Software Overview

Year: \_\_2017\_\_\_\_ Semester: \_\_\_Spring\_\_\_\_\_ Team: \_8\_\_\_\_ Project:\_\_\_\_\_\_Barbot\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Creation Date: \_\_\_\_\_\_\_01/26/2017\_\_\_\_\_\_\_\_\_ Last Modified: 01/26/2017

Author: \_\_\_\_Yudi Wu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Email: \_\_\_\_wu553@purdue.edu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Software Overview** | 4 | x2 | 8 |  |
| **Description of Algorithms** | 5 | x2 | 10 |  |
| **Description of Data Structures** | 5 | x2 | 10 |  |
| **Program Flowcharts** | 4 | x3 | 12 |  |
| **State Machine Diagrams** | 4 | x3 | 12 |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** | 4.5 | x2 | 9 |  |
| **Formatting and Citations** | 5 | x1 | 5 |  |
| **Figures and Graphs** | 5 | x2 | 10 |  |
| **Technical Writing Style** | 4.5 | x3 | 13.5 |  |
| **Total Score** | 89.5/100 | | |  |

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

General Comments:

1.0 Software Overview

The software part of our design could be split into two parts: microcontroller and mobile application. The mobile application will contain a user interface for users to choose their desired drink from known drink list or mix their favorite drink through custom component selection. If user chooses to select house drink, the drink menu page will show up and the list of all built-in drinks will be available for user to inspect. User could inspect drink by click on the name of the target drink, and then an image of the drink as well as the brief description and component list of the drink will show up in a new view. The other way for users to choose drink is to select alcohol and beverage customly by identify the beverage and quantity (25ml is quantity ‘one’).

After users choose their desired drink, they could order the drink by clicking on ‘order now’ button on drink description page. The data will be transmitted to microcontroller through UART protocol[2] of Bluetooth module. The microcontroller will accept the data from Bluetooth module and tell the numbers of cycle needed to send platform to target drink to stepper motor. The microcontroller will keep stepper motor working until all the required drink are processed. Microcontroller will also send the message “Drink is being made” to LCD and let it display the status of the machine, for instance, what drink is being made, how many beverages still need to be added.

The key firmware includes stepper motor, linear actuator and drink disperser. The stepper motor needs to drive the belt rolling and thus move the platform. After platform sent to the target place, the linear actuator is responsible to rise up the stroke and open the valve of the dispenser. After the valve is opened, microcontroller needs to tell linear actuator to withdraw the stroke.

2.0 Description of Algorithms:

In the mobile application, once the user selects the drink, the application will look up pre-designed combinations of the drink in the database. If it’s a customized drink, it will combine the selected drinks into one combination package. Then, this package is sent over the Bluetooth low energy service.

Once the microcontroller received the package, it will look up each individual drink locations in the stored information and push it into the processing queue. For each item of the queue, the microcontroller checks the location of the drink and calculate the distance between the platform’s current location and desired location. Because the stepper motor rotates half cycle each the wave pulse, we calculate how long half cycle move horizontally, and sent “number of pulses = distance / half cycle distance” to the stepper motor driver through PWM peripheral.

Furthermore, microcontroller need be capable to instruct the linear actuator to push the dispenser to dispense the drink. In order to protect the dispenser from the damage, we need to make sure the actuator do not overextend. First, we calculate the distance between the lowest position the actuator and the desired position. We know our linear actuator moves 10mm/s while powered, so the time to keep the positive voltage t = d/s. The PIC microcontroller will utilize its built-in TIM module to power the actuator time equal to t, so the actuator could only reach the desired position without damaging the dispenser.

3.0 Description of Data Structures:

To transfer data from Bluetooth module to microcontroller, UART module[2] is needed. The data package for UART is displayed in figure 1, and for maximum performance, we choose the data to be 7-bit, no parity bit and one stop bit. Microcontroller is responsible for catching those data chunks and combine them together.



Figure 1 - Sparkfun[1]

Hash table[3] is one important data structure for our design. In the mobile application, a string-to-list hash table is used to store the combination of drinks. One line of the table would be “key= ‘mojito’, value = ‘[1,1,1,1,2,2,4,5,5]’”. To save memory and time, drinks are stored in the system by numbers. And instead of storing the quantity of 25ml, we are just going to store each component separately(like “1,1,1” instead of 3 \* “1”) which simplified the data structure and the transmission. And in the microcontroller, a string-to-integer hash table is used to store the location info of each drink. Example of one line would be “key= ‘1’, value=‘124’”, and the value of this hash table indicate the distance between each dispenser and the left start point.

To process the drink combination, we need a FIFO to store drink locations in the SRAM. A queue is used to implement this. Microcontroller will always process the first item in the queue and store it in the current location. One start indicator and one end indicator such as “-1” should be added for the design.

