Using Raspberry Pi & Node-Red to Open/Close a Garage Door

Pete Keefe

February 2019

Raspberry Pi

- Single Board Computer
  - Credit card size or smaller
  - Developed in UK mainly for educational purposes
  - First model released in Feb 2012, lately updates released on Pi Day (3.14)
  - Generally run Linux operating system – Raspbian
  - Current models:
    - Raspberry Pi 3+ B – $35 for 4 core 1.4Ghz, 1 GB mem, 4x USB, connections for Ethernet, WiFi, BLE, camera, audio/video, HDMI
    - Raspberry Pi Zero W - $10 for 1 Ghz, 1GB mem, connections for WiFi, BLE, 1 micro USB, mini HDMI

Node-Red

- Web browser-based flow editor
- Developed by IBM, became open-source in 2016
- Internally written in JavaScript (runs under Google's JavaScript engine used in browsers) programs with JSON data files
- A Message (MSG) object passes from one node to another ("readable" JSON object). MSG fields:
  - Payload [this is generally the field that is important]
  - Topic [usually used for MQTT topics]
  - {user can add other fields as desired}
- Flows or part flow are Exported or Imported as text file
  - Makes for very easy code sharing
- Add-on nodes or pallets also distributed as readable JavaScript code

Setup/Update Node-Red on Pi

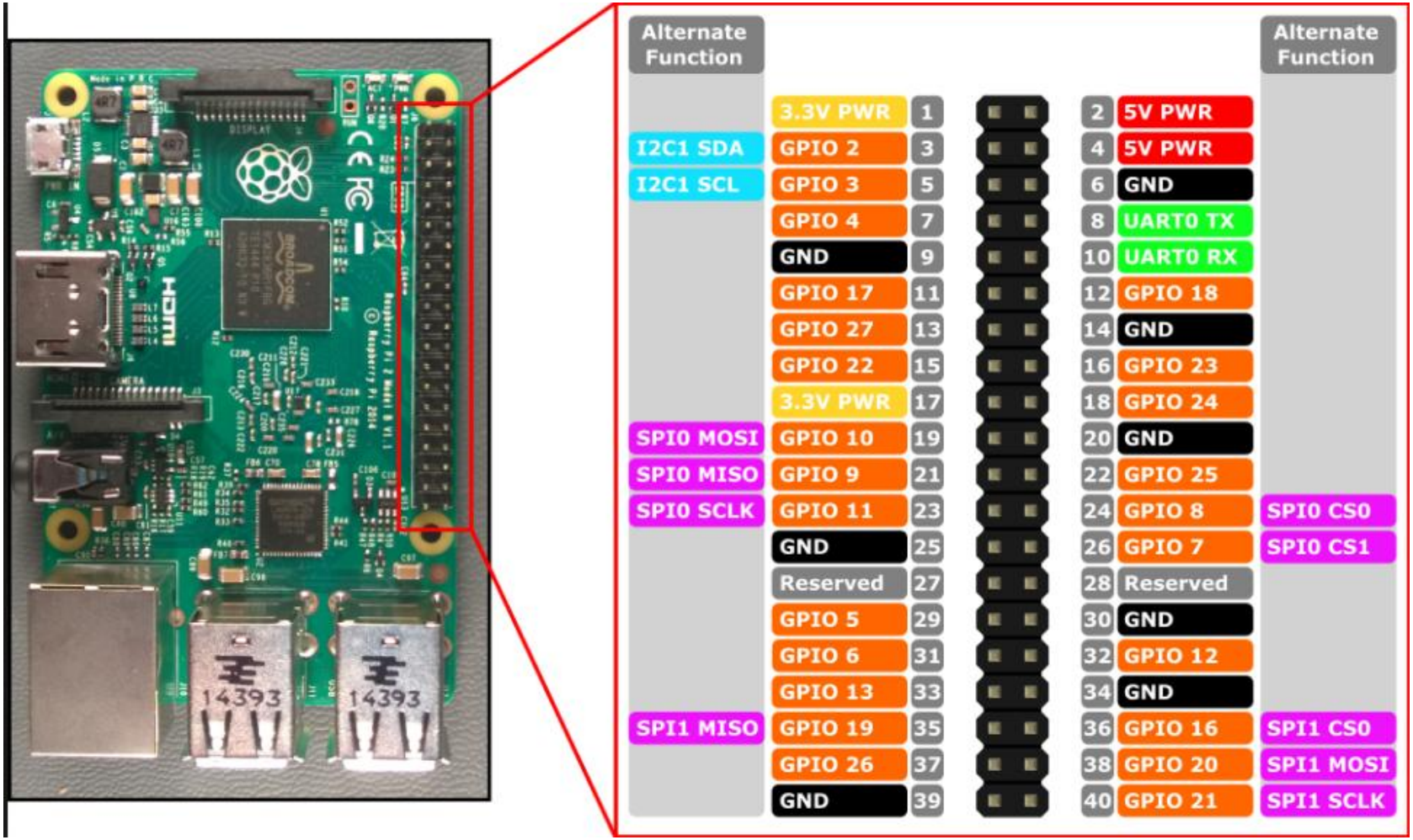
- See Node-Red on Raspberry Pi and run

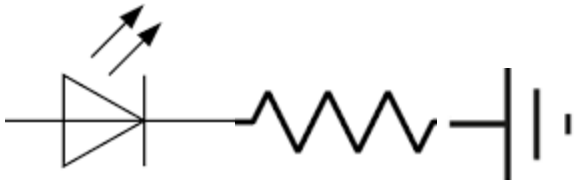
  bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
- Above will update to latest version plus add Pi specific nodes (being used today)
- Above web page also shows how to setup node-red to start automatically at system start
- Manually start in cmd window “node-red-start”
- Access via web at:
  - Design window: http://{ip address of Pi}:1880
  - Dashboard: http://{ip address of Pi}:1880/ui

First steps

- On test Pi I already added *node-red-dashboard* & *node-red-node-pisrf* pallets
- Review pallets nodes
- Flow
  - A way to separate “code”
  - can use links between flows to send/receive messages
- First flow: Inject + Debug
- Export, Delete All, Import
- Dashboard
  - Tabs, Groups, Order
  - Switch + Debug
  - Red, Green, Blue lights
  - All Off button

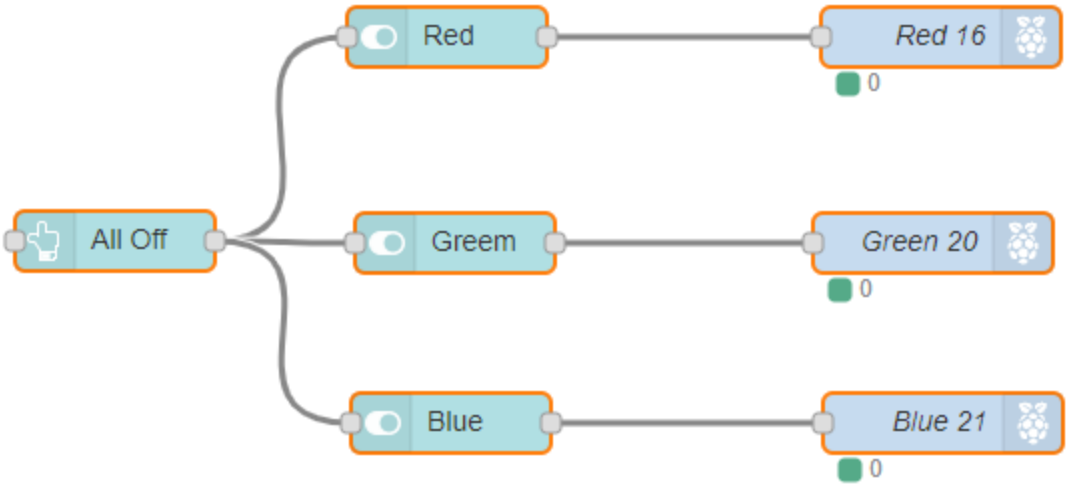
Connecting Raspberry Pi to Outside World - pin # vs GPIO



