

Assignment 2 - Pathfinder Hexadecimal Translation
CSCE 4/5013 – Wearable & Ubiquitous Computing – Fall 2020
Due Date – Friday, October 30th, 11:59 PM CT

Task: In the novel and movie adaptation “The Martian” by Andy Weir, astronaut Mark Watney uses a Mars Rover with a mounted camera to communicate with NASA. He writes on a piece of paper, and the image of those words are communicated by satellite to NASA. To communicate back, NASA sends commands to the Pathfinder rover to point the camera at 16 different locations, each representing a hexadecimal character. Watney writes these down, and translates them to the English Language through the ASCII encoding scheme. They use this communication to deliver a patch that Watney must apply that eventually allow communication of plain-text through the Pathfinder LCD screen. For this assignment, you will create an approximation of the communication between NASA and Pathfinder.

Expected Operation: To implement the communication between “NASA” and “Pathfinder”, you will use a Servomotor (attached to Pin 9) to point at the 16 Hex characters, equally spaced in 180 degrees. This would be character ‘0’ at 0 degrees, character ‘F’ at 180 degrees, and each other character situated 12 degrees separated from one another. “NASA” – A server application – will send comma-separated angles to the Arduino, which must point at the correct characters in sequence. The operator will write these characters down, translate them into an ASCII message. Instead of using a camera, the operator will instead send the ASCII message to “NASA” via an MQTT client running on a desktop or server application (python scripts on Turing using the example code is preferable). Assuming a correctly entered phrase, “NASA” will send the next phrase for a total of 3 phrases. If all 3 phrases are completed correctly, the “Pathfinder” will be patched, and you can now display a secret phrase using the JHD 162A LCD display. The secret phrase will be sent immediately following the third correct entry, and should be displayed on the LCD display. This secret phrase is to be the subtitle of your report.

The MQTT communication will be through the “thor.csce.uark.edu” server, and will communicate with a server application. The port is 1883, and there is no user authorization/authentication. The root topic for all communication will be “uark/csce5013/username/” where username is your uark username without the “@uark.edu”. Servo angles will be published to “uark/csce5013/username/angles”, and messages to the LCD will be published to “uark/csce5013/username/lcd”. The secret phrases sent from a client application to move to the next phrase will be published on “uark/csce5013/username/phrase”, and will respond in-kind on that same topic with an “OK” or an “ERROR” to acknowledge receipt of the phrase, and let the client know if the message was correct or incorrect. The server will start the first message when the client publishes a message of “HELLO” to “uark/csce5013/username/phrase”.

Rubric: The project will be graded according to the following rubric:

Category	Description	Percentage
Pass Tests	There are a set of ten example tests in this document which you must implement. Each test is worth 5% of the grade for this project.	50%
Hardware Diagram	You should draw a schematic demonstrating how you designed the circuitry for the project	25%
Coding Comments & Style	Arduino uses individually written code and a set of libraries based on C/C++. Use the appropriate coding style. Comment functions with a description about their behavior, any parameters, any return values, and any shared variables which it manipulates.	10%
Report	A simple, one- or two-page report (using the class template). The report should have the project name (e.g. Light Gauge), a picture of the setup, a short description of what you did, and the outcomes (e.g. Did it pass all example tests, if not, why not, your design choices, etc...).	15%

Table 1: Grading Rubric

Tests: The following tests will be run, and the expected output is shown:

1. On boot-up the system displays “BROKEN” on the LCD
2. A client application publishes “HELLO” to the server to start communication
3. After “HELLO” is sent, the Arduino should begin to move the servomotor arm and point towards the HEX characters, pausing momentarily at each character
4. The client application enters each of the three decoded messages sent by the “Pathfinder”
5. After all three decoded messages are sent, the secret phrase should display on the LCD screen
6. Publishing “Hello” to the server application will reset the LCD back to the “BROKEN” state by receiving a message from the server on the LCD topic with the a payload of “BROKEN”
7. (10-points) The LCD should display single-line messages of more than 16 characters using the autoscroll function in the Arduino LCD library.
8. (10-points) The LCD should display a message with a “\n” as a two-line message separated by the new-line character (will be less than 16 characters for each, so no autoscroll)

Submission: You should zip your project directory and submit it through the online submission portal at `csce.uark.edu/~ahnelson/file_upload/`. The directory is password protected using the same username and password that we have used for the class.