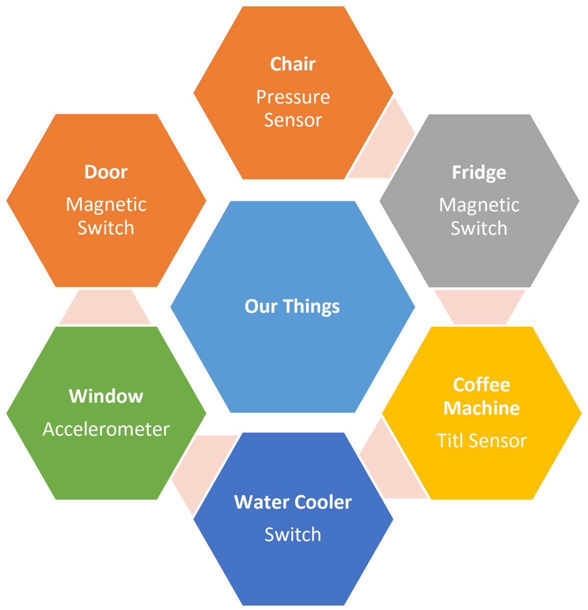
Small-scale simulator of an Internet of Things (IoT)

S.Arora, C.Baber, J.S.W.Cervantes, J.S.Morar
Electrical, Electronics and Systems Engineering, University of Birmingham, UK
sid.irk7@gmail.com

Motivation

- The IoT is a collection of objects embedded with sensors and firmware, communicating to each other over a network.
- Can we simulate an IoT in our own lab?
  - Use available 'things'
  - Embed with sensors and software.
  - Design a communication protocol.
  - Integrate the things into an IoT
- Research: Monitor the data flow in the IoT, analyse the data for patterns, use the patterns to make the network smart (machine learning)



Requirements

Hardware

- Sparkfun ESP8266 Thing (WiFi module)
- Intel Edison
- Arduino Uno
- Sensors:
  - Magnetic switch
  - Accelerometer
  - Tilt sensor
  - Pressure sensor

Software

- Node-RED
- Arduino IDE

Our Nodes

The **chair** is connected with a force sensitive resistor (sensor) at the bottom. The resistance of the sensor is proportional to the force applied on it. The resistance is communicated to the Edison by the ESP8266 through its analog pin. When a person sits on the chair, the force increases, which in turn increases the analog pin value.

The **coffee machine** is equipped with a tilt sensor on its handle as the handle of the machine is tilted 90^0 on opening or closing. The tilt sensor acts like a switch and connects the digital pin of the ESP8266 to the ground. When the machine is used (opened or closed), the sensor is tilted 90^0 and the pin value changes (1 or 0) and hence the change is noted.

The **door** and the **fridge** are equipped with a magnetic switch each. The switch connects the digital pin of the ESP8266 to the ground. When the distance between the two ends of the magnetic switch is < 10mm, the switch is closed and the digital pin = 0. When the door is opened, digital pin = 1. The value of the digital pin is communicated.

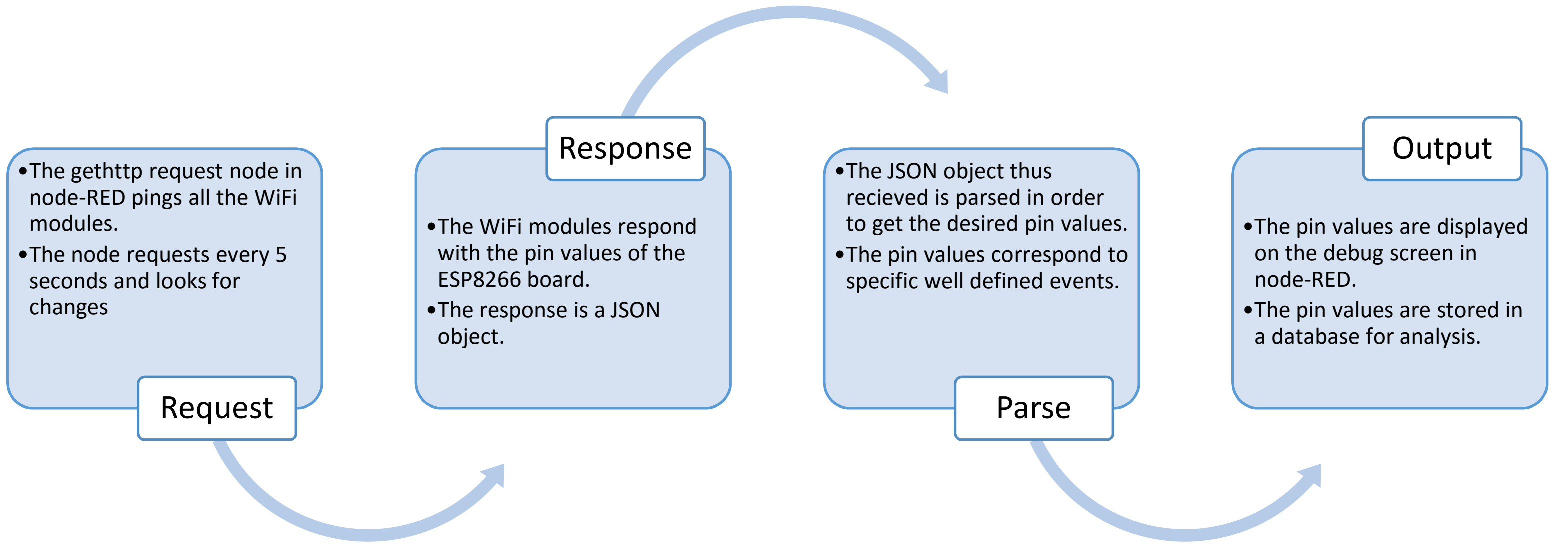
The **window** is equipped with an accelerometer. When the window is opened/closed, the corresponding change in the accelerometer (x, y and z axis) is observed and recorded. These changes are communicated to the Edison through the analog pin values of the ESP8266.

The **water cooler** is equipped with a general switch at its knob. When the cooler is used, the knob is pressed which breaks the connection between the digital pin of the ESP8266 and the ground and hence changes its value. The change is communicated.

Note

The experimental setup is in accordance with the physical objects present in our lab. The sensors have been selected according to our needs. The experimental conditions should not be generalized or standardized. The methodologies though can be applied to different experimental conditions and even on larger networks.

Implementation



Communication

- The ESP8266 WiFi modules and the Intel Edison are connected to the same wireless network.
- The WiFi modules are configured to connect to the wireless network and respond to any requests from devices present on the network (web server).
- The requests to the modules are made by the Intel Edison on a regular interval.
- The sensors present at each node are connected to the analog/digital pins of the module.
- The response contains the values of the analog/digital pins.
- These values are passed on to node-RED running on the Edison.

Pattern recognition

- An event is a change in the pin values of the nodes. A sequence of such events forms a pattern.
- Whenever a new pattern is found, it's labelled and saved. This enables the self-learning capacity of the network.
- If the pattern occurs again, the IoT would be able to recognize and report it. This forms a smart network with lesser human intervention.

Conclusion

A small scale IoT has been successfully simulated. The network works on a centralized architecture with the Edison as the central hub.

Further studies

- Move from a centralized architecture to a decentralized one where each node talks to each other on its own, with no need of a central hub.
- Apply data analysis techniques to the data collected to recognize patterns in the network.
- Apply self-learning capabilities to the network.

UNIVERSITY OF BIRMINGHAM

IBM®

Node-RED