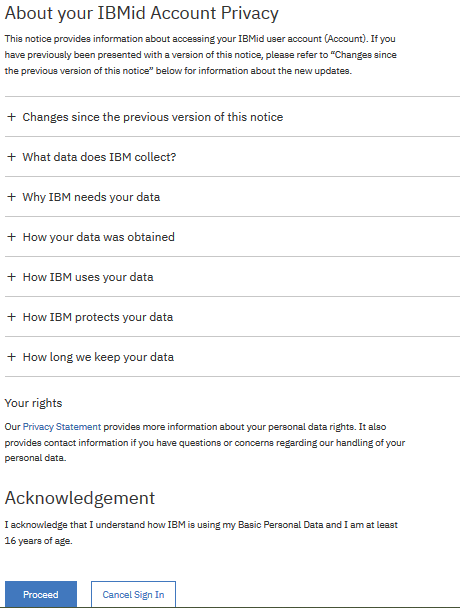
Hình 76 Form Login tài khoản IBM-Nhập địa chỉ email

Sau khi nhập địa chỉ email ta vừa xác thực ta nhấn Continue để chuyển sang bước nhập mật khẩu



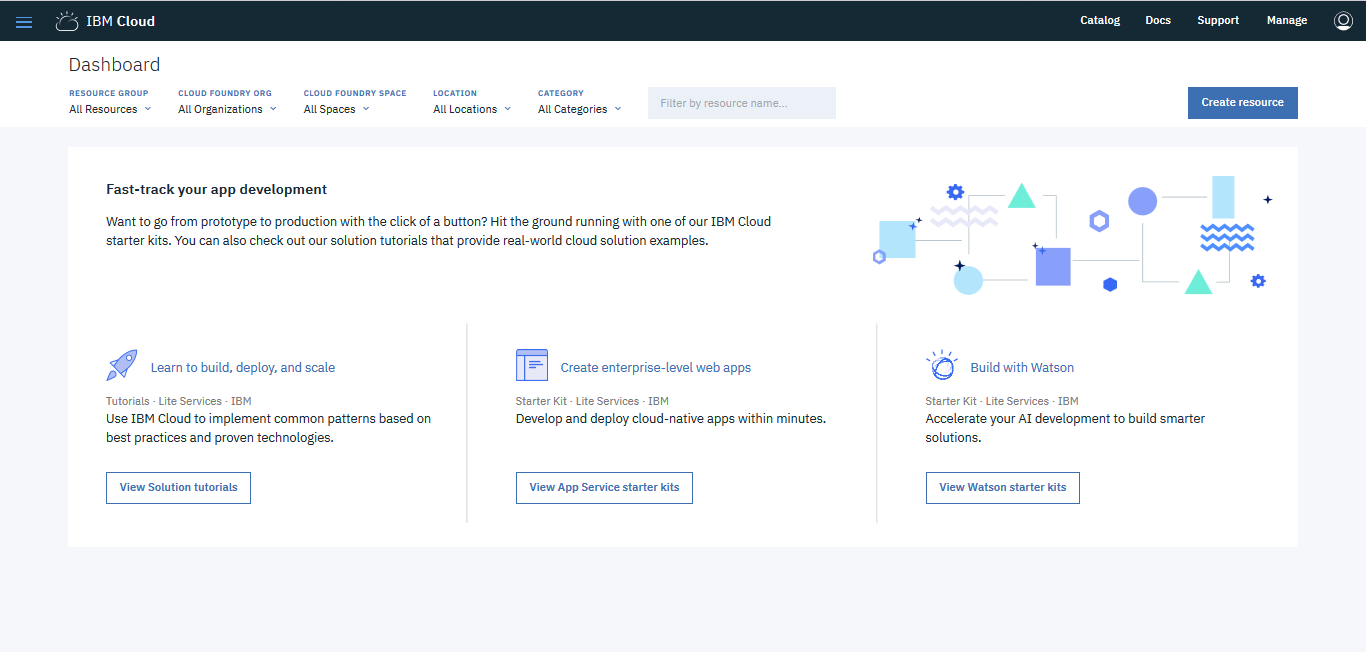
Hình 77 Form Login tài khoản IBM-Nhập mật khẩu

Sau khi điền mật khẩu ta nhấn Sign in để tiến hành truy cập vào trang dịch vụ của IBM. Vì đây là tài khoản mới tạo nên IBM sẽ hiển thị một vài thông tin cơ bản và quyền lợi của người dùng và yêu cầu ta xác nhận để khởi tạo dịch vụ



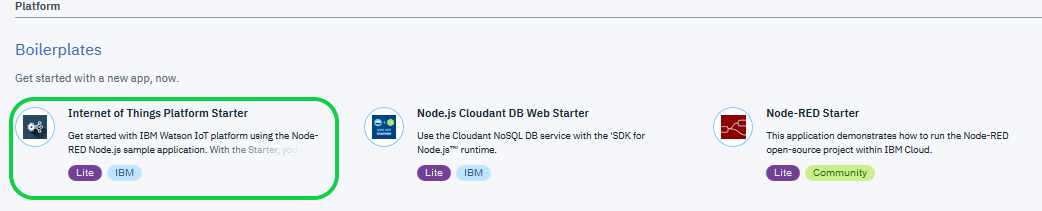
Hình 78 Giới thiệu các điều khoản IBM

Ta nhấn vào Proceed để hoàn tất quá trình xác nhận.



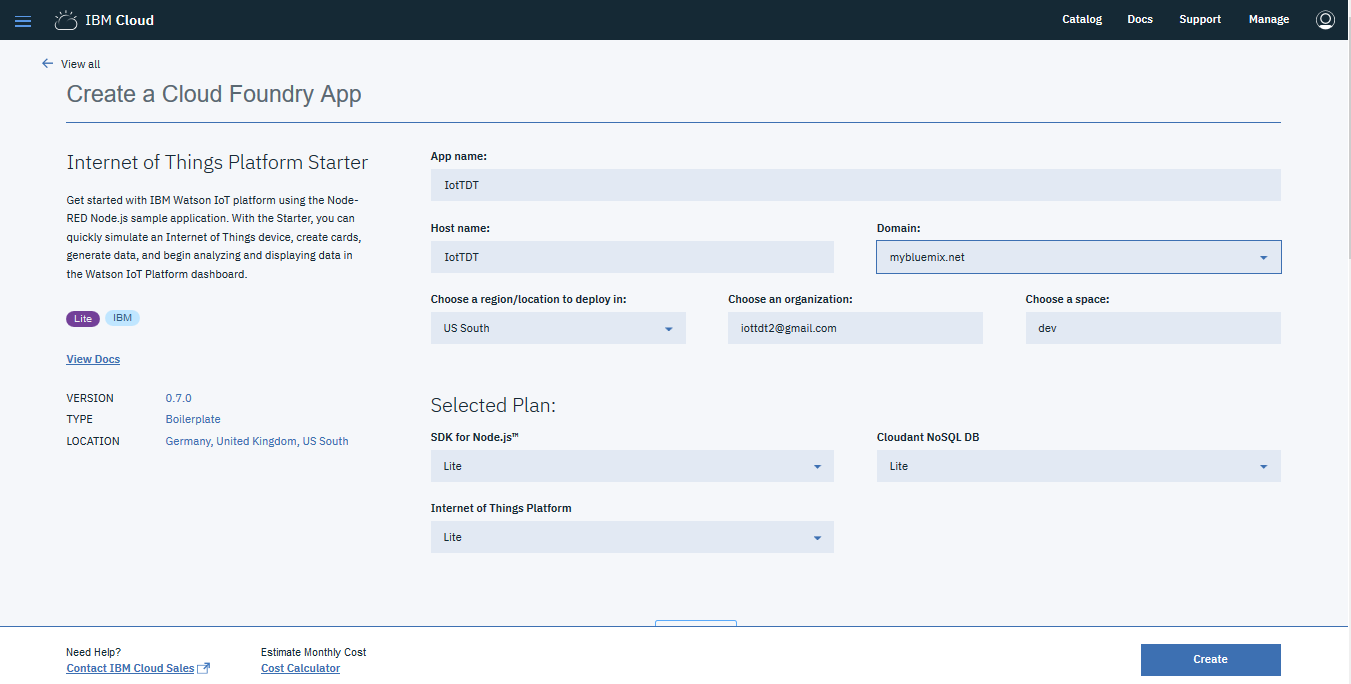
Hình 79 Giao diện Dashboard khi mới tạo tài khoản IBM

Ta tiến hành tạo Resource để khởi tạo các dịch vụ cho tài khoản IBM. Sau khi nhấn vào Create Resource ta sẽ chuyển đến trang chọn dịch vụ của IBM, với đồ án này ta sẽ cần 3 dịch vụ của IBM bao gồm: Internet of Things Starter, Node-red Starter và Node.js Cloudant DB Web Starter. Đầu tiên ta sẽ tạo Internet of Things Starter trước, để tạo ở giáo diện tạo Resource ta kéo xuống mục Platform và nhấn chọn vào Internet of Things Starter như hình dưới:



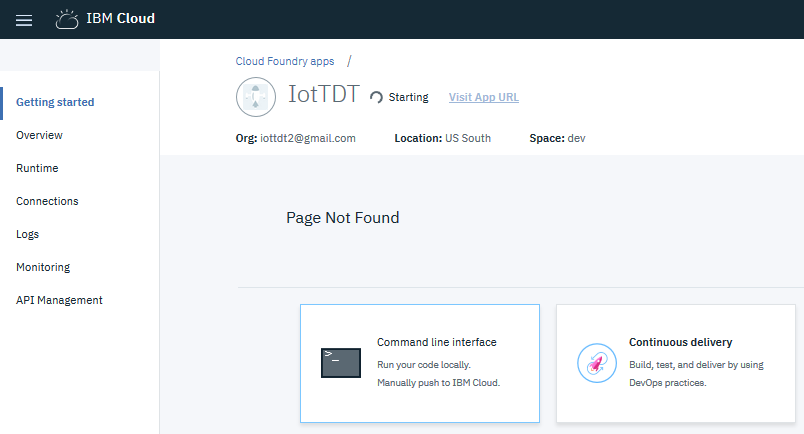
Hình 80 Đăng ký dịch vụ Internet of Things Platform

Ta điền vào tên dịch vụ ta khởi tạo (Lưu ý nếu chọn dịch vụ Internet of Things Platform (ở đây để demo ta chọn dịch vụ bản Lite-Miễn phí của IBM) thì dịch vụ Cloudant bản Lite cũng tự động được chọn.



Hình 81 Điền tên App trên IBM

Sau khi chọn các gói dịch vụ tương ứng ta nhấn vào nút Create ở bên dưới để tạo dịch vụ. Chờ một chút để IBM kích hoạt dịch vụ:



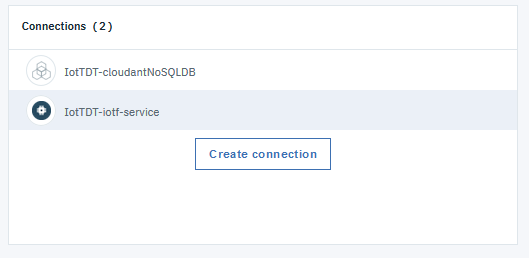
Hình 82 IBM kích hoạt dịch vụ

Nếu quá trình khởi tạo bị đứng ở phần Starting quá lâu ta nhấn vào mục Menu bên trái chữ IBM Cloud rồi chọn Dashboard để hiển thị tất cả dịch vụ. Ta nhấn chọn vào tên dịch vụ ở mục Cloud Foundry Applications để chuyển sang trang quản lý.



