Embedded Systems Development Project

Final Report

EMBEDDED SYSTEM / INTERNET OF THINGS

3. joulukuuta 2017

Made By: Laura Pietikäinen

TVT15SMO

Embedded Systems Development Project

Final Report

1.1 What is IoT?

IoT = Internet of thing, meaning that you can connect your devices to internet.
Device could be anything you can imagine with a on and off switch.
For example: wearable device, headphones, cellphones, coffee maker, lamps or fridge. A normal device added with intelligence; sensors and some wires and connection to internet.

1.2 About MQTT

Mqtt = Message Queue Telemetry Transport protocol.

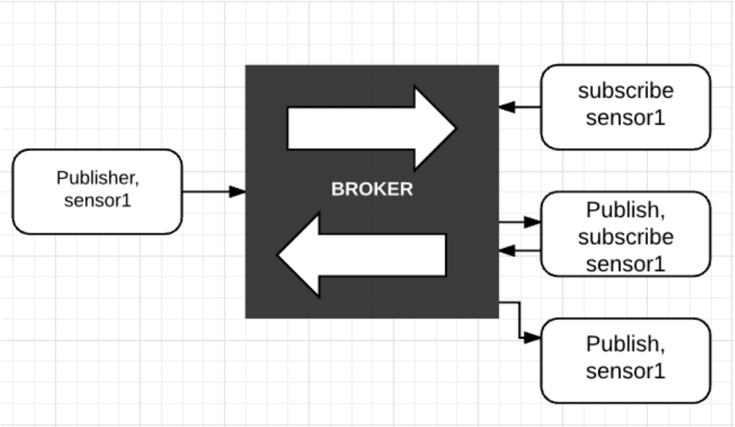
It is a machine-to-machine(M2M) connectivity protocol what is used in

Internet of Things (IoT) and in a Wireless Sensors Networks (WSN).

Mqtt has become a main protocol for Internet of things.

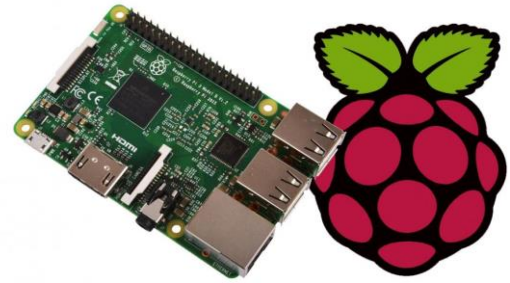
It is a lightweight publish and subscribe message pattern, meaning that any device can publish data and any client can subscribe that data. This happens in the MQTT ***broker*** what keeps track on all publications and subscribes.

The job of an MQTT ***broker*** is to filter messages based on topic and then distribute them to subscribers.

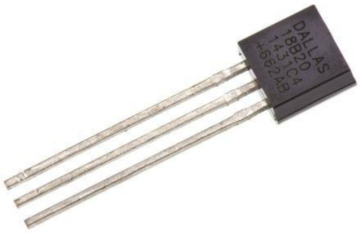


2.1 Project components

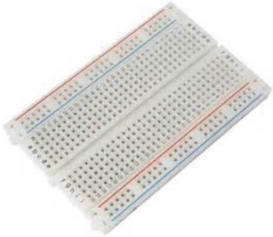
1.Raspberry Pi, it is a small single board computer.



2.Dallas DS18B20 is a 1-Wire digital sensor for measuring temperature. It reports degrees in Celsius with 9 to 12-bit precision. You can measure degrees between -55c to +125c (+/-0.5C). The sensor has unique 64-Bit Serial number in it and the connected device will use for identifying the sensor.



3. A Breadboard is where you add the sensor and resistor and connect it with wires.



4. 4.7kΩ resistor

2.2 Raspbian and Sensor Setup

Raspbian is an operating system for Raspberry Pi. In this Project I'm using version Jessie. There is a new version of Raspbian called stretch, but I had some issues to install MQTT broker in it, so I'm using the older one.

