

IoT and Networking

1 Overview

The goal of this project is add IoT support to Project 1 by implementing an MQTT client. The solution must be implemented on a TM4C123GXL board using an ENC28J60 ethernet interface. The solution will be able to publish topics and subscribe to topics on an MQTT broker, such as Mosquitto. As the intent of this project is to understand the details of these simple elements, your code solution can be based only on provided class code and code you write. You should not be incorporating more than a few lines of code from other sources.

2 Command-line Interface Requirements

The solution must provide these additional command-line interface using UART0 and configuring the device and reading out the status. The command-line interface should support the following commands additional (to Project 1) commands at a minimum:

set MQTT w.x.y.z

This command sets the IP address of the MQTT broker and stores this values persistently in EEPROM.
publish TOPIC DATA

This command publishes a topic and associated data to the MQTT broker.

subscribe TOPIC

This command subscribes to a topic and then prints the data to the screen when a topic is received later.

unsubscribe TOPIC

This command unsubscribes from a topic.

connect

This command send a connect message to the MQTT broker.

disconnect

The command disconnects from the MQTT broker.

3 Power-Up Requirements

On power-up, the solution must initialize the clocks, UART, ethernet controller, and timer service.

The MQTT address must be retrieved from EEPROM on power-up.

If a valid MQTT address is provided, then the MQTT client should try to connect.

4 Additional Command-line Interface Requirements for Rules

help INPUTS

The command displays the possible local inputs to the MQTT client. These nouns normally include the pushbuttons, temp sensor, time, UART, and UDP port.

help OUTPUTS

This command displays the possible local outputs to the MQTT client. These nouns normally include the LEDs, UART, and UDP port.

help SUBS

This command lists the current subscribed topics.

If supporting the time option or sunrise/sunset options, you will need to add a time and date setter (and perhaps getter) functions. For sunrise/sunset, you will also need to add a lat/long getter. The hibernation module RTC will be useful to maintain the time, but timer 1 can also be used.

If supporting a UDP port as an input/output, you will need to add port support somehow.

if CONDITION then ACTION

CONDITION is one of the following:

TOPIC|INPUT (the reception of this topic is the trigger)

TOPIC|INPUT ==|!=|>|>=|<|<= VALUE (evaluations the condition)

TOPIC|INPUT ==|!=|>|>=|<|<= VALUE && TOPIC ==|!=|>|>=|<|<= VALUE2

where the TOPIC is a subscribed MQTT topic. An event is one of the local inputs.

ACTION is one of the following:

TOPIC|OUTPUT = VALUE2

Examples:

if /env/temp >= 75 then LED = 1

if PUSHBUTTON == 0 then /env/message = ButtonPressed

The rules should accept a wild card (#) as in the following:

Suppose topics /env/temp and /env/humidity exist. Subscribing to /env/# gets both messages.