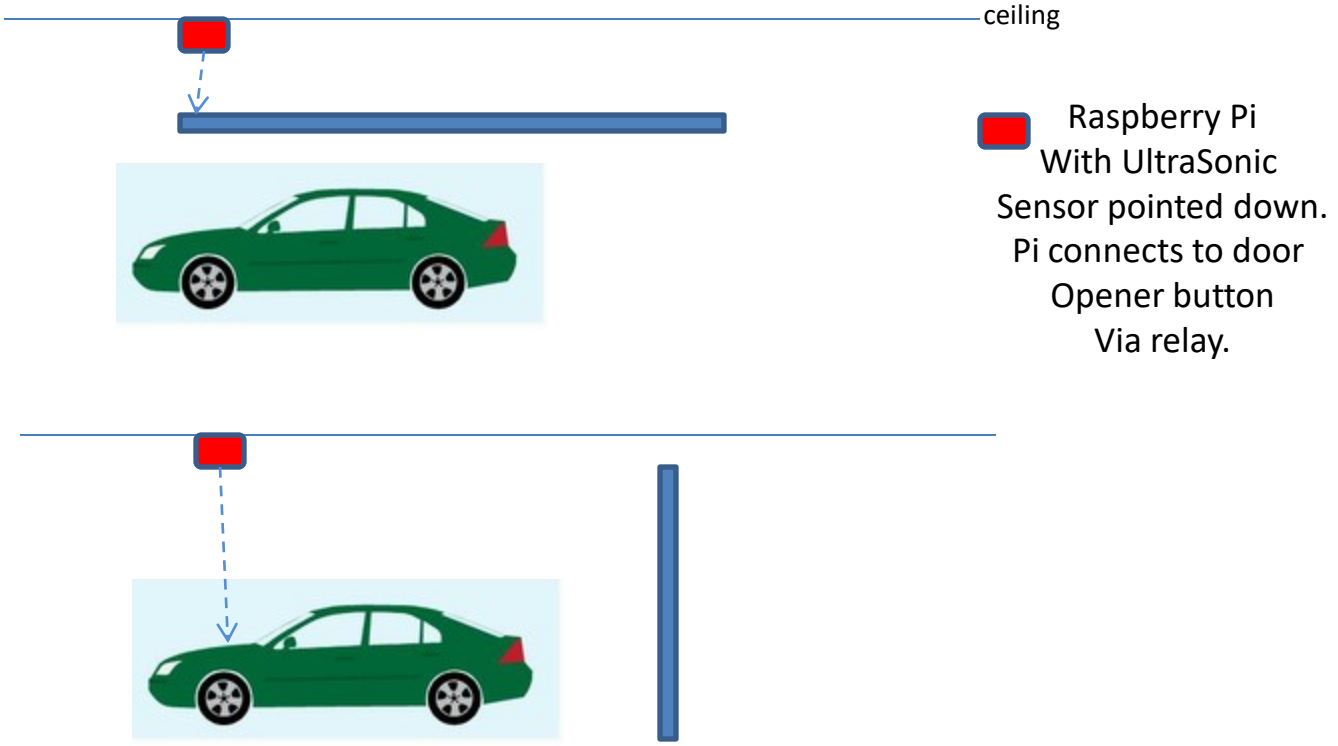


Red = 16, Green = 20, Blue = 21, Ground = 6

Demo Flow



Garage Door



UltraSonic sensor gives distance to object, car present and door down, door up.

