ESP8266 - OTA

and a bit more ....

Outline

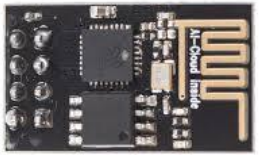
- ❏ BRIEF INTRO ESP8266
- ❏ PROGRAMMING BY SERIAL
- ❏ PROGRAMMING OVER-THE-AIR (OTA)
- ❏ NODE-RED
- ❏ MQTT

Demo Objectives

- What’s an ESP8266?
- How to program an ESP8266?
- What is OTA, why do it, how you can do it?
- What is NodeRed and what can it do?
- What is MQTT and why use it?

ESP8266

- Microcontroller - 17 GPIO pins - supports SPI, I2C, I2S, PWM,10-bit ADC
- 32-bit Tensilica RISC microprocessor - 64 KiB instruction / 96 KiB data
- Usually packaged as a module with up to 16 MiB (4 MiB common) flash
- Can be programmed in many languages - LUA, C, Javascript, Python
- Built in WiFi
- Operates at 3.3 VDC
- CHEAP - $2.50 - 8.00



Programming by serial interface

- Send program to chip
- IDE to program with - ArduinoIDE, PlatformIO,ESPlorer, Mongoose, ...
- USB to Serial Adapter

ArduinoIDE for Serial Programming

1. Install ArduinoIDE at the 1.7 level or later (from Arduino website).
2. Start ArduinoIDE and open Preferences window
3. Enter https://github.com/esp8266/Arduino/releases/tag/2.4.0-rc2 into Additional Board Manager URLs field. If reqd. you can add multiple URLs, separating them with commas.
4. Open ‘Boards Manager’ from Tools > Board menu and install *esp8266* platform
5. Plug-in your development board - and select comm port
6. Select your ESP8266 board from Tools > Board menu and upload

Over The Air - OTA Updating - Why?

> access to the devices

> faster - many devices deployed

> no USB / serial interface

OTA Sequence

1. put code that can receive new code in firmware
2. device makes a request for new firmware
3. device downloads new firmware
4. device reboots to new firmware
5. device deletes old firmware

* need to keep program smaller b/c OTA bootloader + 2 programs must fit in memory

Options for Triggering OTA Update

- number of cycles
- power on
- scheduled time - once a month
- response to GPIO input
- remotely triggered

OTA Update Options

> ArduinoIDE - https://github.com/esp8266/Arduino/tree/master/doc/ota_updates

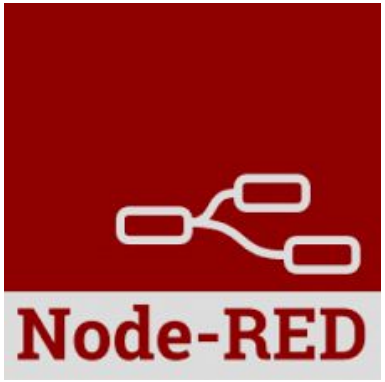
> https://iotappstory.com/

> https://mongoose-os.com/

> Webserver (PHP/Flask)