Hình 83 Chọn Cloud Foundry Applications

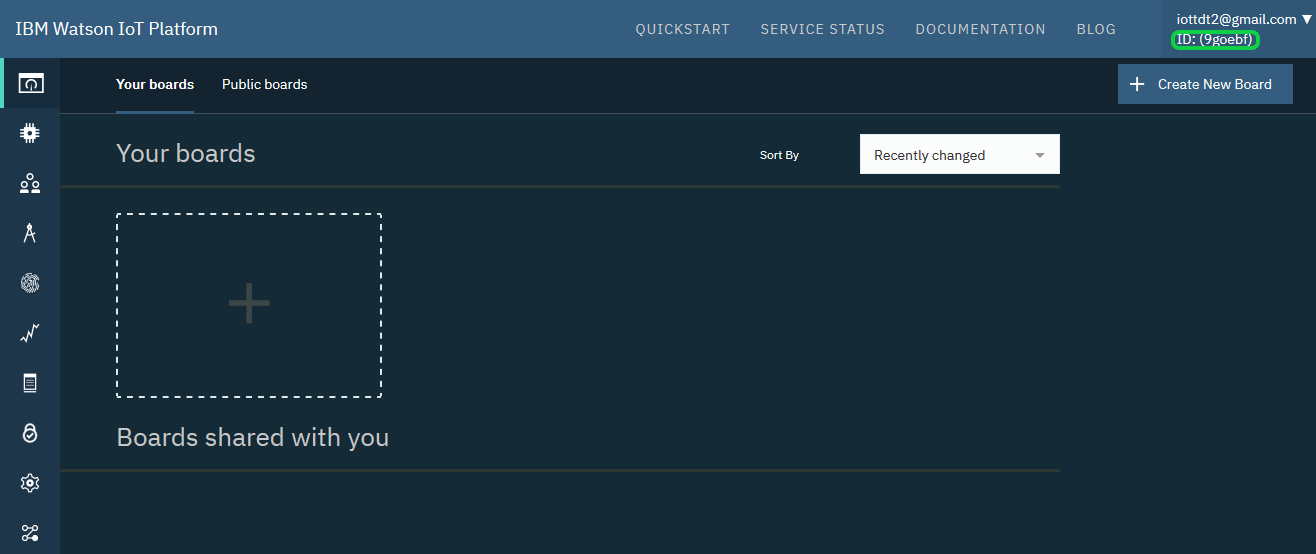
Sau khi chuyển đến trang quản lý dịch vụ ta chọn mục iotf-service ở phần Connections.



Hình 84 Chọn IoT Platform

Sau khi nhấn vào IoT Platform ta sẽ chuyển tới trang dịch vụ của Watson IBM, ta chọn Launch để truy cập dịch vụ của Watson IoT.

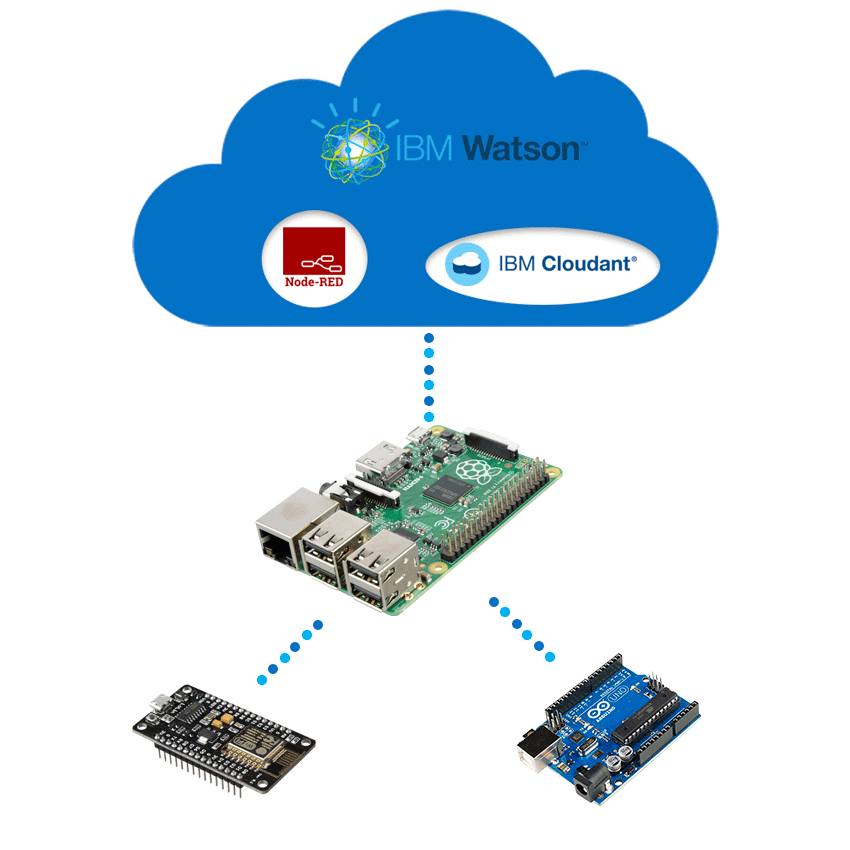
Sau khi nhấn Launch ta sẽ được chuyển đến trang của IBM Watson IoT với giao diện như bên dưới:



Hình 85 Giao diện Watson IoT Dashboard

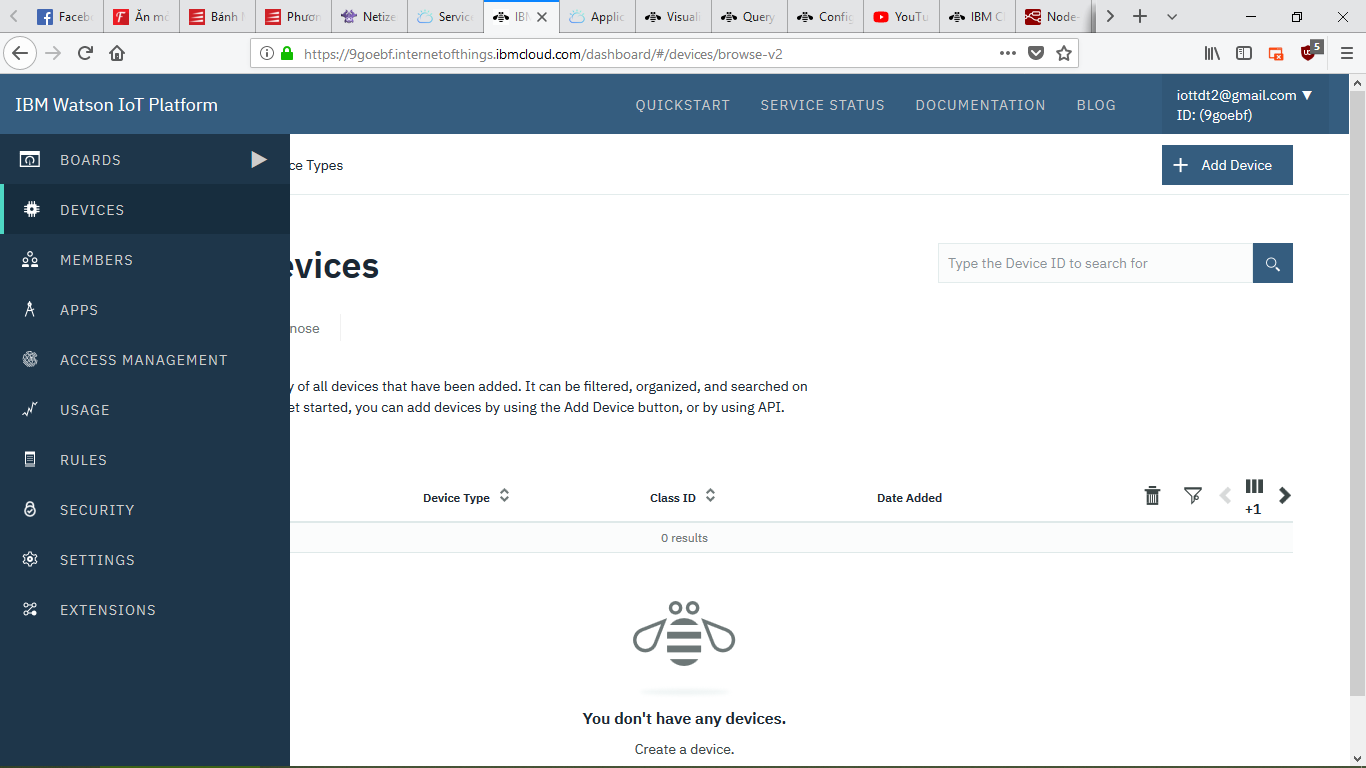
Lưu ý: Phần ID được đóng khung ở góc bên phải màn hình dung để liên kết thiết bị trong node-red với Watson.

Tiếp theo chúng ta sẽ khởi tạo thiết bị (mẫu thiết bị) nhằm tạo liên kết với thiết bị (Raspberry) và gửi dữ liệu lên IBM thông qua Watson IoT.



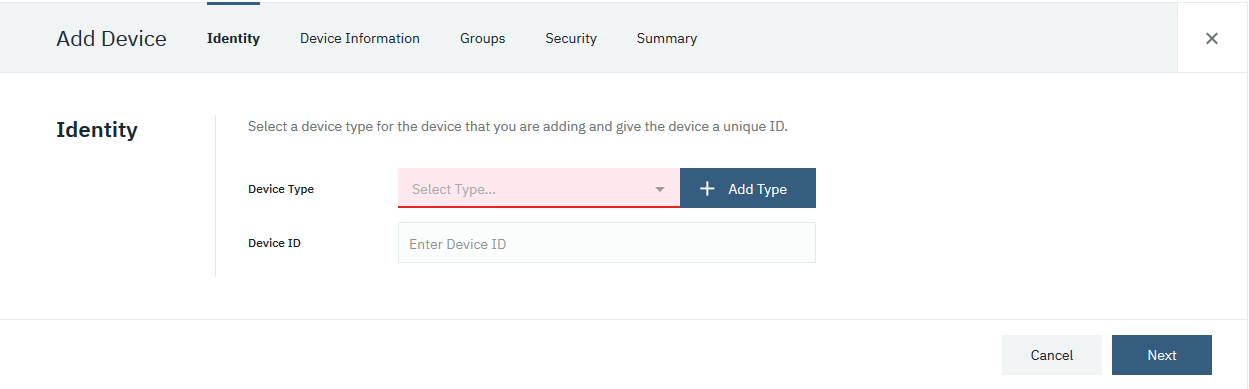
Hình 86 Sơ đồ kết nối tổng quan giữ các thiết bị và IBM Watson

Để khởi tạo thiết bị ta rê chuột vào tab menu ở bên trái màn hình và chọn mục Devices như hình dưới:



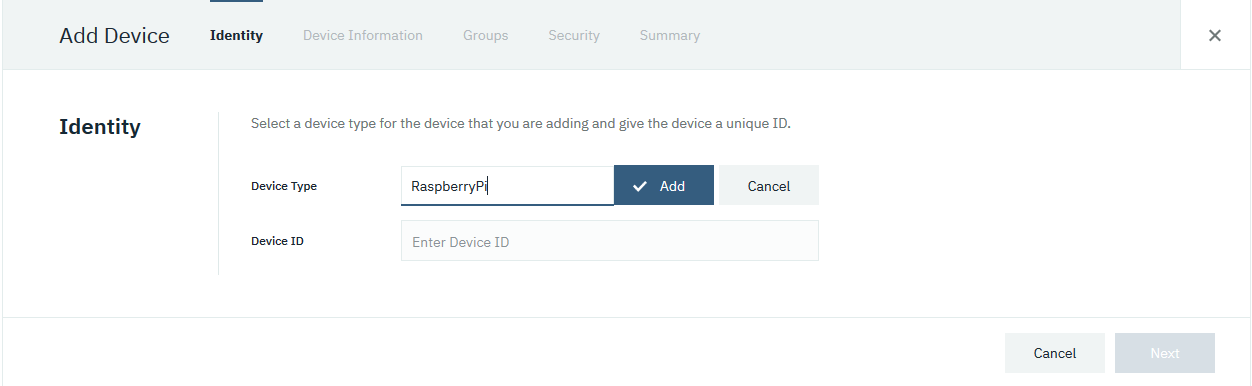
Hình 87 Chọn tab Devices

Sau đó ta nhấn vào Add Device để thêm thiết bị.