UltraSonic

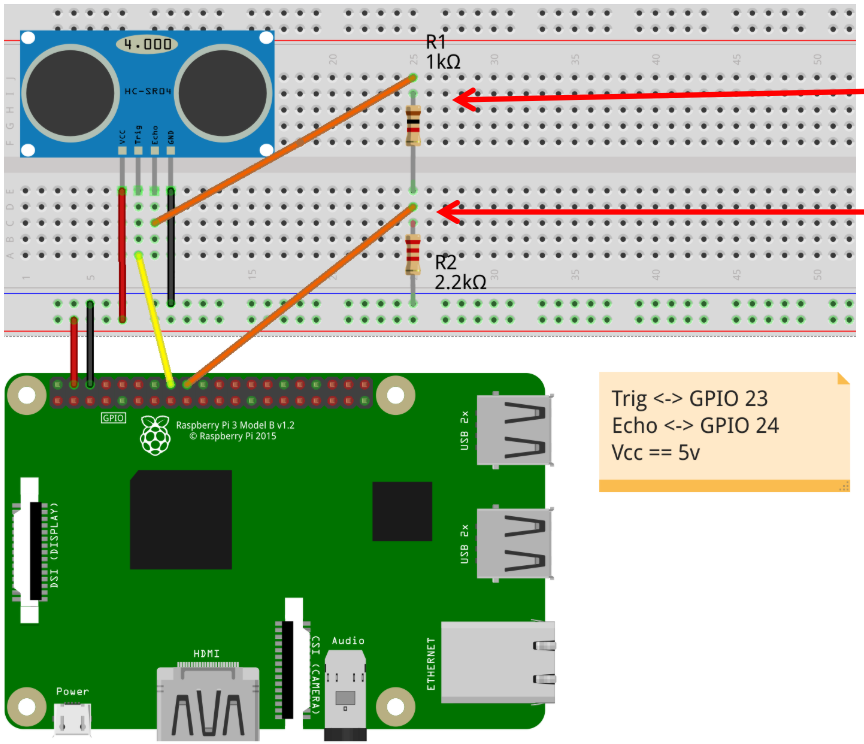
- Pallet: node-red-node-pisrf
  - Requires two GPIO pin numbers, the trigger pin and the echo pin
  - Measures in centimeters (limit 5 meters)
  - Produces one measure every 0.5s (by default) - but only if the distance is different from the previous reading.

HC-SR04 Specifications

- Working Voltage: DC 5V
- Working Current: 15mA
- Working Frequency: 40Hz
- Max Range: 4m
- Min Range: 2cm
- Measuring Angle: 15 degree
- Trigger Input Signal: 10µS TTL pulse
- Echo Output Signal Input TTL lever signal and the range in proportion
- Dimension 45 * 20 * 15mm

Protect Your Raspberry Pi

- HC-SR04 requires 5v for power (newer version can use 3.3v) but 5v is too much for Raspberry Pi (can burn out port / computer). Use voltage divider to reduce signal line.



