Software Overview

Year: 2018 Semester: Spring Team: 16 Project: Track-on-track

Creation Date: January 24, 2018 Last Modified: January 25, 2018

Author: Email:

Member 1: Nick Geirland Email: ngeirlan@purdue.edu

Member 2: Nathan McNally Email: nmcnall@purdue.edu

Member 3: Yunsheng Li Email: li1436@purdue.edu

Member 4: Aaron Kaiser Email: kaiser20@purdue.edu

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Software Overview** |  | x2 |  |  |
| **Description of Algorithms** |  | x2 |  |  |
| **Description of Data Structures** |  | x2 |  |  |
| **Program Flowcharts** |  | x3 |  |  |
| **State Machine Diagrams** |  | x3 |  |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** |  | x2 |  |  |
| **Formatting and Citations** |  | x1 |  |  |
| **Figures and Graphs** |  | x2 |  |  |
| **Technical Writing Style** |  | x3 |  |  |
| **Total Score** |  | | |  |

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

1.0 Software Overview

Track-On-Track will require a firmware component and a phone application software component. One main functionality of the firmware will be to wait for incoming commands from the phone application software. When an SMS message command is received, the firmware will need to securely validate the sender before issuing a response. Our initial plan for user authentication will be to use a simple password system. This way if a user does not have access to the phone originally used to communicate with the device, they can still issue commands via a different phone. A Bluetooth connection will be synced accordingly to the Bluetooth protocol. Once a connection is validated, the microcontroller’s firmware will communicate with either the GSM cell modem or Bluetooth module to retrieve the device’s GPS coordinates or connection strength, respectively.

If the request is an SMS request for GPS coordinates, the microcontroller’s firmware will construct a message that includes the coordinates and ship it off to the cell modem to be sent back to the phone application. This information will then be used with Google Maps in order to display the device’s current location to the user directly. With a Bluetooth connection, the microcontroller will need to determine the connection strength to relay back to the phone application. This information will then be displayed to the user in the form of getting ‘closer’ or ‘further’ depending on the changes in the Bluetooth signal strength. If the request is utilitarian, such as changing a setting, sounding the speaker, or displaying information, the microcontroller will update its memory, send a signal through the PWM to the speaker, or send data to the LCD using SPI.

The phone application software will need to display the location of the device on a map once it receives the coordinates from the device. It will also be required to issue commands to the device whether through SMS or Bluetooth; these commands can be used to retrieve the device’s location or to sound a speaker on the device to locate at short distances. The device’s settings will also be configurable through the application software, which includes triggering airplane mode, viewing battery life, displaying lost information on the LCD, etc.

2.0 Description of Algorithms

Our firmware will need to parse the incoming request strings to determine which action to perform. It will also require a table lookup for processing the cell modem’s return codes.[[1]](https://cdn-shop.adafruit.com/datasheets/SIMCOM_SIM5320_ATC_EN_V2.02.pdf)

The phone application will use Bluetooth signal strength thresholds to display approximate proximity to the device when using a Bluetooth connection. The GPS coordinates received from the cell modem will require a conversion from a raw string format to proper minute and second GPS format to be displayed by the phone application.

We will also be monitoring the rechargeable, lithium-ion battery’s charge level. Multiple threshold voltages will be set (determined by the battery’s datasheet) to display a battery life meter on the LCD to indicate if the device needs charged. This battery level will also have to be sent to the phone application upon request.

