**Project Proposal for programming project (Ubiquitous Computing – Course ID: 2012)**

**Virtual smart home automation system simulating MQTT network**

**Course Teacher: Fredrik Kilander**

**Submitted By: Rafat Khan (rafatk@kth.se)**

The goal of this project is to create a virtual smart home automation system simulating a complete MQTT network, which is an OASIS standard messaging protocol for the Internet of Things (IoT). The MQTT (Message Queue Telemetry Transport) data will be cached by an HTTP server and made available via HTTP as well as via WebSockets to allow for real time updates in the dashboard. I will build separate program for different components which will communicate using MQTT network and can be deployed to IOT devices using azure-iot-device functions.

Technology

I have planned to implement the system using JavaScript and node JS framework.

Components

Weather station

The weather station will display the current temperature and a description of the weather on initial load (data will be fetched via HTTP GET from openweathermap API).

Heating System

The temperature will drop, if we turn off the heater and if it’s colder outside. If it’s hotter outside, the temperature will slowly rise.

Blinds

There will be configuration option to set the mode of the blinds to manual and change the value of how open we want the blinds to be. If set to automatic, blinds will remain shut between sunset and sunrise and open between sunrise and sunset.

Dashboard

The UI will be implemented in HTML/CSS with frameworks such as Font Awesome, jQuery and more.

MQTT MQ Telemetry Transport

MQTT (MQ Telemetry Transport) is a lightweight IoT messaging protocol based on the publish/subscribe model. It is a simple messaging protocol which provides an easy solution for communication between devices. It can provide real-time and reliable messaging services for networked devices with very little code and bandwidth. It is widely used in the industries such as the IoT, mobile Internet, smart hardware, Internet of Vehicles and power energy.

Constrained Devices

Low Bandwidth

So, perfect solution for IOT applications

What can we do:

Send a command to control an output

Read data from sensor and publish it

MQTT basic concepts

Publish/Subscribe

Messages

Topics

Broker: Aedes - Stream-based MQTT broker.

MQTT Clients:

Publishes a message to a connected MQTT broker.

mqttClient.publish();

mqttClient.on("connect", () => {

subscribe();

});

Heating System: Publish Data (Heating Elements), Subscribe Different Topics (Weather, Heating and Simulation Speed)

Roller Blind: Publish Data (Roller Blind), Subscribe Different Topics (Weather, Blinds)

Weather Station: Publish Weather Data and Alerts

Http Server is running on "http\_port": 3000

\*Publish current weather data every 2 seconds

MQTT broker is running on "mqtt\_port": 1885,

An MQTT broker is a server that receives all messages from the clients and then routes the messages to the appropriate destination clients. An MQTT client is any device (from a micro controller up to a fully-fledged server) that runs an MQTT library and connects to an MQTT broker over a network.

MQTTServerWrapper:

MQTT-Wrapper package provides easy-to-use MQTTv3 and MQTTv5 connection for Go projects. It supports Request/Response pattern for MQTTv5 connection.

In the case of MQTT over Websockets the websockets connection forms an outer pipe for the MQTT protocol. The MQTT broker places the MQTT packet into a websockets packet, and sends it to the client. The client unpacks the MQTT packet from the websockets packet and then processes it as a normal MQTT packet.