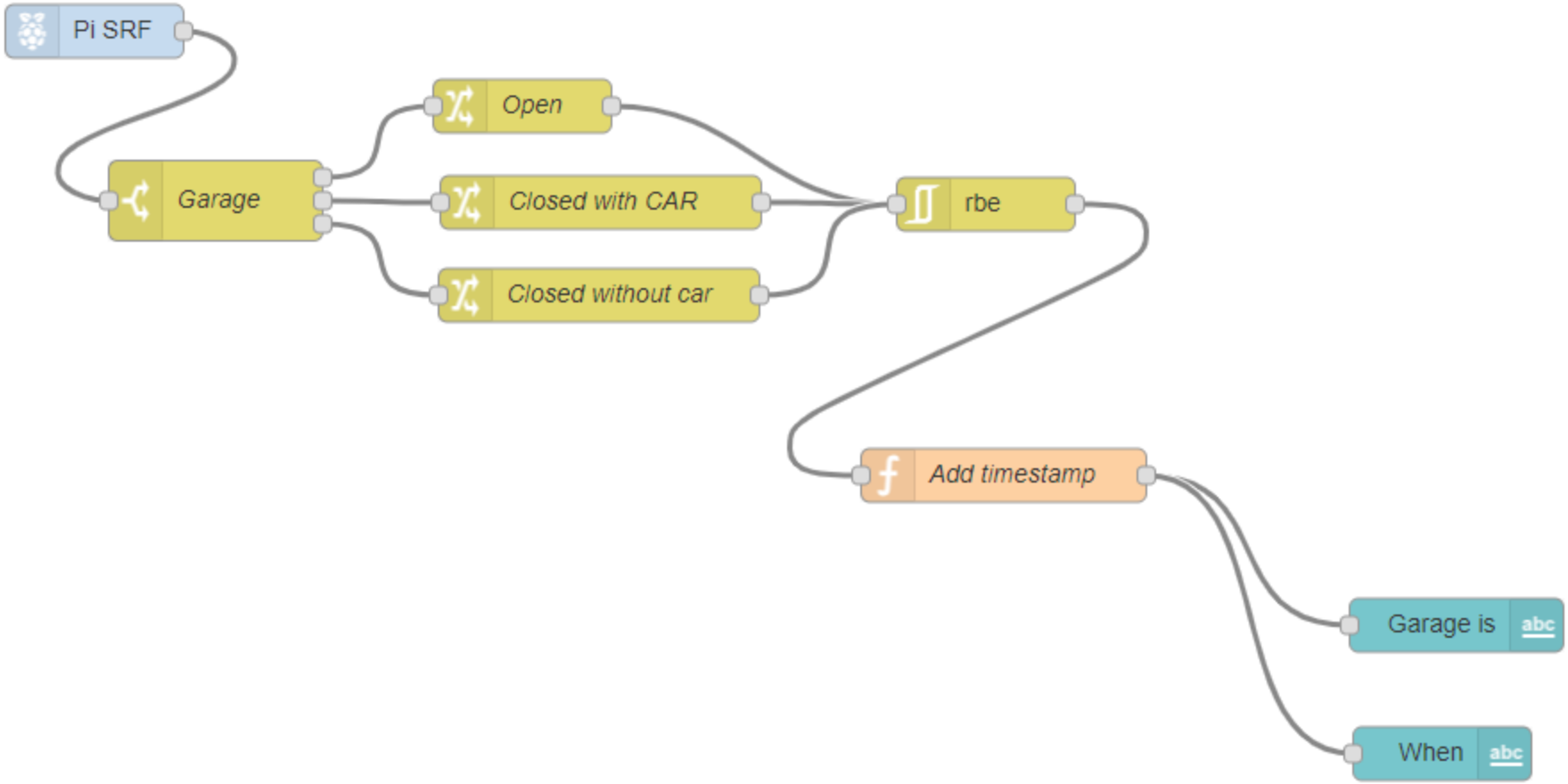
5v

$3.3v = \frac{5v * 2.2K}{1K+2.2K}$

Echo Output Signal to Pi

Trig <-> GPIO 23
Echo <-> GPIO 24
Vcc == 5v

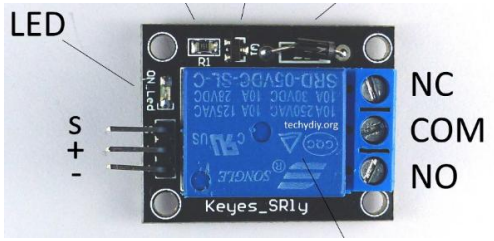
Demo Flow



Thanks to work done by John Scott!