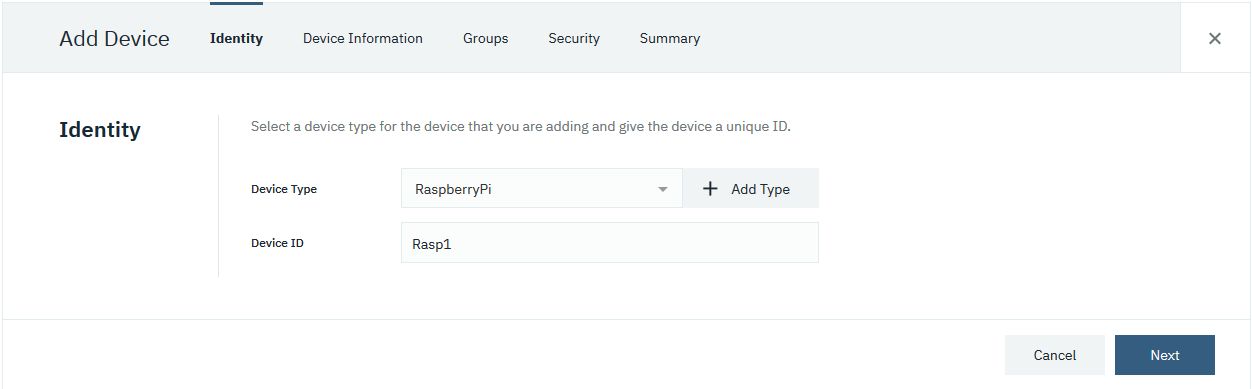
Hình 88 Thêm thiết bị trên IBM Watson

Vì đây là lần đầu khởi tạo thiết bị nên ta phải khởi tạo loại thiết bị trước. Ta nhấn vào Add Type để chuyển đến giao diện thêm loại thiết bị.



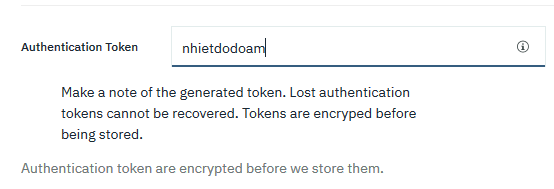
Hình 89 Thêm loại thiết bị

Ta điền tên loại thiết bị vào mục DeviceType và nhấn Add để thêm loại thiết bị. Vì ở đồ án này ta chủ yếu sử dụng Raspberry để giao tiếp với IBM Watson nên ta chỉ cần khởi tạo một loại thiết bị là RaspberryPi.



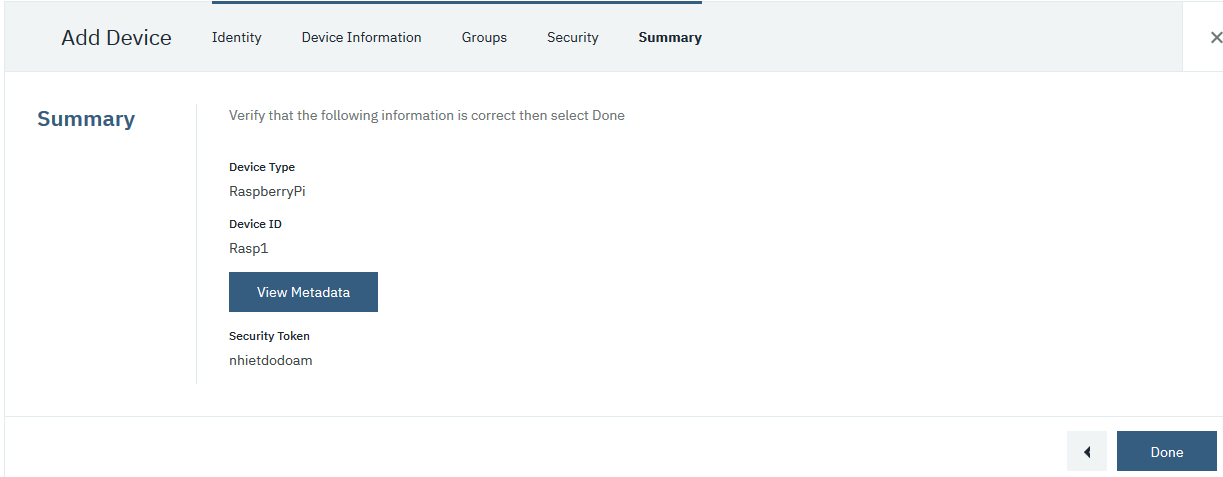
Hình 90 Tạo thiết bị trên Watson IoT

Sau khi khởi tạo loại thiết bị ta sẽ được chuyển về lại giao diện thêm thiết bị ta sẽ chọn ở mục Device Type loại thiết bị ta vừa tạo và RaspberryPi và ở mục Device ID chính là mã định danh cho thiết bị. Mã định danh này dùng để phân biệt nếu trường hợp có nhiều loại thiết bị có cùng loại kết nối tới IBM Watson. Sau khi điền Device ID xong ta nhấn Next cho đến khi tới mục Security (các phần không đề cập sẽ để trống và nhấn Next). Tại mục Sercurity này ta sẽ tạo khóa để liên kết với Watson IoT, ở phần Authentication Token ta điền vào khóa để liên kết và nhấn Next.

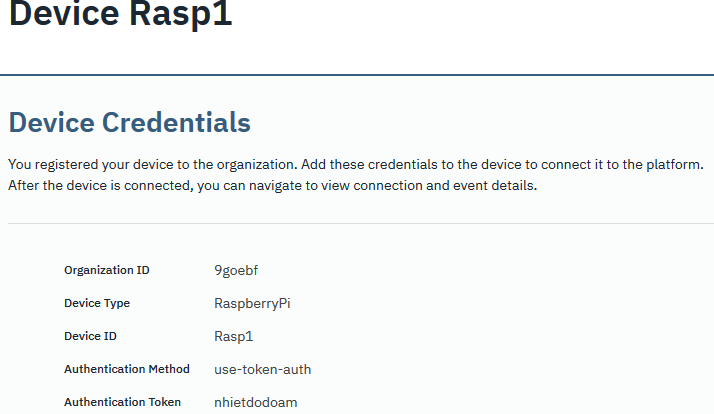


Hình 91 Tạo mật khẩu liên kết thiết bị

Sau đó ta nhấn Done để hoàn tất quá trình tạo thiết bị (ở mục này sẽ hiển thị lại các thông tin về thiết bị mà ta vừa tạo để kiểm tra).



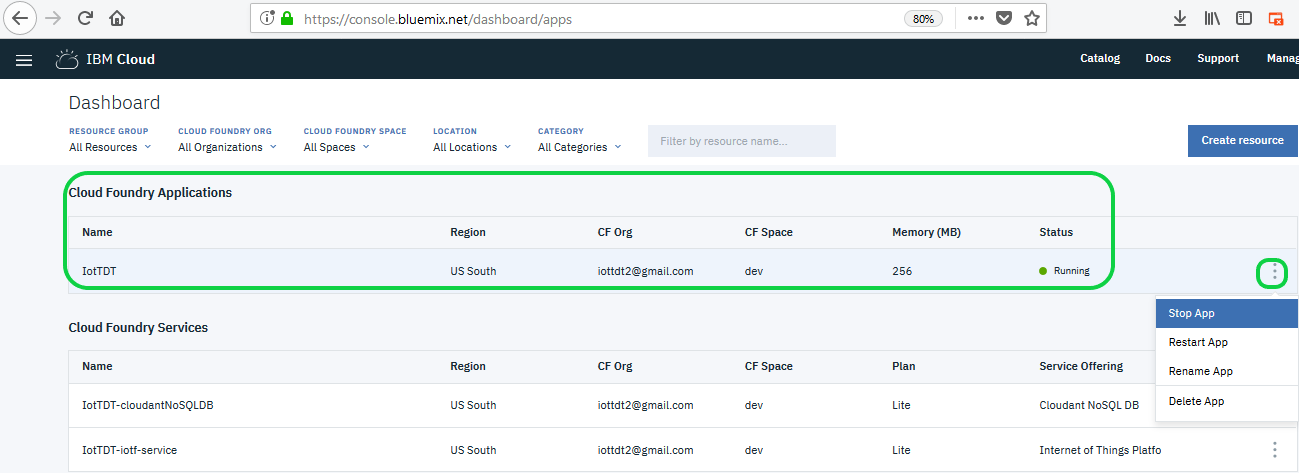
Hình 92 Hoàn tất khởi tạo thiết bị trên Watson IoT



Hình 93 Thông tin thiết bị đã đăng ký với Watson

Lưu ý: Với Token đã tạo thì không thể thay đổi, nếu quên Token ta buộc phải tạo lại thiết bị mới để tiếp tục sử dụng dịch vụ.

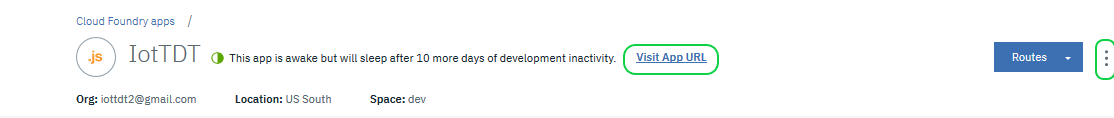
Như vậy ta đã hoàn tất quá trình đăng ký tài khoản và dịch vụ của IBM để có thể liên kết cũng như nhận dữ liệu từ Raspberry. Với các tài khoản mới đăng ký thì mặc định dịch vụ node-red đã được tích hợp chung lúc khởi tạo IoT platform. Để truy cập vào node-red trên IBM ta vào <https://console.bluemix.net> và chọn mục Dashboard.