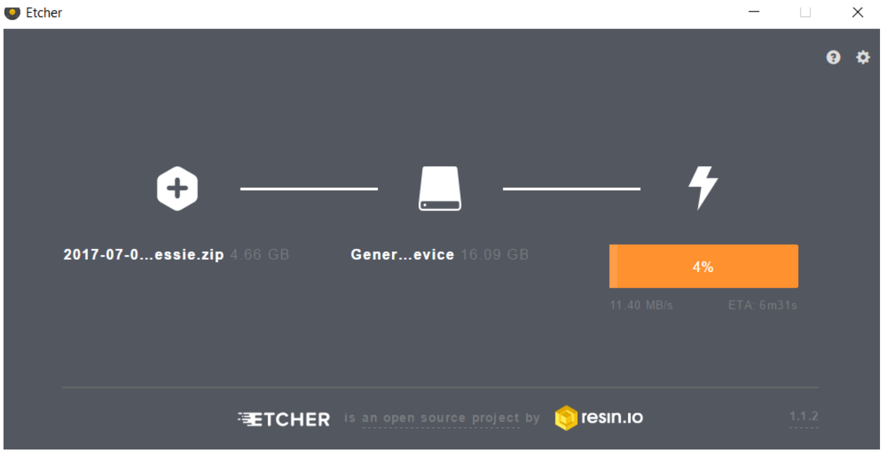
1. First step is to download Raspbian from: https://www.raspberrypi.org/downloads/ or http://downloads.raspberrypi.org/raspbian/images/ to download a different version.

2. Then download the Etcher from: https://etcher.io/ This is for burning images to your SD cards & USB drivers

3. To make sure your SD card is empty and in case you need to re install Raspbian use SD card formatter: https://www.sdcard.org/downloads/formatter_4/

4. Move the Raspbian files from downloads to your SD card and if there is a several folders in extracted files you need to move files to one folder and remove left overs. Otherwise the system will not see the files you need to install.

5. Use the Ether to burn the image. https://etcher.io/


Etcher
2017-07-0...essie.zip 4.66 GB
Gener...evice 16.09 GB
4%
11.40 MB/s
ETA 6m31s
ETCHER is an open source project by resin.io
1.1.2

6. Remove card reader or card safely from computer and insert it in to raspberry pi. Connect the power to raspberry and add HDMI cable between pi and monitor/tv or pc. If you are using it separate from your pc connect keyboard and mouse too.

7. Install Raspbian. Connect Pi to internet and set date.

8. Go to Preferences and from there to Raspberry Pi Configuration to set up the resolution from system and the go to interfaces and enable at least SSH, Serial, 1-wire, and Remote GPIO.

9. If you want to use SSH connection install Putty to your pc from: http://www.putty.org/ Open the Terminal window from pi and type **hostname -I** it will give you your Pi's IP address what do you need to write to Putty's address line, then choose SSH connection and use the default Pi and Raspberry login. In this point you might want to change your password and to do that type: **sudo passwd,** to terminal window, or change it from pi's configurations.

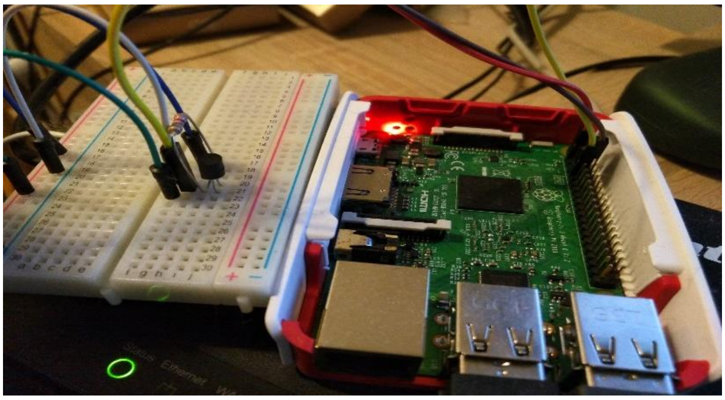
10. Go to Pi's files and add a following line to /boot/config.txt
dtoverlay=w1-gpio
then connect pi to temperature sensor:

GPIO GND (Pin 6),

GPIO 3.3V power line (Pin 1)

4.7kΩ resistor (+ (powerline), (DS18B20 pin2)