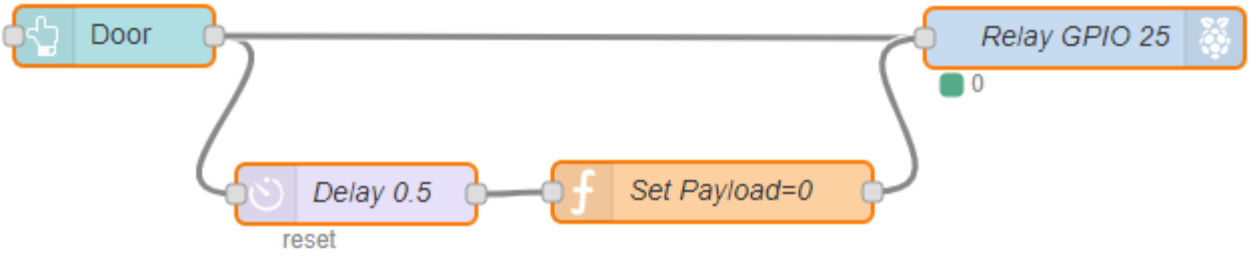
Relay to Activate Door Opener

- Use relay to isolate Pi from Opener
- Connect Normally Open connections in parallel with the current opener button
- Connect signal pin to spare gpio – send a 1 to “push button”, wait 0.5 seconds and revert to 0 (don’t want to hold down button too long)



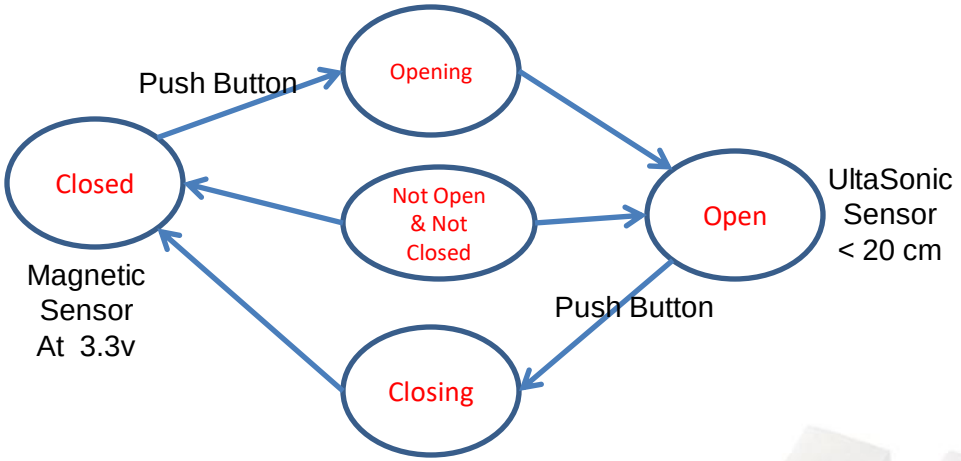
+ = 5V, - = Ground, S = signal (GPIO)
NO = Normally open – connect to opener

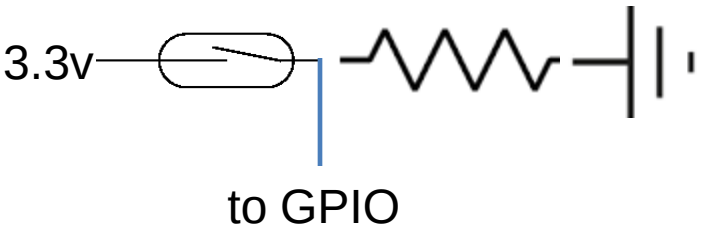
Demo Flow



Wait, wait there's more

- What if door is partially open or closed?
  - Possible solution: put magnetic sensor to detect when door fully closed. State diagram now:





Magnetic Sensor

Flow for Door Sensor

Output Link
to Another Flow

global.set('sensor',msg.payload+1)

Setting global = 2 if closed, 1 if open
var x = global.get("sensor")
(global.get thinks null if set = 0)

Alexa

- Pallet: node-red-contrib-wemo-emulator
  - Allows for Alexa to discover defined nodes as wemo lights
  - Limited to turn on/off, no special response back to Alexa
- Pallet: node-red-contrib-alexa
  - Setup AWS Skill, secure nod-red for https that AWS calls, port forwarding in your router
  - Will allow programming of requests and responses
  - No auto discovery
- Pallet: node-red-contrib-alexa-remote2
  - Can easily send announcements to Echo
  - Still learning about other capabilities
- Also might want:
  - Tell Alexa to open or close door
  - Announce via Alexa when door opened
  - Ask Alexa for status of door