Hình 94 Giao diện Dashboard trên IBM Cloud

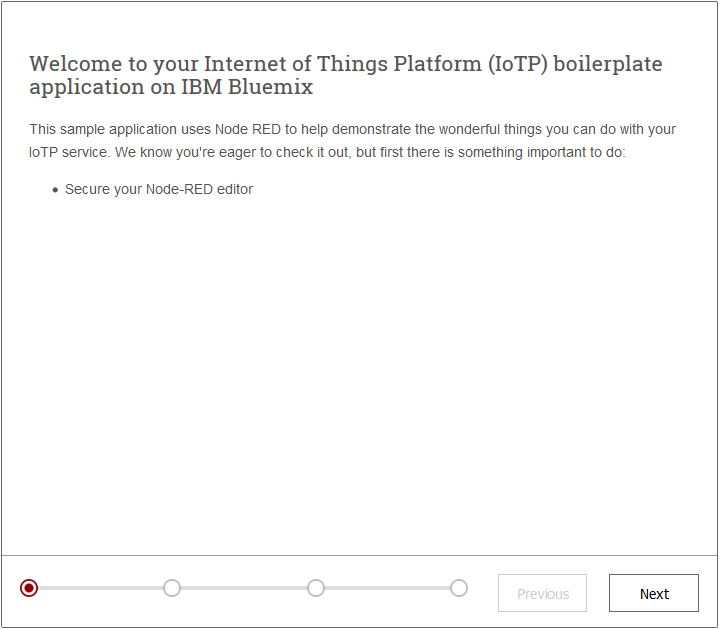
* + 1. Cài đặt Node-red trên IBM

Để kích hoạt dịch vụ Node-red (Cloud Foundry Applications) ta nhấn vào nút 3 chấm bên cạnh tên dịch vụ và chọn Start (chờ một lúc cho dịch vụ được khởi động). Như hình là dịch vụ đã được khởi chạy hoàn tất rồi, ta nhấn vào tên dịch vụ để chuyển tới trang quản lý dịch vụ. Tại đây tương tự như ở Dashboard ta có thể ngừng/khởi chạy dịch vụ và truy cập dịch vụ.



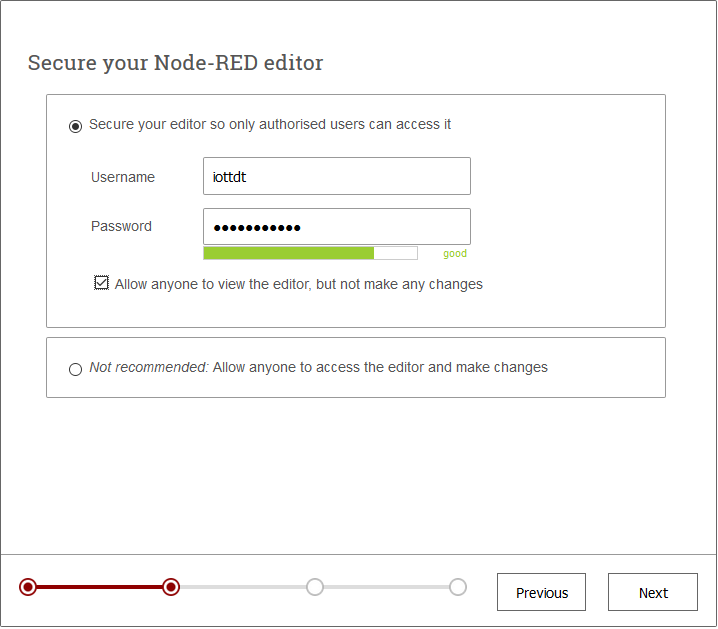
Hình 95 Trang quản lý dịch vụ Node-red trên IBM

Để truy cập vào Node-red trên IBM ta nhấn vào đường dẫn Visit App URL để chuyển sang IBM Node-red. Với lần đầu truy cập node-red sẽ hiển thị một vài thông tin hướng dẫn và giới thiệu:



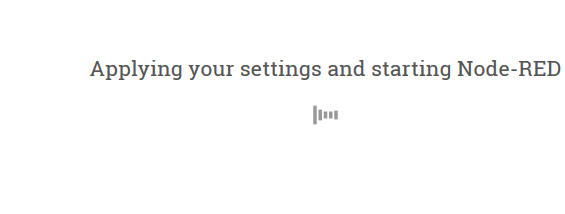
Hình 96 Truy cập Node-red

Ta nhấn Next để chuyển qua bước tiếp theo là khởi tạo tài khoản và mật khẩu quản lý node-red (tài khoản này dung để chỉnh sửa Flow trên node-red). Dựa vào mục đích của dự án mà ta có thể chọn không cần tài khoản hoặc khi tạo tài khoản vẫn cho người khác xem nhưng không được chỉnh sửa khi không có tài khoản đăng nhập như hình dưới:



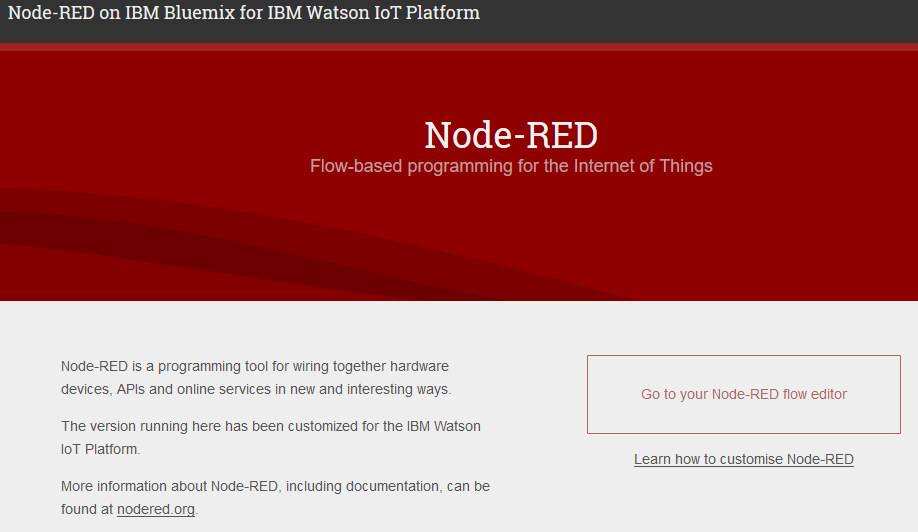
Hình 97 Tạo tài khoản quản lý Node-red trên IBM

Sau đó ta nhấn Next – Next và Finish để hoàn tất cài đặt Node-red trên IBM. Quá trình lưu cài đặt có thể mất vài phút.



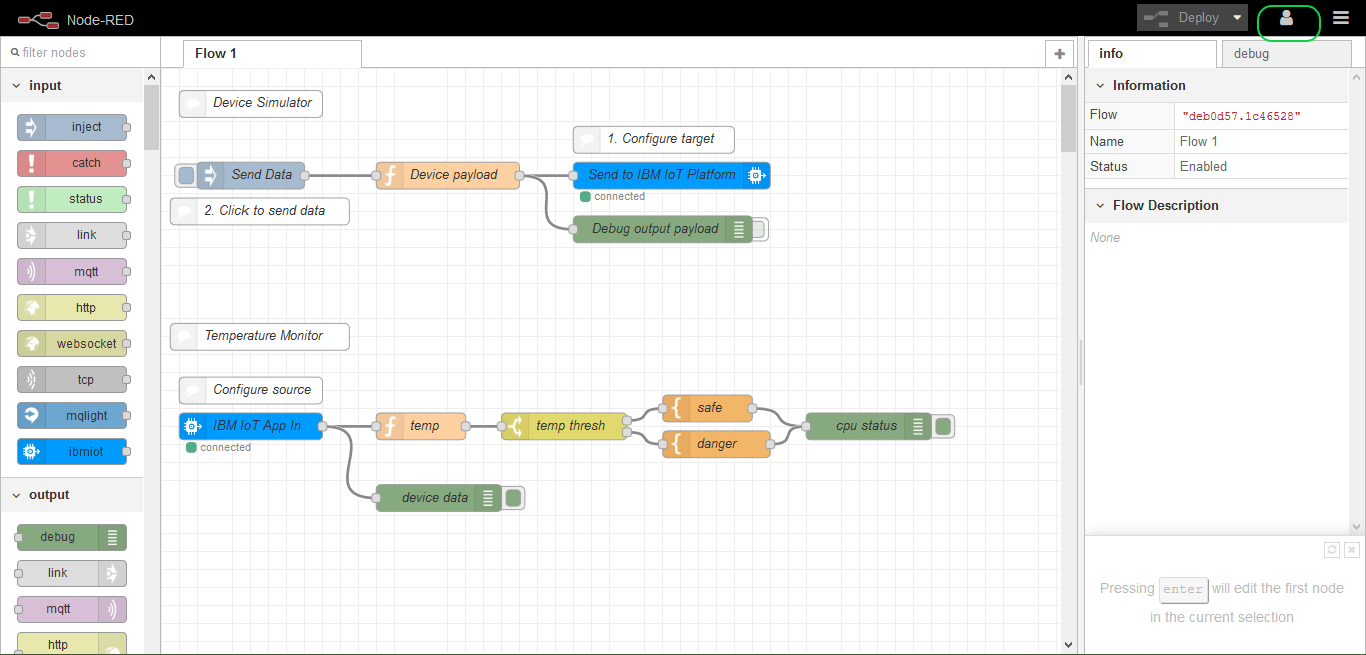
Hình 98 Hoàn tất cài đặt Node-red trên IBM

Sau khi lưu các thiết đặt xong ta sẽ được chuyển tới trang của Node-red như bên dưới:



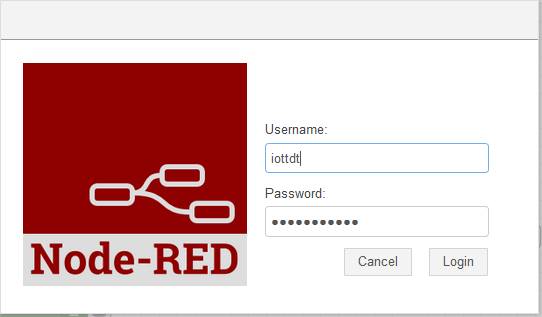
Hình 99 Giao diện trang Node-red trên IBM

Ta chọn Go to your Node-Red flow editor để truy cập vào node-red để tạo và chỉnh sửa flow.



Hình 100 Giao diện Flow trên Node-red IBM

Như trên hình thì mặc định IBM sẽ tạo một Flow demo cho chúng ta thử nghiệm, các flow này có thể xóa được. Ta nhấn vào hình người (vòng xanh lá trên hình) để đăng nhập tài khoản quản lý node-red trên IBM.



Hình 101 Đăng nhập quản lý trên Node-red IBM

Sau khi đăng nhập tài khoản đã tạo ở quá trình cài đặt node-red ta có thể tạo và chỉnh sửa cũng như kích hoạt các flow. Như vậy ta đã hoàn thành quá trình cài đặt Node-red trên IBM để quản lý IBM Watson.

