



Internet of Things
2015 – 2016

Project title:
Data collection with Thingspeak

Mohammad Ala Amjadi 1

Introduction

A wireless sensor network (WSN) has important applications such as remote environmental monitoring and target tracking. This has been enabled by the availability, particularly in recent years, of sensors that are smaller, cheaper, and intelligent.

These sensors are equipped with wireless interfaces with which they can communicate with one another to form a network. The design of a WSN depends significantly on the application, and it must consider factors such as the environment, the application's design objectives, cost, hardware, and system constraints.

Scenario

In this project we implemented two simulated WSN on different channels. Each of the WSNs contains two sensor nodes (Temperature and Humidity) and one sink node.

Sensors will send their data to the sink node then sink node will send it through serial port to node-red tool. We will simulate the wireless sensor networks on Cooja simulator.

In node-red we buffer the received data, parse it and send the data of each WSN to a specific channel of ThingSpeak. We can read the collected data from ThingSpeak in Node-red tool, in order to trigger an alarm.

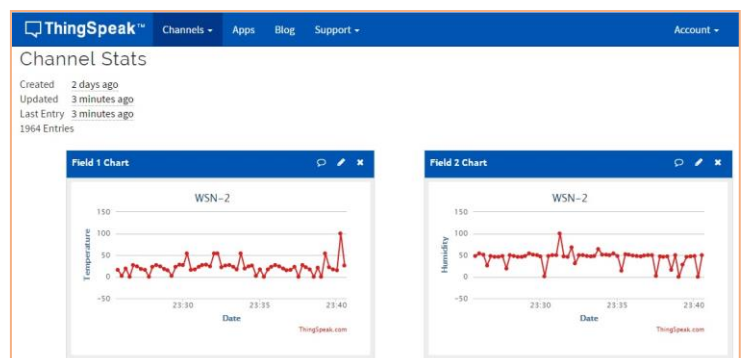
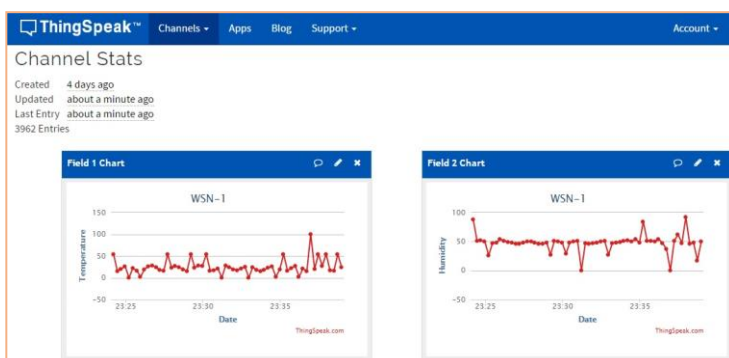
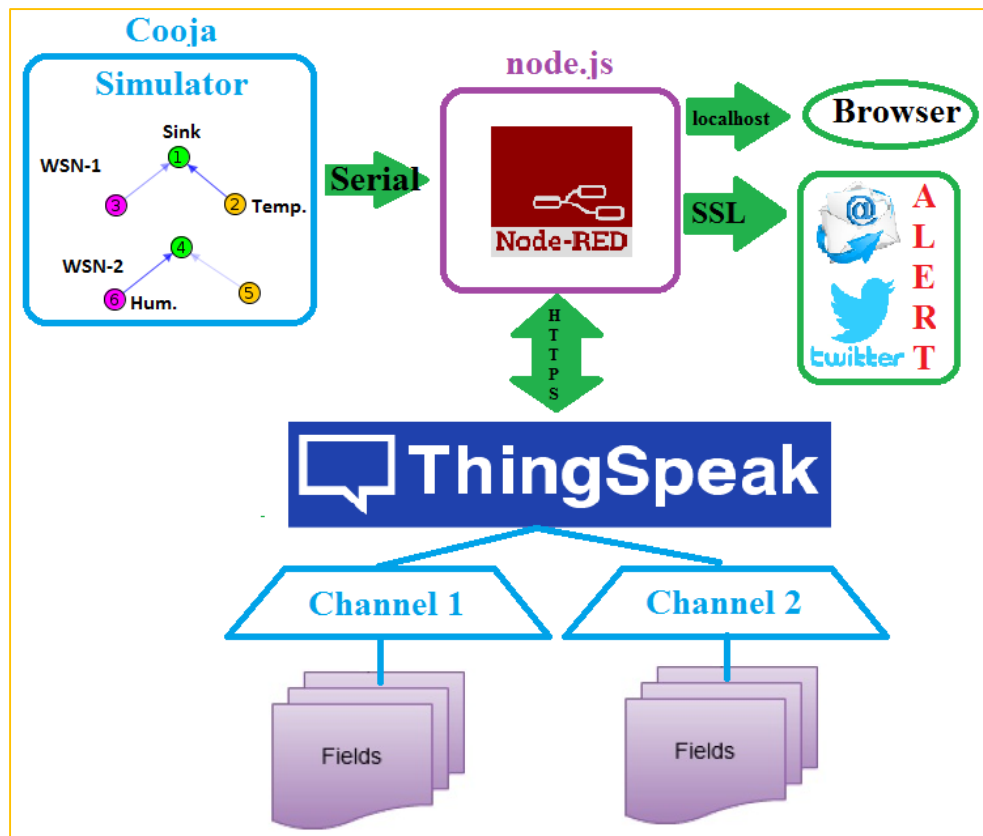
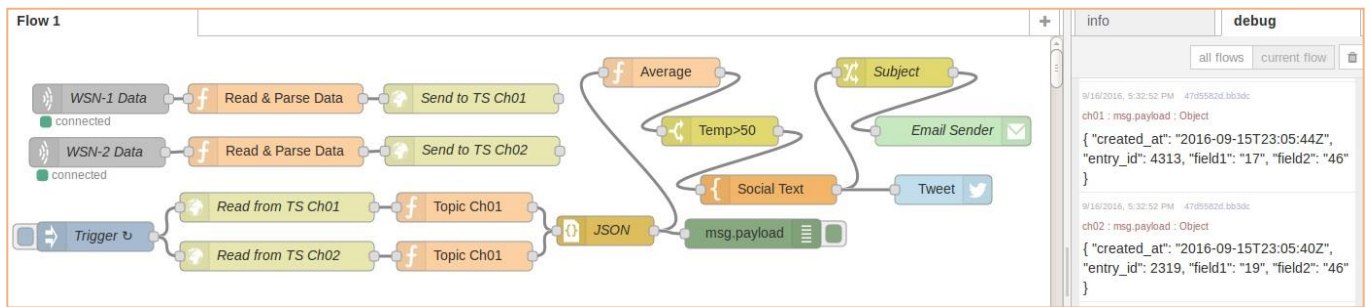
Step by step scenario:

- Motes will boot, start their radio interfaces
Each packet has its own type ID in order to differ between sensors and sink motes.
 - Sensors: sense the temperature or humidity and send the value to sink mote.
 - Sink: collects both packets from sensor motes, make a new packet and send it through serial
- Simulation of WSNs is done in Cooja, the collected data will go through a serial socket server which we can have TCP connection on localhost ports
- Node-red will listen on the same port as Cooja is sending data
- Data received in Node-red are TCP packets. We should parse the packet payload to get the data out.
- Obtained data will be sent using HTTPS protocol over the internet to ThingSpeak with a specific write API key to store the data from each WSN on a specific channel
- There are two channels for our two WSNs, each channel has two fields, one for temperature sensor and the other one for humidity sensor
- After storing data on ThingSpeak, we want to read the data in order to work with different functions and set alarms on Node-red.
- Getting data from ThingSpeak will happen by using HTTPS protocol with a specific read API key. We have a lot of options for retrieving data. In this project we were interested to read the last result on ThingSpeak and have it on Node-red. We had put a trigger that ignite every one second for this http request.
- In Node-red we get the average of two last results from channel 1 and 2. We set a rule that if the average temperature of two WSNs is higher than 50 degrees trigger an alarm.
- The alarm can be different kinds. We used social one such as email and twitter.
- The real time data are monitors for any exceeding threshold which than can trigger an email alarm

The project contains:

- Files for each node including sink and sensor nodes are in their specific named folder (Sink, Temp and Hum)
- Each node has header, module and configuration files in their folders. Sensor nodes have specific SensorC and SensorP that will provide all the interfaces and commands needed to read from sensors. Since the project is simulated we used a header file for sensor's data.

The project can be seen as following diagram which shows a simple schematic on how different parts were connected and communicated with each other. The result of social alert is also available in the pictures.



node-red_alert

MOHAMMAD ALA AMJADI < @gmail.com >
To: @yahoo.com

This alert has sent to notify high Temperature detection: 54 Celsius !
Take care.

Reply, Reply All or Forward | More

Tweets Tweets & replies

Mohammad Ala Amjadi @AlaAmjadi · 1h

This alert has sent to notify high Temperature detection: 54 Celsius !
Take care.

Today at 1:06 AM