

Red Pitaya as an IoT device

Tomaž Rojc,

University of Ljubljana, Faculty of Computer and Information Science,

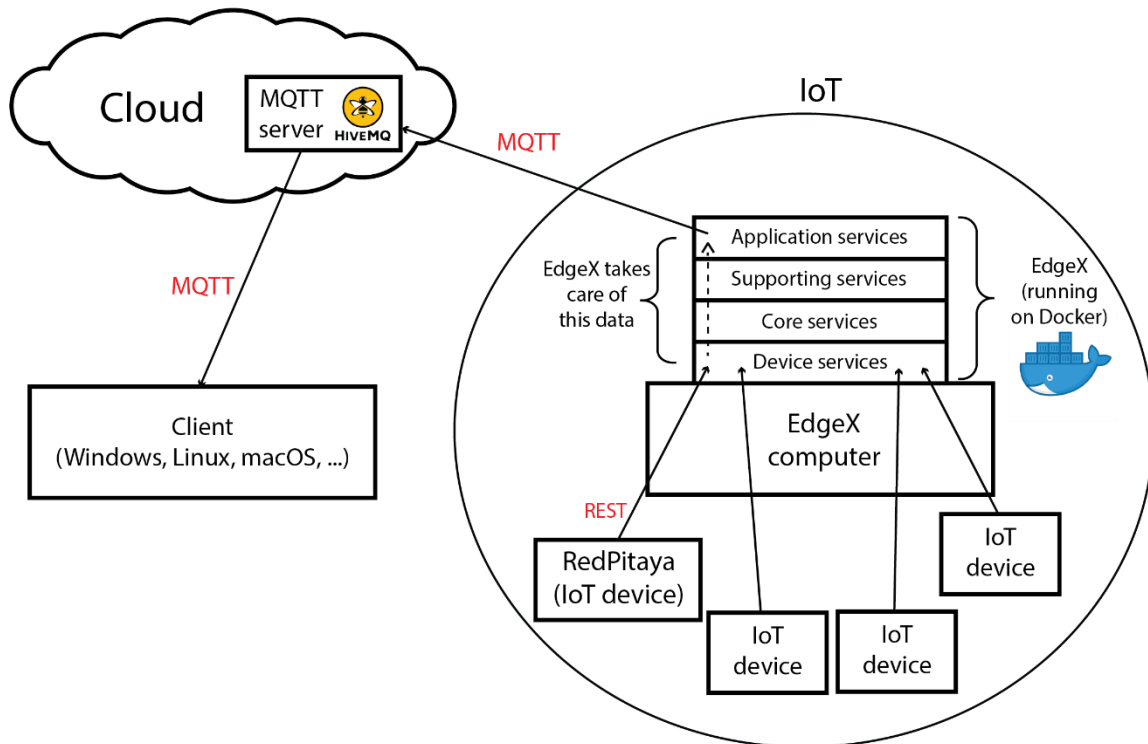
Email: tomaz.rojc2000@gmail.com,

Mentor: Uroš Dragonja

1. Introduction

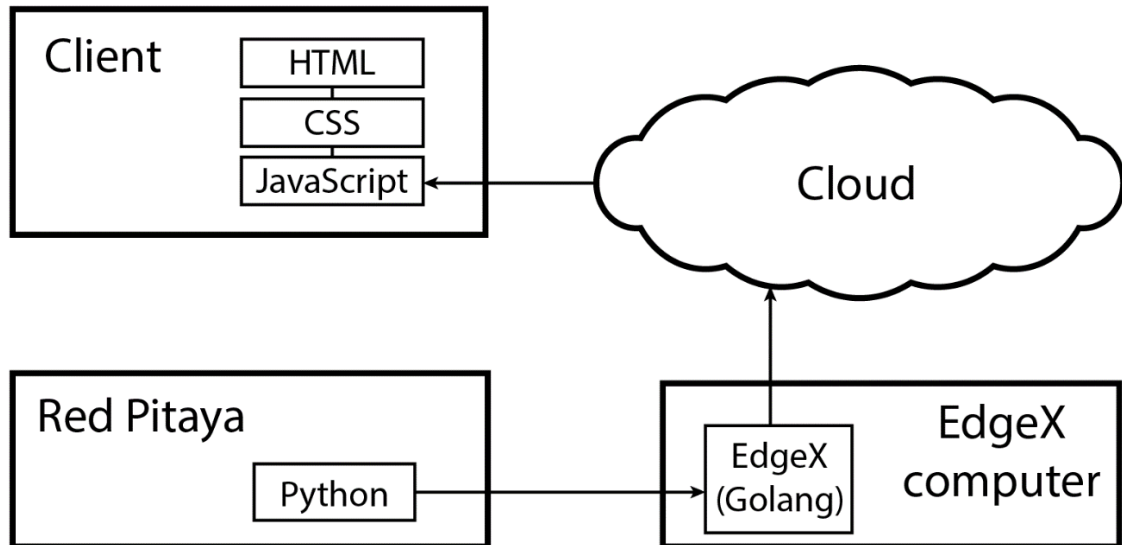
The IoT (Internet of Things) describes devices, such as sensors, that are connected together and exchange data over the internet. This data can also be sent to cloud servers, where it is stored. EdgeX Foundry is an open-source framework that allows you to communicate with IoT devices and configure them to send or receive data automatically. I turned the Red Pitaya SIGNALlab 250-12 into an IoT device. I used it's input 1 as a sensor and sent the readings of amplitude to the cloud.