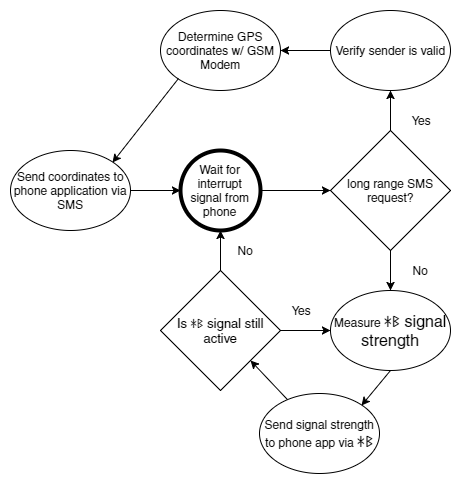
3.0 Description of Data Structures

All the communications between the microcontroller, the cell modem and the bluetooth module are via UART interface. The microcontroller will send display instructions to the LCD display through SPI interface. The instructions will be sent through 2G cellular network and received by the cell modem. The cell modem will convey the instructions to microcontroller. Then the microcontroller sends instructions to the cell modem to get its location information which is then transmitted to user’s phone through SMS on the same cell modem. The cell modem communicates by sending and receiving strings while usually sitting in wait but it’s also able to interrupt in the case of a text being received. The cell modem is controlled by using the AT Hayes command set [1] with significant extensions added to support all of the functionality. The commands all have their own specific return values so most output from the modem will need to be checked against constants.

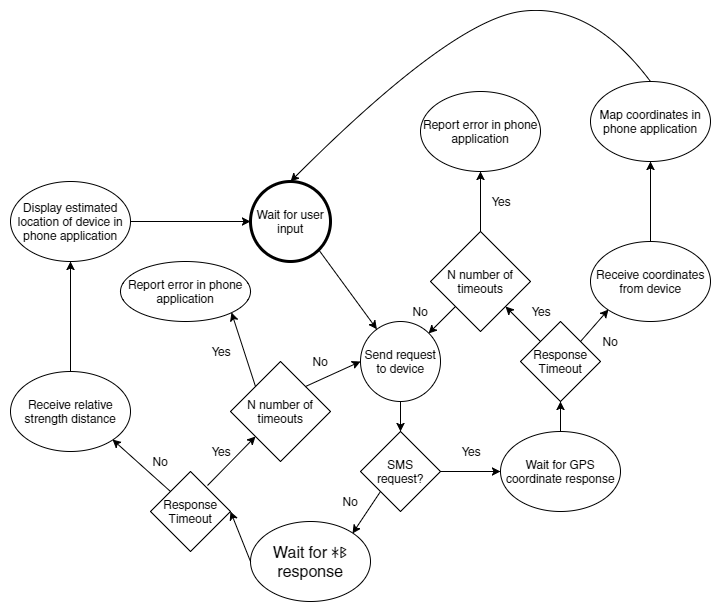
4.0 Sources Cited:

[1] Adafruit. (2014). AT Commands Set. [online] Available at: <https://cdn-shop.adafruit.com/datasheets/SIMCOM_SIM5320_ATC_EN_V2.02.pdf>[Accessed 26 Jan. 2018].

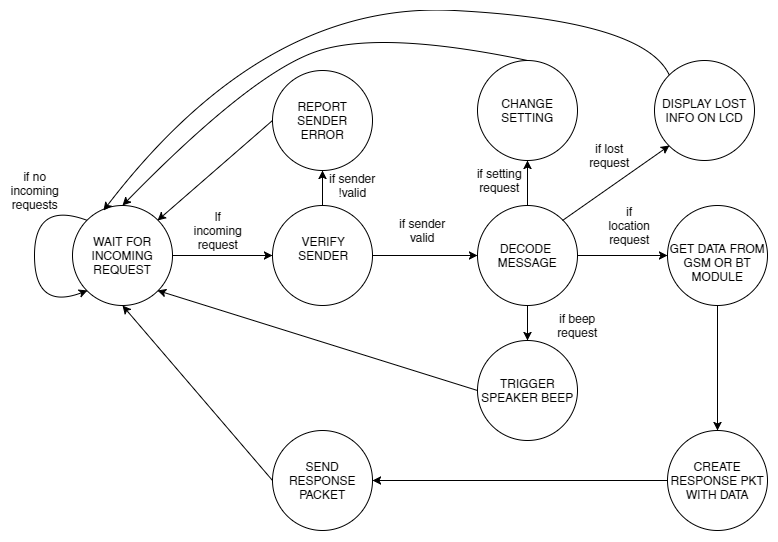
Appendix 1: Program Flowcharts

**

*Figure 1: Track-On-Track Firmware Flowchart*

*Figure 2: Track-On-Track Phone Application Software Flowchart*

Appendix 2: State Machine Diagrams



*Figure 3: State Transition Diagram*