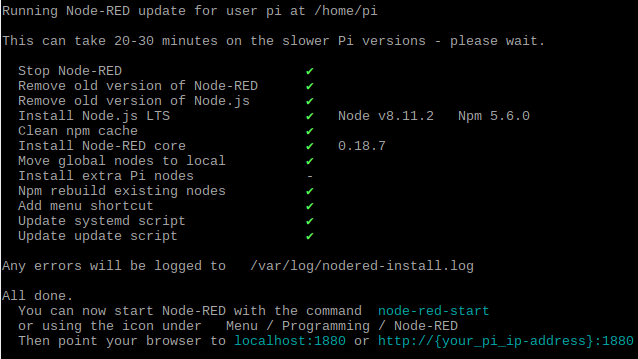
* + 1. Xử lý gửi và nhận dữ liệu giữa Raspberry và IBM Watson
       1. Gửi dữ liệu

Ý tưởng của dự án là lập một trạm tập trung dữ liệu bằng Raspberry và nhận dữ liệu từ các Arduino và gửi lên cho IBM Watson, từ đó ta sử dụng node-red trên IBM để đọc dữ liệu từ IBM Watson sau đó cung cấp giao diện xử lý cho người dùng hoặc lập trình viên. Vì vậy sẽ có 3 giai đoạn: giai đoạn 1 là gửi và nhận dữ liệu giữa các Arduino và Raspberry (Chương 2 mục 2.4.5); giai đoạn 2 là gửi và nhận dữ liệu giữa Raspberry (sử dụng node-red) và IBM Watson; giai đoạn 3 là gửi và nhận dữ liệu giữa node-red trên IBM và IBM Watson. Tại chương này ta sẽ tiến hành giai đoạn 2 và giai đoạn 3.

Giai đoạn 2 : Gửi và nhận dữ liệu giữa Raspberry và IBM Watson. Để thực hiện giai đoạn này ta sẽ cần cập nhật lại phiên bản node-red trên Raspberry để có đầy đủ thư viện và các node sử dụng trong bài. Cách thực hiện như sau : mở Terminal và gõ từng dòng lệnh dưới đây:

|  |
| --- |
| sudo apt-get update -y  sudo apt-get upgrade -y  update-nodejs-and-node-red -y |

Sau khi quá trình cập nhật node-red trên Raspberry được thực hiện xong Terminal sẽ hiển thị như bên dưới:



Hình 102 Cập nhật node-red trên Raspberry

Sau khi cập nhật node-red ta sử dụng câu lệnh dưới đây để khởi động node-red:

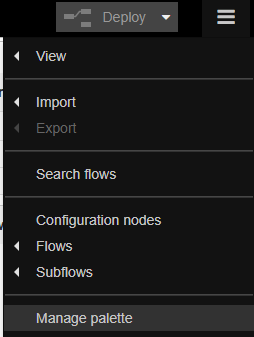
|  |
| --- |
| node-red |

Lưu ý: để node-red tự khởi động khi cấp nguồn cho Raspberry ta sẽ phải thêm node-red vào danh sách phần mềm khởi động cho hệ điều hành của Raspberry. Để thêm ta mở Terminal và nhập dòng lệnh dưới đây:

|  |
| --- |
| sudo systemctl enable node-red.service |

Sau khi khởi động node-red ta sẽ truy cập vào địa chỉ localhost:1880 để sử dụng node-red trên Raspberry. Với giai đoạn này ta cần cài đặt thêm node node-red-contrib-scx-ibmiotapp, node này dùng để kết nối với IBM. Cách cài đặt node này như sau:

Ta nhấn vào menu của node-red và chọn Manage palette



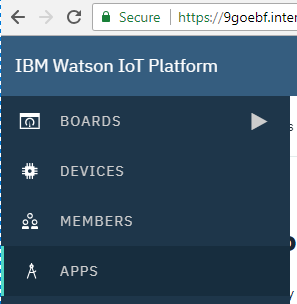
Hình 103 Truy cập vào Manage palette - Quản lý node

Tại mục Palette ta chọn tab Install và nhập tên node node-red-contrib-scx-ibmiotapp vào ô tìm kiếm, sau khi đã tìm thấy node ta cần, ta nhấn vào Install để tiến hành cài đặt node. Như hình là đã cài đặt thành công:



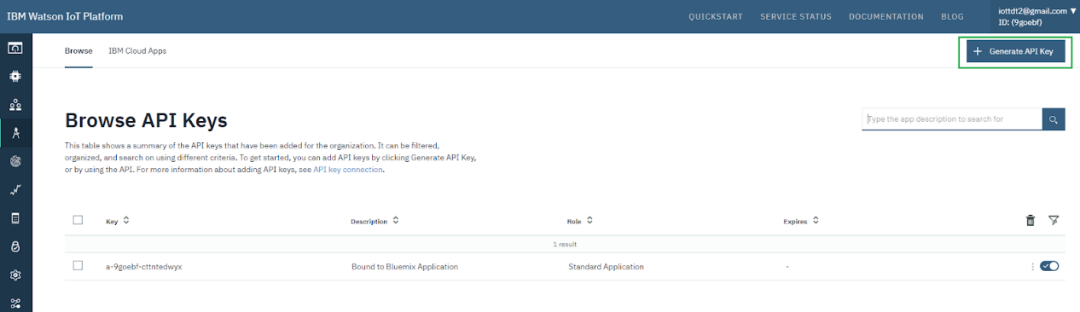
Hình 104 Cài đặt node kết nối IBM cho node-red

Bây giờ ta sẽ tiến hành đăng ký API key trên IBM Watson để truyền và nhận giữ liệu cho giai đoạn 2 này. Cách thực hiện như sau: đầu tiên ta truy cập vào IBM Watson IoT như đã hướng dẫn ở phần tạo tài khoản IBM (Chương 3.2.1), tại giao diện của IBM Watson IoT ta chọn mục Apps từ menu chính như hình dưới:



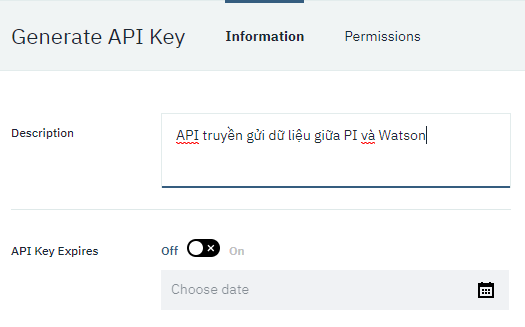
Hình 105 Chọn APPS để tạo khóa API

Tại giao diện tiếp theo ta chọn vào Generate như hình dưới để tạo thêm API key:



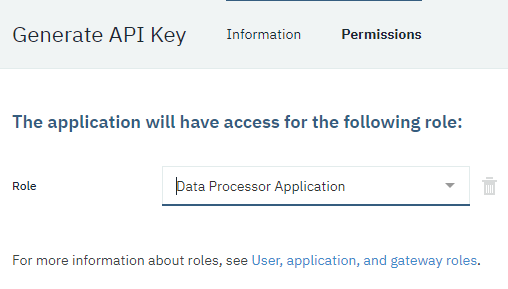
Hình 106 Chọn Generate API Key

Ta điền mô tả về API key này vào phần Description, và tùy chỉnh ngày hết hạn cho khóa (có thể không chỉnh), sau đó nhấn Next để chuyển sang phần tiếp theo.



Hình 107 Mô tả API Key

Tại phần tiếp theo ta chọn Role cho API key này là Standard Application để thực hiện quyền gửi và nhận dữ liệu ( ta có thể chỉnh sửa các quyền này sau).



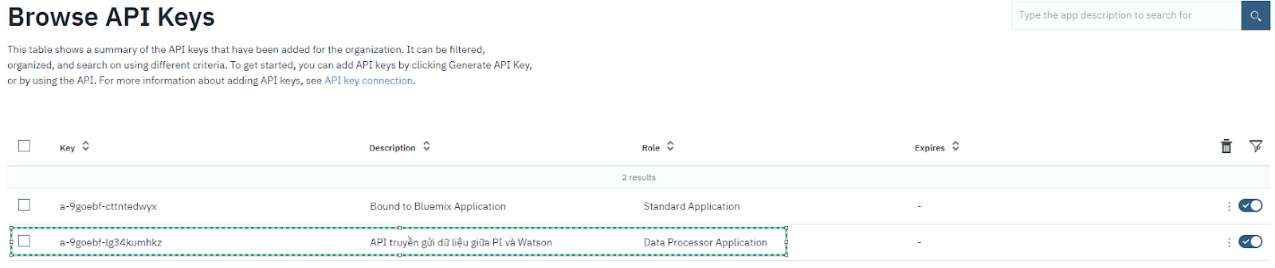
Hình 108 Chọn Role cho API key

Sau đó ta nhấn vào Generate Key để tạo API key ta vừa điền các thông tin trên. Sau khi ta tạo thành công IBM sẽ hiển thị lại thông tin API key ta vừa tạo như sau:



Hình 109 Thông tin API key

Lưu ý: Authentication Token chỉ được cấp 1 lần và không thể thay đổi, nên ta phải lưu lại để sử dụng, nếu mất ta phải thực hiện tạo lại API key khác để sử dụng.



Hình 110 API key ta vừa tạo

Như vậy ta đã hoàn tất tạo API key để kết nối giữa node-red trên Raspberry và IBM Watson. Tiếp theo chúng ta sẽ thiết lập node node-red-contrib-scx-ibmiotapp để tiến hành gửi và nhận dữ liệu.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại Node | Tên Node | Chức năng |
| 1 | Output | Ibmiot (out) | Gửi dữ liệu lên IBM Watson sử dụng API Key |
| 2 | Input | Ibmiot (in) | Nhận dữ liệu từ IBM Watson sử dụng API Key |

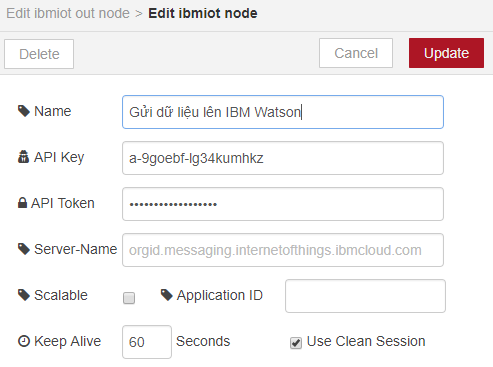
Bảng 10 Danh sách các node sử dụng kết nối với IBM Watson trên node-red

Ta sẽ cấu hình node ibmiot out như bên dưới:



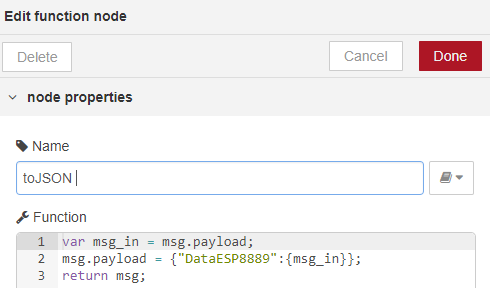
Hình 111 Cấu hình node ibmiot out

Ta chọn Authentication là API Key như hình trên sau đó nhấn vào nút có hình cây bút chì để chuyển về giao diện điền thông tin API Key, ta sẽ tiến hành điền thông tin API Key mà ta đã tạo ở bước trên sau đó nhấn Update như hình dưới:



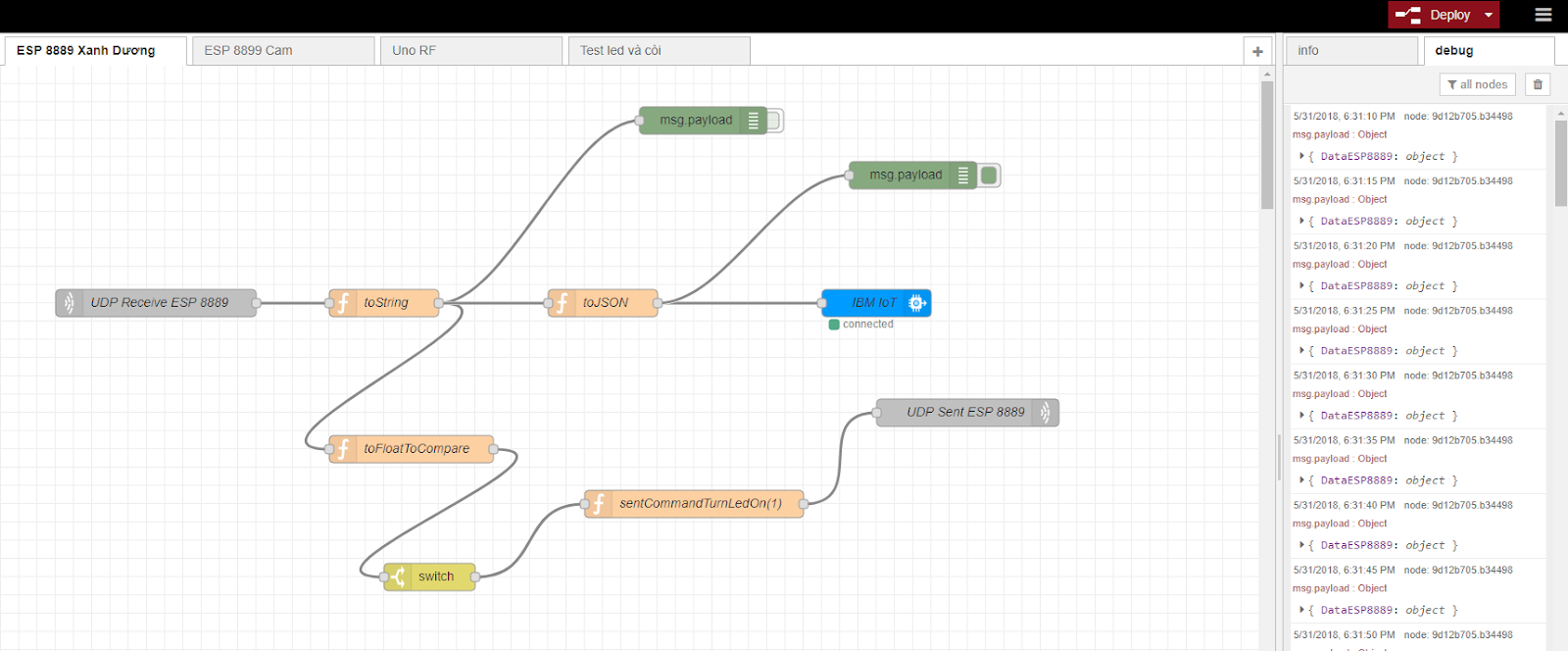
Hình 112 Thiết lập thông số API key trên node-red

Sau khi nhấn Update ta sẽ được chuyển về lại giao diện thiết lập node ibmiot out, ta sẽ điền Device Id đây là Device Id của thiết bị mà ta đã tạo ở bước đăng ký thiết bị trên IBM Watson, Device Type ta điền loại thiết bị ta đã tạo trên IBM Watson, Data ta điền msg.payload. Sau đó nhấn Done để hoàn tất quá trình thiết lập. Như vậy ta đã hoàn tất thiết đặt node ibmiot out, bước tiếp theo ta cần tạo node chuyển kiểu dữ liệu ta cần gửi lên IBM thành dạng JSON. Để chuyển dữ liệu lên IBM ta sẽ tạo node Fucntion để chuyển thành JSON như hình dưới:



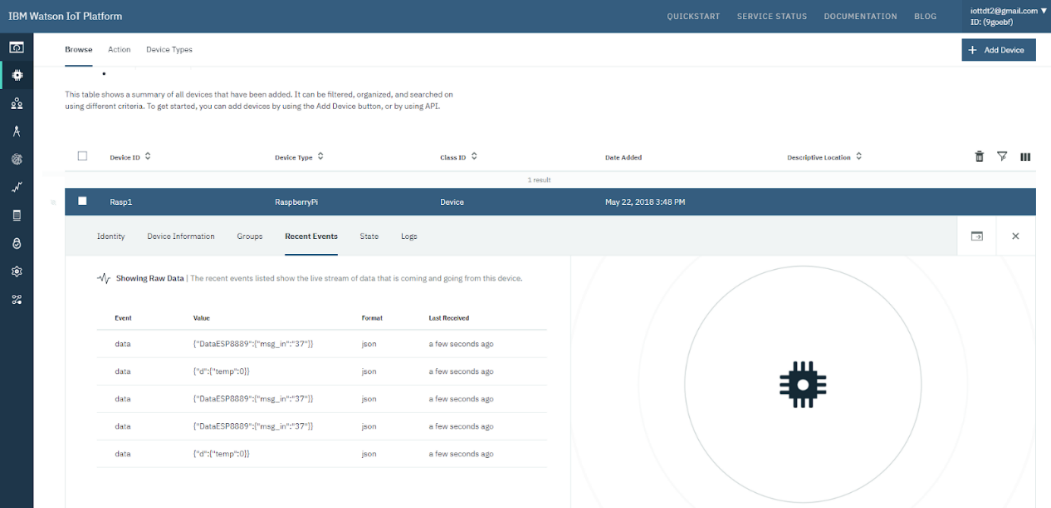
Hình 113 node toJSON

Sau khi đã tạo và thiết đặt các node ta nối các node lại với nhau để có một Flow hoàn chỉnh như bên dưới:



Hình 114 node-red Pi gửi dữ liệu lên IBM Watson

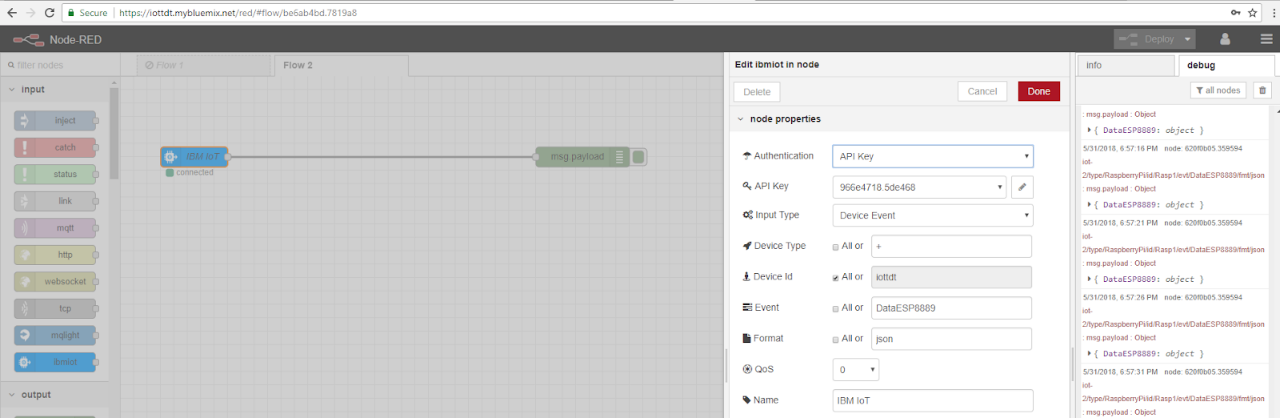
Sau khi nhấn Deploy để gửi dữ liệu ta sẽ vào phần Device Event trên IBM Watson để kiểm tra.



Hình 115 Kiểm tra dữ liệu gửi lên IBM Watson

* + - 1. Nhận dữ liệu

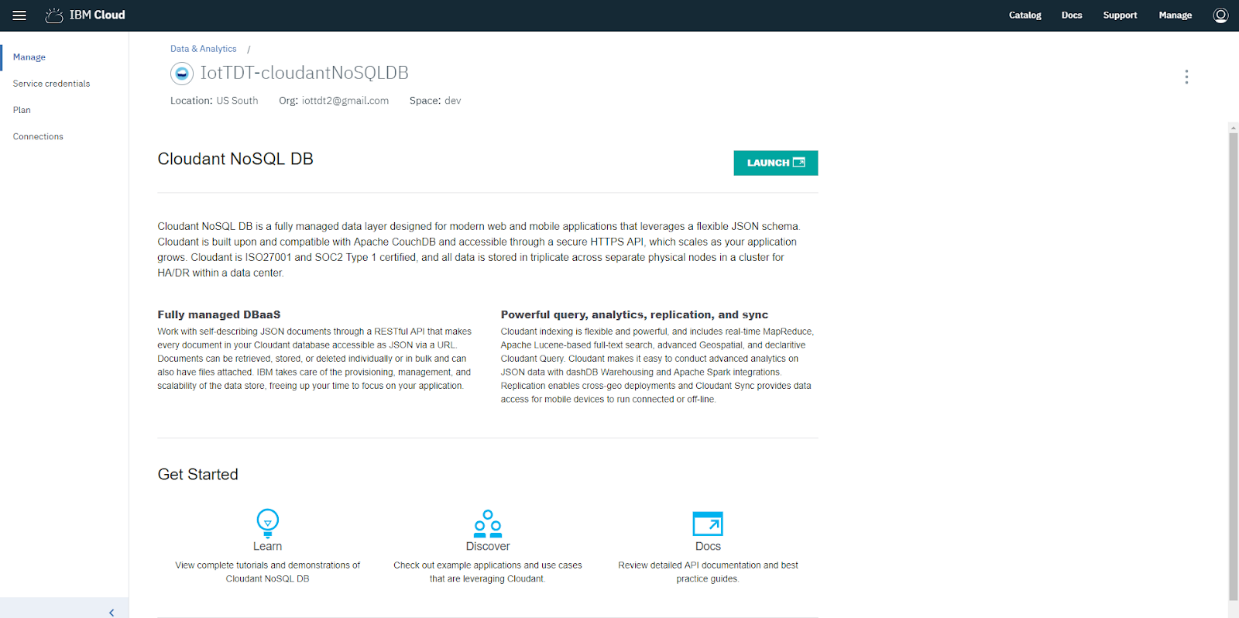
Bây giờ ta sẽ tiến hành giai đoạn 3: Nhận dữ liệu từ IBM Watson vào Node-red trên IBM. Ta sẽ truy cập vào dịch vụ node-red trên IBM và tiến hành đăng nhập bằng tài khoản đã tạo ở bước đăng ký IBM. Quá trình thiết lập gửi dữ liệu ở giai đoạn này sẽ giống y hệt như ở node-red trên Raspberry Pi, tức là giống cài đặt, giống API Key (ta sẽ gửi và đọc đối với cả 2 node-red này cùng 1 luồng dữ liệu). Vì vậy ở đây chúng ta sẽ làm nhận dữ liệu trên node-red IBM (phần ở node-red trên Rasberry Pi sẽ tương tự. Để bắt đầu ta sẽ sử dụng node ibmiot in trên node-red. Cấu hình của node này sẽ giống như cấu hình của node ibmiot out (sử dụng cùng API Key). Cấu hình của node này như bên dưới