> Node-Red

Node-Red



> included with Raspbian

> visual programming language based on NodeJS

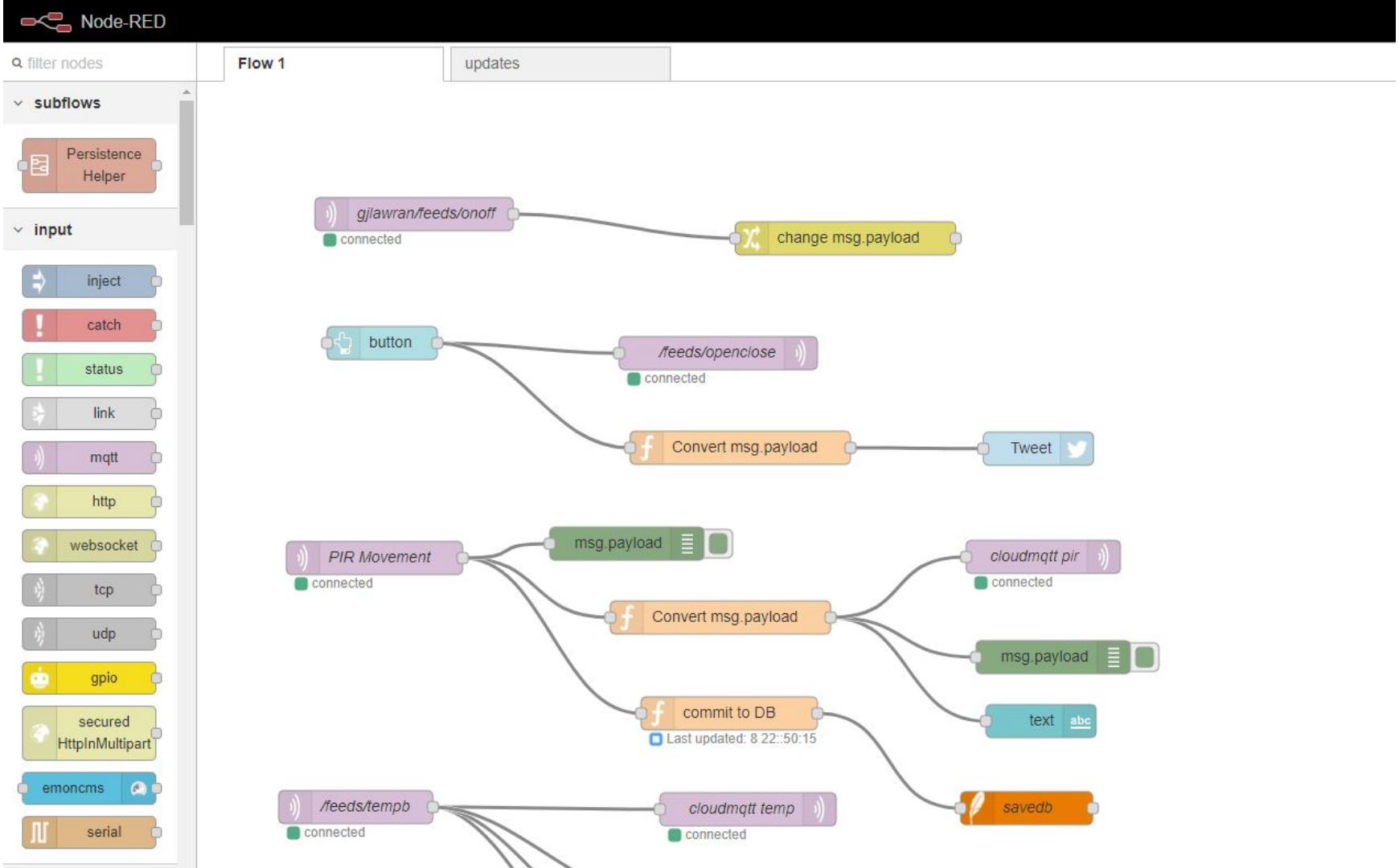
> makes it very easy to reuse code - forms for configuration

> provides a very easy way to manage flow of data from IOT devices

> provides a simple web interface for building dashboard

> processing / interfaces with services that are hard/in on lightweight IOT device

> your own adafruit.io and IFTTT in a box


Node-RED
filter nodes
Flow 1
updates
subflows
Persistence Helper
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
gpio
secured HttpInMultipart
emoncms
serial
gjlawran/feeds/onoff
connected
change msg.payload
button
/feeds/openclose
connected
Convert msg.payload
Tweet
PIR Movement
connected
msg.payload
cloudmqtt pir
connected
Convert msg.payload
msg.payload
commit to DB
Last updated: 8 22::50:15
text
savedb
/feeds/tempb
connected
cloudmqtt temp
connected

Garage Door

BUTTON

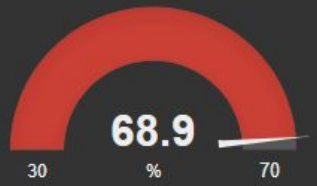
Motion Detectors

Movement in garage - Sat Dec 09 2017 04:18:19 GMT+0000 (UTC)

Temperature

Temperature

Humidity



OTA Update Demo with NodeRed

1. Add OTA Update Flow to NodeRed
   https://github.com/gjlawran/node-red-contrib-fs-ops/
2. Using OTA Update Sketch template compile and export sketch from ArduinoIDE
   https://gist.github.com/gjlawran/0488d55901a89a45d2460bacad02fb2e
3. Upload firmware from sketch directory to NodeRed
   http://dietpi.local/esp8266-ota/update
4. Link the firmware with the MAC address of the modules you would like to update - use it from the NodeRed upload page
5. Reboot the ESP … or otherwise trigger it to update

MQTT

> a protocol for interchanging data between IOT devices

> publication / subscription by channel

> different levels of certainty for message receipt

> efficient

> direct or via broker

> brokers for free trial (https://github.com/mqtt/mqtt.github.io/wiki/public_brokers)

> Mosquitto (https://mosquitto.org/)