Project 2: Pathfinder

Luke Brandon
Secret Phrase: subway

***Overview*—The goal of this project was to emulate a scene out of the book "The Martian" where an astronaut stranded on Mars communicates with NASA via a servo motor, pointing to 16 different hex values that can be converted into ASCII. We were to start communication with an MQTT server that is provided (NASA), read in values of angles that are written to the servo motor, and then converted into ASCII that can be read easily by humans. I accomplished this goal by utilizing the on board WiFi module of the MKR 1000, and MQTT library called PubSubClient, and some code read in the values and do all the necessary conversions and control of the servo motor and the LCD Screen. The outcome of this project was a fully functional "Pathfinder" emulation device that accomplishes all of the necessary tasks.**

I. Introduction

THE goal of this project was to emulate a scene out of the book "The Martian" where an astronaut stranded on Mars communicates with NASA via a robot that has a camera mounted on a servo, pointing to 16 different hex values that can be converted into ASCII. We were to start communication with an MQTT server that is provided (NASA), read in values of angles that are written to the servo motor, and then converted into ASCII that can be read easily by humans. We were also to subscribe to the uark/csce5013/lhbrando/lcd topic that will send instructions on what to display on the LCD screen that is necessary to wire up, this will display the status of the "Pathfinder" which is mostly "broken" and once all 3 of the phrases that were sent from the server were correctly received and decoded, then the LCD display would display the secret code-word that is sent on the uark/csce5013/lhbrando/phrases topic as well, namely "subway".

II. Application Design

This device required the use of the WiFi101 library that makes use of the on-board WiFi chip on the MKR1000, as well as the PubSubClient library which is an MQTT library that is used in order to communicate with the MQTT broker to get the phrases, LCD information, and angles. Establishing a WiFi connection is relatively straightforward using the library, simply call a conned method with the name and password of the WiFi network that you want to use. MQTT is a pub/sub protocol, so once the WiFi was connected and MQTT client initialized, it was necessary to subscribe to 3 different topics uark/csce5013/lhbrando/lcd, uark/csce5013/lhbrando/phrase, and uark/csce5013/lhbrando/angles. The LCD topic would send a string that should be displayed onto the LCD screen utilizing a function that I wrote in order to simplify writing to the LCD screen. The angles topic is the topic in which the broker (NASA) sent all of the angle values that are to written to the servo. I then had to decode these angles into Hexadecimal values and convert those into the correct ASCII character on the ASCII table (each ASCII character is 2 hexadecimal bytes), once this phrase was decoded, I published the decoded phrase to the same "phrase" topic also using a function that I wrote in order to do so and if correct the server would send a new phrase. After 3 correct phrases the server sends the secret phrase on the LCD topic, which is the final topic. In the messageReceived function that I wrote (a callback set in the MQTT client using the PubSubClient library) the code decides what to do depending on the topic, for angles and the phrase topics I describe what is done above, but for the LCD topic the message that is sent over MQTT is simply displayed onto the LCD screen. The main function of this entire project is the messageReceived callback function that is invoked every time a topic that is subscribed to receives a message from the MQTT broker, then smaller functions are dispatched that do rudimentary work in order to get the expected result whether it be conversion to ASCII, writing angles to a servo or writing strings to an LCD display. We were also tasked with making the LCD display handle lines longer than 16 characters by using the autoscroll feature and also new line characters. I did this by writing the LCD function in such a way that each character is printed and the loop has knowledge of home many characters have been written on a given line, if the number of characters is 15 then it will begin autoscrolling for all future characters. If the string comes across a new line character, the cursor will move to the second line and begin writing characters from the beginning of the second line. Because of the way the function is written, the autoscroll feature also works on this line as well. For the second MQTT Client that is used by the "operator", I used http://www.hivemq.com/demos/websocket-client/.

III. Results

The pathfinder clone that I build accomplishes all of the written test objectives, therefore the result of this project was fully functional pathfinder clone according to the assignment guidelines.

IV. Future Work

If I were to continue working on this project, I would abstract all of the hardware control logic into a library or a couple of small libraries so that the main code and MQTT control would be much cleaner.

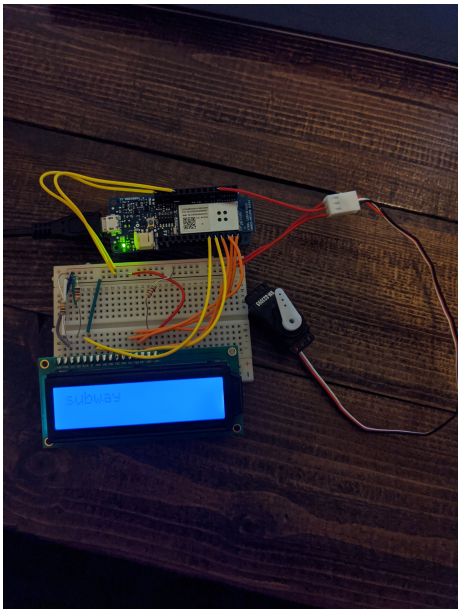

Fig. 1. Picture of the device fully assembled