Hình 116 Cấu hình node ibmiot in trên Node-red on IBM

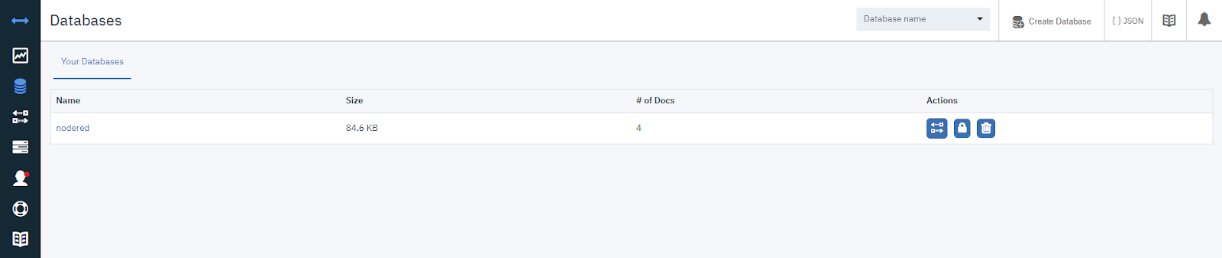
* + 1. Lưu trữ dữ liệu trên Cloudant IBM

Để hoàn thiện đồ án này ta cần lưu trữ dữ liệu đo lại, với tài khoản IBM ta đã đăng ký thì sẽ được miễn phí 1GB lưu trữ dữ liệu tại dịch vụ Cloudant của IBM. Để sử dụng dịch vụ này ta truy cập vào đường dẫn sau <https://console.bluemix.net/services/cloudantnosqldb/> và tiến hành đăng nhập xong ta sẽ được như bên dưới:



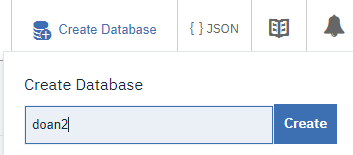
Hình 117 Đăng nhập dịch vụ Cloudant trên IBM

Ta sẽ nhấn vào LAUNCH để truy cập vào dịch vụ DB này, ta sẽ được chuyển tới giao diện như hình dưới:



Hình 118 Giao diện Cloudant

Để tạo DB ta nhấn vào icon DB bên góc trái để mở mục DB ra, sau đó để tạo DB mới ta chọn Create Database để tiến hành:



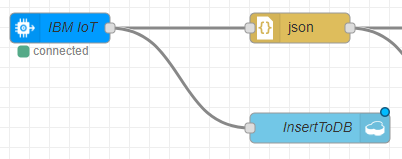
Hình 119 Tạo DB trên Cloudant

Ta điền tên DB và nhấn nút Create để tạo DB (tên DB phải là chữ viết thường và không có ký tự đặc biệt). Sau khi tạo DB xong ta sẽ tiến hành lưu dữ liệu đo đạc vào, ở đây ta sẽ sử dụng node Cloudnat trên node-red trên IBM để lưu dữ liệu vào Cloudant. Ta sẽ cấu hình node này như hình bên dưới:

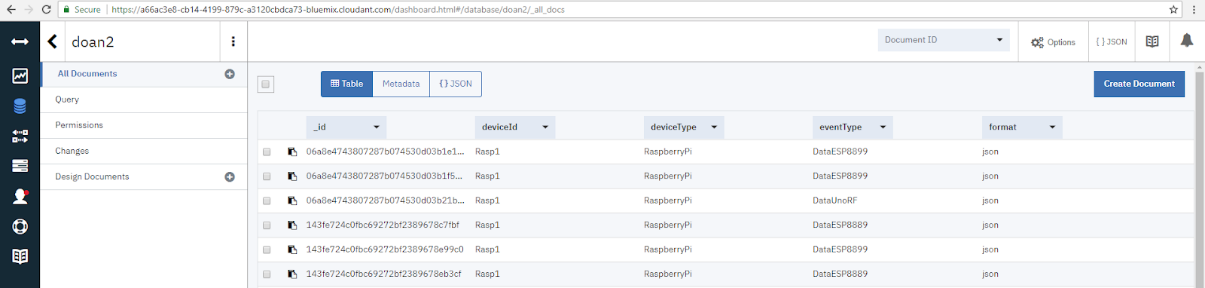


Hình 120 Cấu hình node Cloudant out

Với Service ta chọn đúng tên dịch vụ cloudant ở phần quản lý dịch vụ của IBM, Database ta điền tên DB ta vừa tạo, Operation ta chọn là insert. Sau đó ta tiến hành nối dây giữa node ibmiot in và node cloudant vừa cấu hình là đã xong.



Hình 121 Nối dây để hoàn tất lưu vào Cloudant



Hình 122 Kết quả lưu trữ trên Cloudant

* 1. Các vấn đề gặp phải khi thực hiện dự án
     1. Truyền nhận thông tin qua Wifi với Arduino ESP 8266 MCU

﻿Số ký tự test gửi hiện tại tối đa là 1450 ký tự và tương đương khoảng 1.41KB. Vấn đề ở đây có thể là do tràn bộ nhớ trên Arduino esp 8266 MCU. Trên Arduino esp 8266 MCU cần cấp nguồn đủ nếu không sẻ xảy ra tình trạng không kết nối được wifi hoặc bị reset.

Sử dụng UDP thay cho TCP vì khi chạy TCP node-red (trên raspberry chiếm quyền Server) không thể chuyển quyền được. Việc sử dụng TCP phải thực hiện quá trình bắt tay mỗi lần gửi dữ liệu, vì thế nó gấy tốn thời gian và lãng phí bộ nhớ. UDP không sử dụng bắt tay nên có thể gửi và nhận hầu như không có thời gian trễ, dung lượng gói tin nhỏ hơn nhờ đặc thù kết nối của UDP so với TCP.

* + 1. Truyền nhận thông tin qua RF loại Zigbee UART CC2530+PA V2

Số ký tự test gửi thực tế tối đa khi sử dụng RF là 825 ký tự tương đương với 825 bytes. Với sóng RF tùy vào câu lệnh dài hay ngắn ta cần có khoảng thời gian để câu lệnh gửi hết mới có thể gửi tiếp nếu không tín hiệu thường sẻ bị nhiễu. Vấn dề về khoảng cách cũng vậy với dữ liệu ngắn thường tín hiệu không bị nhiễu nhưng với dữ liệu dài thì khi nhận sẻ bị nhiễu đôi khi còn mất khúc tính hiệu.

Thông số của Zigbee UART CC2530+PA V2 có thông sốkhoảng cách truyền lý tưởng là 1000m nhưng trong thành phố với mật độ nhà dày đặc thì khoảng cách thự tế tối đa tầm 30m (tín hiệu không đồng đều).

Với Zigbee UART CC2530+PA V2 ở chế độ broadcast chúng ta không thể tùy chỉnh chuyển đổi thành router, modem và client. Từ đó ta không có trung gian phân luồn nên khi có một con thứ 3 chen vào tín hiệu sẽ bị chèn nhiễu vào nhưng thường thì sẻ không ảnh hường tới gói tin nhận ta có thể tách phần nhiễu đó.

Khi thực hiện lệnh trên IBM gửi lệnh xuống có một nhược điểm đó là bị nhiễu thông tin khi truyển, trùng lặp lệnh

* + 1. Lý do sử dụng Raspberry pi

Sử dụng Raspberry pi làm trung tâm xử lý thông tin. Trên Raspberry ta có hệ điều hành vì thế ta dễ dàng điều khiển cũng như thay đổi hơn, với cấu hình cao hơn ta có thể xử lý với chuỗi câu lệnh dài và phức tạp hơn. Đây cũng như là một trạm xử lý nhỏ local.

Raspberry cũng đóng vai trò như một router kết nối với nhiều phương thước kết nối internet wifi, cổng ethernet và có thể cắm usb 3G/4G giúp duy trì và đảm bảo luôn kết nối thay cho một con router cao cấp để làm việc tương tự. Nếu sử dụng từng con Arduino để gửi thì khi đường truyền mạng có vấn đề ta phải config và nạp lại code cho từng con một rất tốn thời gian và công sức so với raspberry.

* 1. Xây dựng hệ thống thực nghiệm

Để dễ quản lý cung như tao tác khi xây dựng hệ thống trên Node-red sẻ tạo ra nhiều Flow khác nhau. Từ các Flow ta sẻ quản lý từng con cảm biến kết nối đến hoặc một công việc nào đó. Như trong hệ thống thực nghiệm này sẻ có 4 Flow: 3 Flow quản lý 3 con Arduino độc lập và 1 Flow chuyên xử lý gửi nhận dữ liệu với IBM.