4.0 Sources Cited:

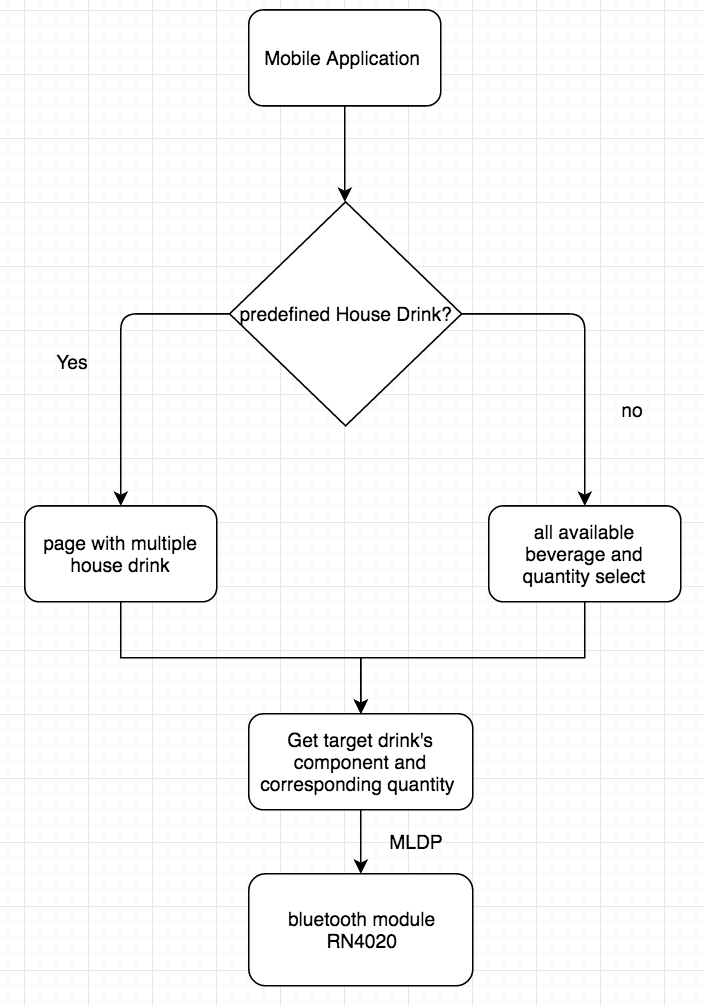
[1] JIMBO, "Serial communication," in Sparkfun. [Online]. Available: https://learn.sparkfun.com/tutorials/serial-communication. Accessed: Jan. 26, 2017.

[2] "Bluetooth® Low Energy Module," in Microchip. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/50002279A.pdf. Accessed: Jan. 26, 2017.

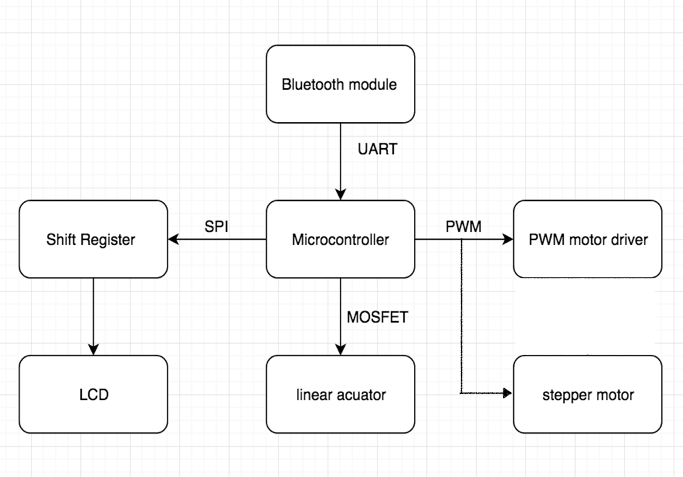
[3] J. Aspnes, "C/HashTables,". [Online]. Available: http://www.cs.yale.edu/homes/aspnes/pinewiki/C(2f)HashTables.html?highlight=(CategoryAlgorithmNotes). Accessed: Jan. 26, 2017.

Appendix 1: Program Flowcharts

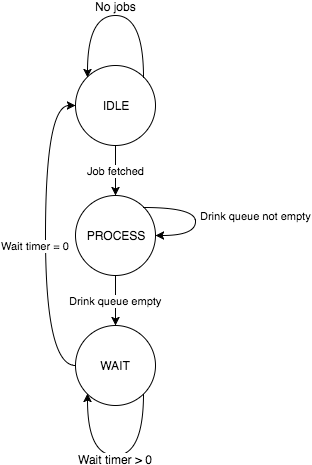
Mobile Application



Microcontroller



Appendix 2: State Machine Diagrams



Microcontroller state transition diagram