DS18B20+ GND (Pin 1, Pin 6, (DS18B20 Pin2 -> Pin 7/GPIO 4



11. To add drivers and read the temperature from command line type:

$sudo modprobe w1-gpio

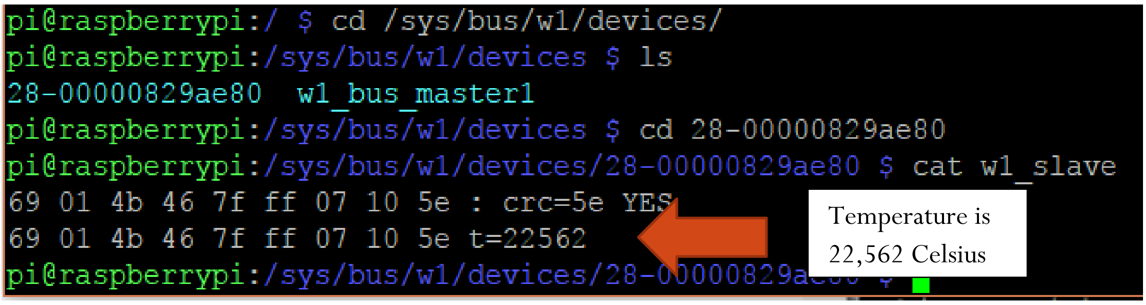
$sudo modprobe w1-therm

go to 1-wire device folder type:

$cd /sys/bus/w1/devices/

then type **ls** to see your sensor and then **cd 28-00000xxxxxx**

type **cat w1_slave** to see the temperature.



```
pi@raspberrypi:/ $ cd /sys/bus/w1/devices/
pi@raspberrypi:/sys/bus/w1/devices $ ls
28-00000829ae80  w1_bus_master1
pi@raspberrypi:/sys/bus/w1/devices $ cd 28-00000829ae80
pi@raspberrypi:/sys/bus/w1/devices/28-00000829ae80 $ cat w1_slave
69 01 4b 46 7f ff 07 10 5e : crc=5e YES
69 01 4b 46 7f ff 07 10 5e t=22562
pi@raspberrypi:/sys/bus/w1/devices/28-00000829a
```

Temperature is 22,562 Celsius

Test code: https://github.com/LauraPietikainen/Embedded-Systems-Development-Project/blob/master/first_code_ver1.py

2.3 CREATE IOT WITH NODES

1. **Install npm package manager for node.js components.**

   For installing the npm, type to terminal window:

   $sudo apt-get update

   $sudo apt-get upgrade

   $sudo apt-get install nodejs npm

2. **GitHub to Raspbian**

   $sudo apt-get update

   $sudo apt-get install git-all

   ------Configure------

   $git config --global user.name "Your Name"

   $git config --global user.email email@example.com

   To get started:

   https://projects.raspberrypi.org/en/projects/getting-started-with-git

2.4 Node Red dashboard

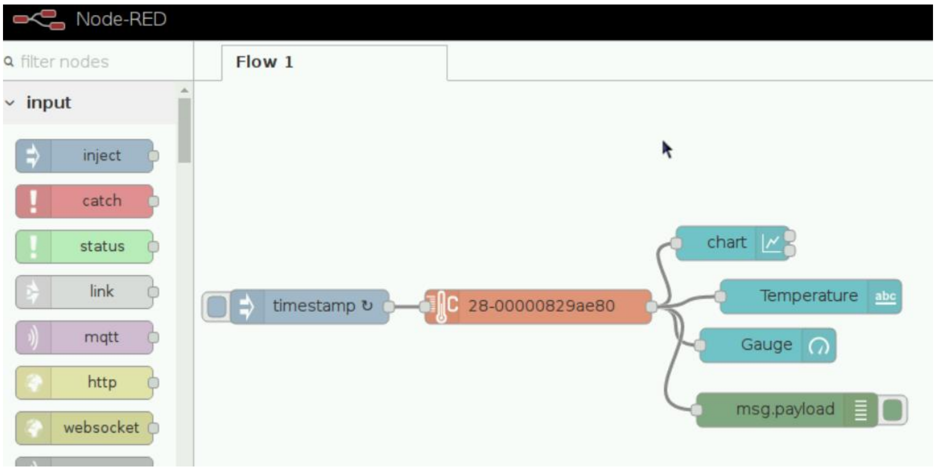
This is A visual tool for wiring the Internet of Things

Type to terminal window:

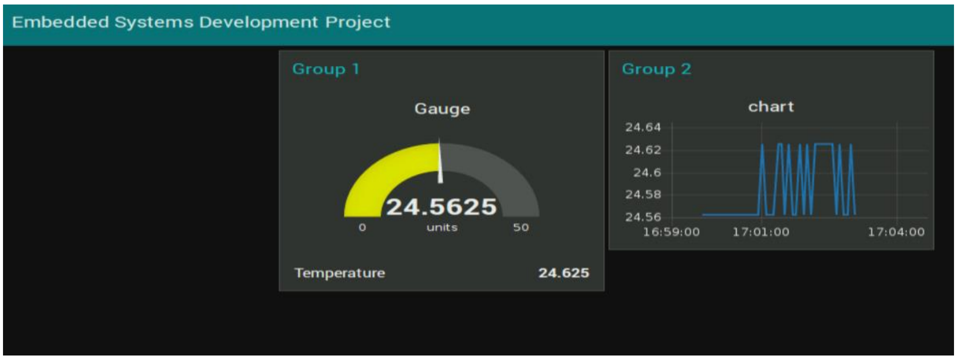
```
$ sudo npm install -g node-red
$ node-red
$cd node-red
$npm install
$sudo npm install -g grunt-cli
```

Install / Upgrade Script:
https://nodered.org/docs/hardware/raspberrypi
$update-nodejs-and-nodered
Open localhost and add more available nodes from manage palette, for example node-red-dashboard and DS18B20 for measuring temperature.

Node RED Flow



Node RED Dashboard



3.1 Cayenne My Devices

1.Create a free account to my Devices: https://mydevices.com/

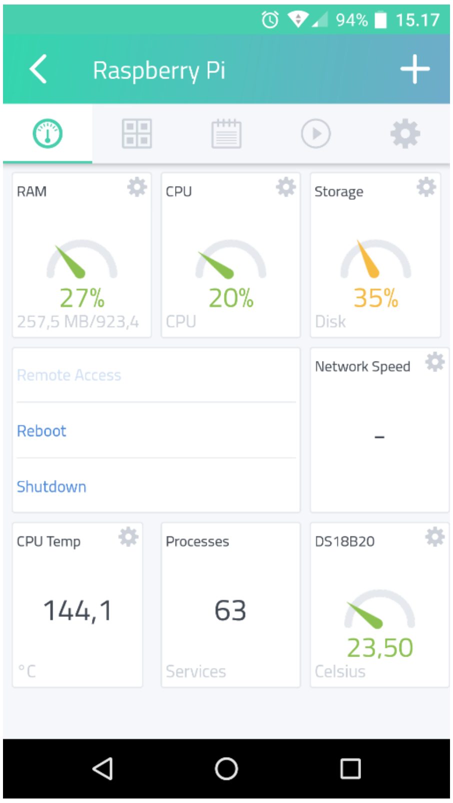
This is a Cloud service where you can add your raspberry Pi.

There is a IoT Project builder called **Cayenne**. It allows you to remote and control your devices from your PC or from mobile app. It can all so read sensors that are connected to Pi's GPIO pins.

Cayenne MQTT API is in use when you connect your device, you don't have to use terminal window to get it, it will install everything when connecting to Raspberry Pi. If the app does not find the device, it asks to type Pi's IP address to connect.

here is a link to Cayenne: https://mydevices.com/cayenne/docs/bring-your-own-thing-api/
This YouTube link shows the basics how you can manage your IoT in Cayenne: https://youtu.be/a_0xyMPGamY