2. Architecture



EdgeX is running on Docker that is installed on a computer in the local network. In this network we can plug multiple IoT devices. EdgeX provides a wide range of protocols to connect to any device. In this case the Red Pitaya is using the REST protocol to send measurements to EdgeX. From there EdgeX takes care of the data, which is ultimately sent to the cloud over the MQTT protocol. I used HiveMQ's free public MQTT server for testing purposes. The last step was to create a simple web application that connects to the MQTT server and gets the readings. This app should work on any network, because the data is in the cloud.

3. Software



It all starts with a Python script that reads the measurements on Red Pitaya's input 1 and sends it to EdgeX Foundry. The script is stored on Red Pitaya and before you run it you have to enable the SCPI server, because the script can also be used from another computer that communicates with Red Pitaya over this server. This is how you can get measurements from Red Pitaya's inputs and outputs. I use localhost for the server's IP, so the Red Pitaya communicated with itself and the script operates autonomously.

Next, we come to EdgeX. This has to be configured to accept Red Pitaya's data, process it and send it to the cloud. It has to be told exactly which kind of data it will be accepting, and which kind of data it will be sending. The data is then sent to the cloud to a certain MQTT server under the desired MQTT topic.

Finally the application that reads data from the MQTT server is written in HTML. It uses the Eclipse Paho JavaScript library to connect to the desired MQTT topic, subscribe to it and get the messages from server. It parses and displays them on the screen. It also plots a simple graph of amplitudes that it has read.

This is how the demo application looks like in disconnected and connected states:

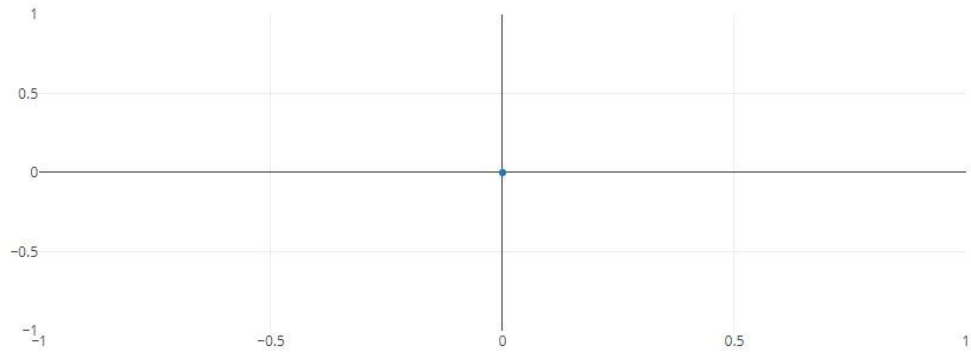
Insert your topic:

Start capturing data

Stop capturing data

Level:

Connection status:



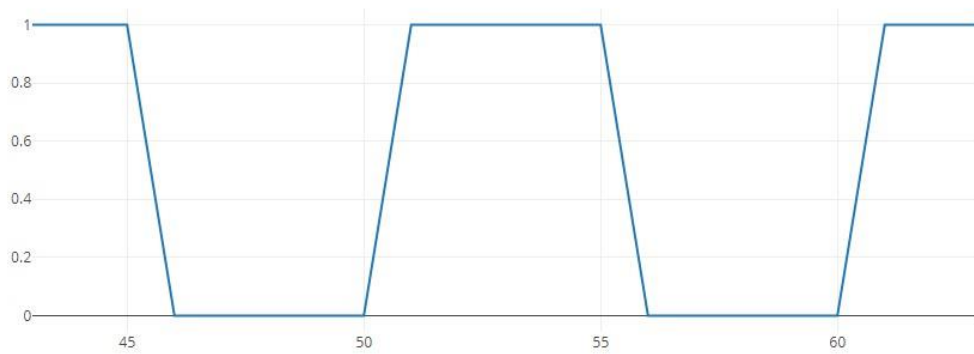
Insert your topic:

Start capturing data

Stop capturing data

Level:

Connection status:



4. Conclusion

This is a great way to turn a Red Pitaya or any other device into an IoT device. You can see live data on the server and it is accessible from anywhere. There is so much more that EdgeX Foundry can do, and it has great potential. The only drawback is that it is a bit complicated to get started, although the benefits that EdgeX has to offer outweigh the work you put into it.