**Các Flow trên Raspberry Pi có các nhiệm vụ được liệt kê dưới đây:**

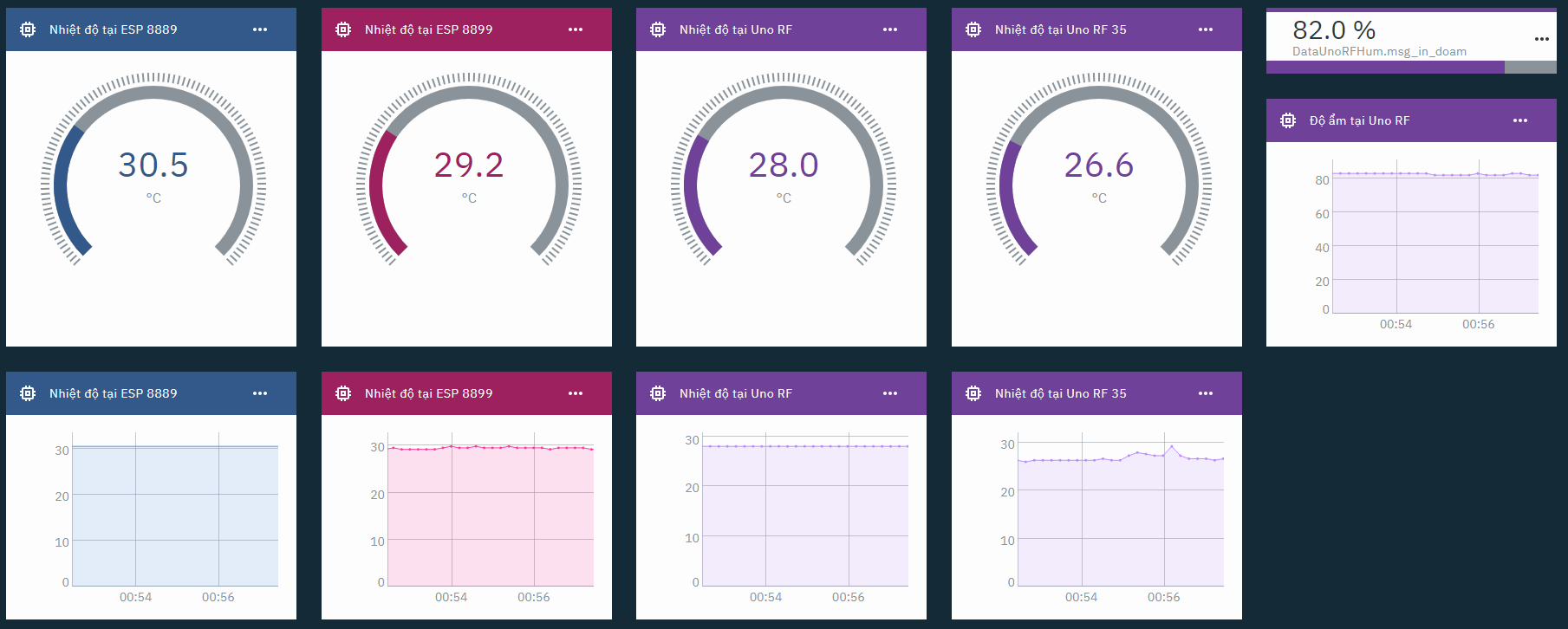
1. Xử dụng để nhận dữ liệu Wifi được đọc lên từ Arduino esp 8266 MCU (sử dụng cảm biến LM35) thứ nhất, đưa ra cảm báo xuống Arduino esp 8266 MCU thứ nhất khi có nhiệt độ cao. Đồng thời hiển thị ra giao diện và gửi dữ liệu lên IBM.
2. Xử dụng để nhận dữ liệu Wifi được đọc lên từ Arduino esp 8266 MCU (sử dụng cảm biến LM35) thứ hai, đưa ra cảm báo xuống Arduino esp 8266 MCU thứ hai khi có nhiệt độ cao. Đồng thời hiển thị ra giao diện và gửi dữ liệu lên IBM.
3. Xử dụng để nhận dữ liệu RF được đọc lên từ Arduino Uno (Sử dụng cảm biến DHT11 và LM35), đưa ra cảm báo xuống Arduino Uno khi có nhiệt độ cao. Đồng thời hiển thị ra giao diện và gửi dữ liệu lên IBM.
4. Xử lý cung cấp giao diện và nhận lệnh từ IBM xuống dùng để bật tắt đèn led, thông báo đèn và còi, sau đó thực hiện gửi các lệnh xuống các con arduino nó đang quản lý

**Các Flow trên IBM có các nhiệm vụ được liệt kê dưới đây:**

1. Xử dụng để nhận dữ liệu trên Raspberry đọc lên từ Arduino esp 8266 MCU (sử dụng cảm biến LM35) thứ nhất. Đồng thời lưu trữ dữ liệu lên Cloudant IBM.
2. Xử dụng để nhận dữ liệu trên Raspberry đọc lên từ Arduino esp 8266 MCU (sử dụng cảm biến LM35) thứ hai. Đồng thời lưu trữ dữ liệu lên Cloudant IBM.
3. Xử dụng để nhận dữ liệu trên Raspberry đọc lên từ Arduino Uno (Sử dụng cảm biến DHT11 và LM35. Đồng thời lưu trữ dữ liệu lên Cloudant IBM.
4. Xử lý cung cấp giao diện và nhận lệnh từ Raspberry Pi lên dùng để bật tắt đèn led, thông báo đèn và còi, sau đó thực hiện gửi các lệnh xuống các con Raspberry Pi

Các Flow sau khi cài đặt sẻ có các node như hình bên dưới, để có được các Flow đó bạn có thể lấy từ Clipboard ở CD đính kèm

Sau khi thêm tất cả ta có thể tìm thấy các luồn dữ liệu đi lên IBM và sau đó cho chúng hiện thị lên Dashboard IBM Watson IoT Platform xem nhanh như hình dưới đây

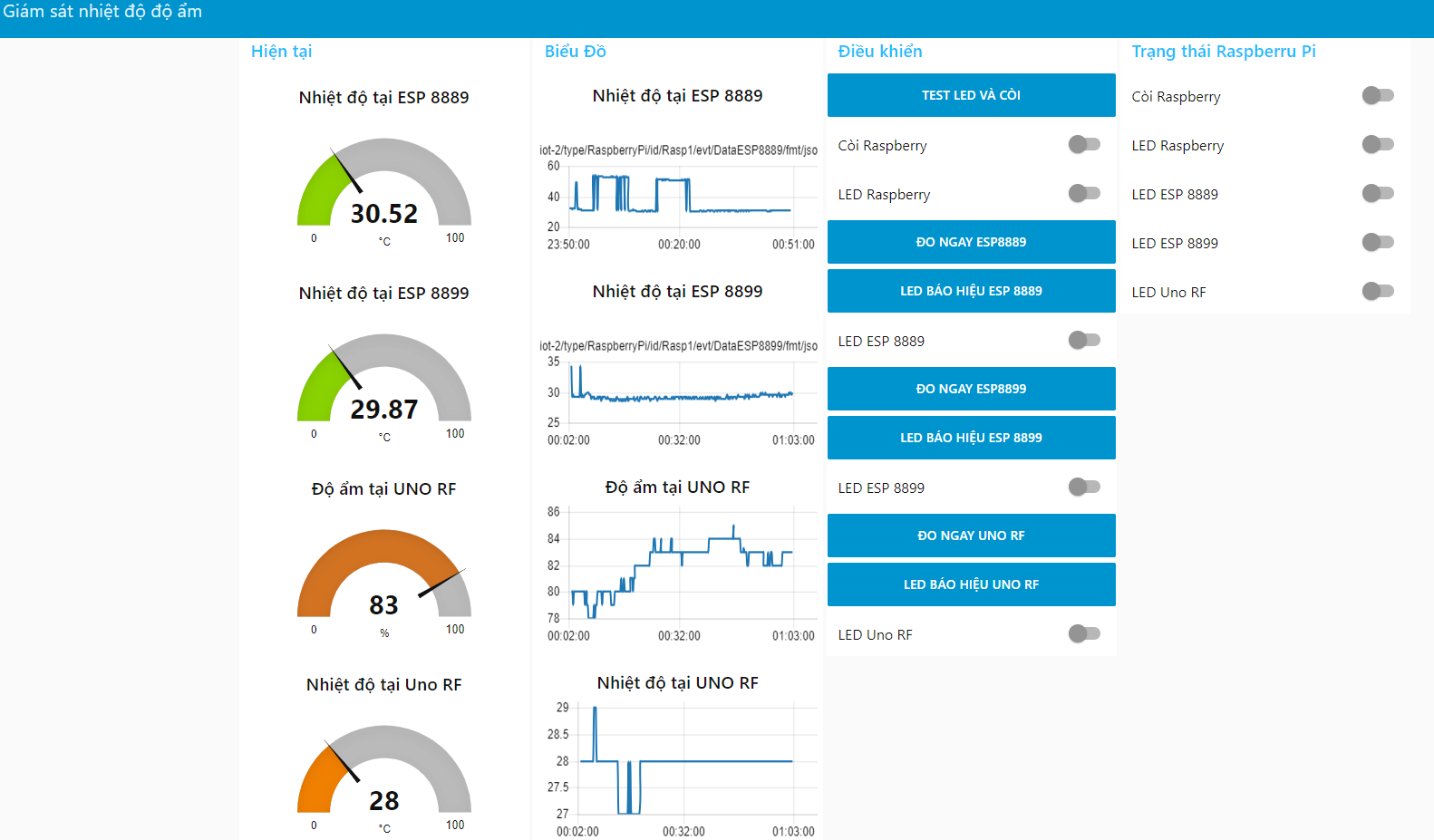


Hình 123 Giao diện hiển thị dữ liệu nhận đươc của IBM cung cấp

Bên cạnh đó ta cũng có giao diện để điều khiển trên IBM lẫn Raspberry Pi như dưới đây



Hình 124 Màn hình điều khiển cũng như theo dõi của Node-red trên Raspberry Pi



Hình 125 Màn hình điều khiển cũng như theo dõi của Node-red trên IBM

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. Đinh Hồng Thái, Giới thiệu vắn tắt kít phát triển ESP8266 (chíp nạp CP2102), http://arduino.vn/bai-viet/1141-gioi-thieu-van-tat-kit-phat-trien-esp8266-chip-nap-cp2102
2. Nguyễn Ngọc Cường, Phạm Hồng Bình, “Tự động hóa hệ thống vườn nhà kính”, Khóa luận/đồ án tốt nghiệp 2017, Đại học Tôn Đức Thắng – Khoa Công Nghệ Thông Tin
3. Trần Anh Tuấn, Phan Tuấn Trọng, “Xây dựng hệ thống giám sát nhiệt độ, độ ẩm tại từng phòng của tòa nhà, ứng dụng IoT với raspberry pi”, đồ án 1 2017, Đại học Tôn Đức Thắng – Khoa Công Nghệ Thông Tin
4. Hshop, Mạch Thu Phát RF Zigbee UART CC2530+PA V2, http://hshop.vn/products/mach-thu-phat-rf-zigbee-uart-cc2530-v1-1

**Tiếng Anh**

1. [Patrick Catanzariti](https://www.sitepoint.com/author/pcatanzariti/), Connecting a Raspberry Pi to IBM Watson, Bluemix and Node-RED, <https://www.sitepoint.com/connecting-a-raspberry-pi-to-ibm-watson-and-bluemix/>
2. Karen E Lewis, Chris Anderson, Demo of how to use IBM Watson IoT and Node-RED to connect Raspberry Pis to control devices, <https://www.youtube.com/watch?v=0rPA1w6M2Q0>
3. Vivek Gite, How To Check and Use Serial Ports Under Linux, <https://www.cyberciti.biz/faq/find-out-linux-serial-ports-with-setserial/>
4. [Dejan Nedelkovski](https://howtomechatronics.com/author/howtom12_wp/), Arduino Wireless Communication – NRF24L01 Tutorial, https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/
5. Ivan Grokhotkov, UDP communication between ESP8266 and an external client, UDP, http://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/udp-examples.html#udp
6. [Harsh Mangukiya](https://create.arduino.cc/projecthub/harshmangukiya), Serial Communication Between Two Arduino Boards, https://create.arduino.cc/projecthub/harshmangukiya/serial-communication-between-two-arduino-boards-d423e9
7. [Rahul Iyer](http://www.deviceplus.com/author/rahul/), Arduino Communication Protocols Tutorial, http://www.deviceplus.com/how-tos/arduino-guide/arduino-communication-protocols-tutorial/
8. [Kevin Darrah](https://www.youtube.com/channel/UC42d7zFnWU0dYVk_M0JED6w), 2 Arduinos Talking How-To Hardware Serial, https://www.youtube.com/watch?v=JINRMVFeoy4
9. [Marcelstoer](https://github.com/nodemcu/nodemcu-firmware/commits?author=marcelstoer), Building the firmware for NodeMCU, https://nodemcu.readthedocs.io/en/master/en/build/
10. [Marcelstoer](https://github.com/nodemcu/nodemcu-firmware/commits?author=marcelstoer), Flashing the firmware, https://nodemcu.readthedocs.io/en/latest/en/flash/