Cayenne Dashboard and sensor chart



RAM 27% 257,5 MB/923,4

CPU 20% CPU

Storage 35% Disk

Remote Access

Reboot

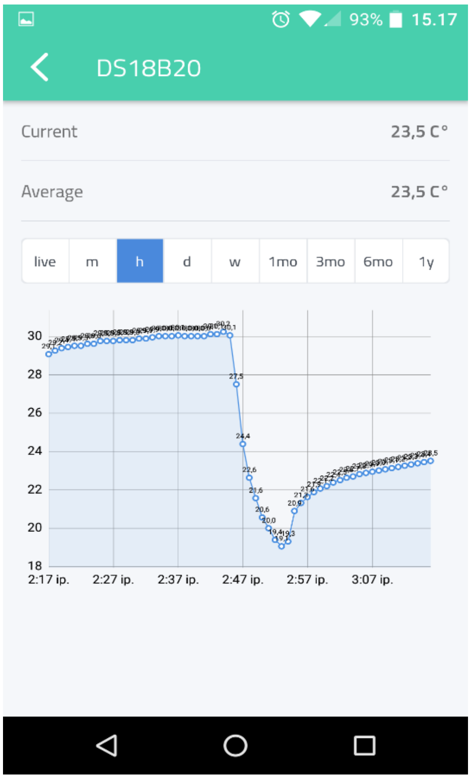
Shutdown

Network Speed –

CPU Temp 144,1 °C

Processes 63 Services

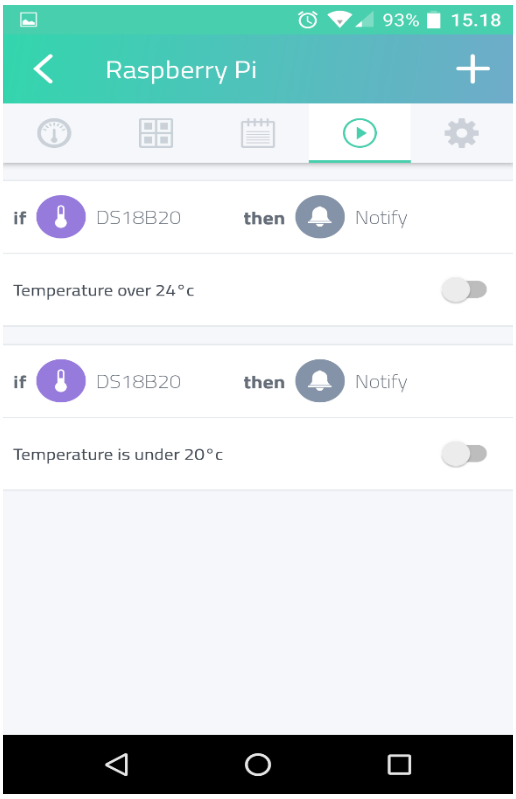
DS18B20 23,50 Celsius



Current 23,5 C°

Average 23,5 C°

Trigger can send a message to your email, if the temperature is off the limits.

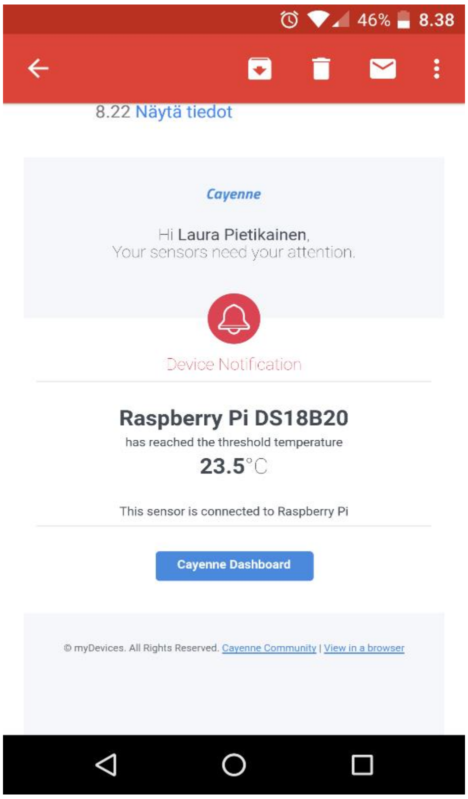


if DS18B20 then Notify

Temperature over 24°c

if DS18B20 then Notify

Temperature is under 20°c



8.22 Näytä tiedot

Cayenne

Hi Laura Pietikainen,
Your sensors need your attention.

Device Notification

Raspberry Pi DS18B20
has reached the threshold temperature
23.5°C

This sensor is connected to Raspberry Pi

© myDevices. All Rights Reserved. Cayenne Community | View in a browser

4.1 LINKS AND SOURCES

- https://cayenne.mydevices.com
- https://www.modmypi.com/blog/ds18b20-one-wire-digital-temperature-sensor-and-the-raspberry-pi
- https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php
- https://github.com/LauraPietikainen/Embedded-Systems-Development-Project
- https://nodered.org
- https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf
- https://en.wikipedia.org/wiki/1-Wire