Software Formalization

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

Creation Date: \_\_\_\_\_\_\_02/22/2017\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Last Modified: March 3, 2015

Author: \_\_\_\_\_\_\_\_\_\_\_\_\_Mengshi Feng\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Email: \_\_\_\_\_\_\_\_\_\_\_feng96@purdue.edu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Third Party Software** | 5 | x2 | 10 |  |
| **Description of Components** | 5 | X3 | 15 |  |
| **Testing Plan** | 4 | x3 | 12 |  |
| **Software Component Diagram** | 5 | x4 | 20 |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** | 5 | x2 | 10 |  |
| **Formatting and Citations** | 4 | x1 | 4 |  |
| **Figures and Graphs** | 5 | x2 | 10 |  |
| **Technical Writing Style** | 5 | x3 | 15 |  |
| **Total Score** | 96 | | | Excellent! |

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

General Comments:

1.0 Utilization of Third Party Software

For this project, the main programming for the Barbot includes UART communication and Timer control. UART is used to communicate between Bluetooth RN4020 and PIC24FJ128GA010. The example code on the Microchip website is used as a reference for initializing RN4020 and PIC24. The rest code for the communication between Bluetooth and microcontroller will be developed by our team. The interrupt-based Timer is used to count time and it will also be developed by our team. There is no third-party software used in this project.

2.0 Description of Software Components

2.1 LCD: LCD is responsible for displaying the current status of the design. Data are sent over through 8 data pins, and 3 control bits are used to control the LCD driver. The LCD\_WriteString is able to write a string directly to LCD as long as the length of the string is no more than 15 characters (one-line limit of our chosen LCD). This part is ported to our project from the Microchip PIC24 example code. [2]

**2.2 Timer:** There are two timers in our design. Every 100ns, the timer generates an interrupt and every ten interrupts, it will update counters. The first timer is responsible for counting wait time, and the second timer is responsible for changing LED colors periodically. To be more specific, we have to give the stroke of linear actuator enough time to lift up, otherwise it will start going down before it reaches the valve of dispenser. The stroke will also need to stay still for two to three seconds for liquid in the buffer could be fully dispensed. The stepper motor also needs the delay, for example, the platform needs to stay at the place until the stroke of linear actuator fully shrinks. This part is fully developed by ourselves to meet our delay requirements.

**2.3 UART:** The data will be received by RN4020 through Bluetooth LE protocol from Android application.UART module is responsible for transmitting data between the microcontroller and RN4020. The microcontroller will generate two interrupts related to UART. One of them is for receiving data from Bluetooth module and the other is for transmitting data back to Bluetooth module. The ReadRxBuffer function could read one byte of data at a time from transmitting buffer, store the byte data into a buffer for microcontroller later parsing procedure. The WriteRxBuffer functions in the same way as ReadRxBuffer. It will write given one byte of data back to transmitting buffer. The IsNewData will determine whether there is new data coming or not. If yes, main loop will call ReadRxBuffer until the buffer is empty. I used the code I found from MicroChip RN4020 Bluetooth tutorial, and revise it so that it will work for the data we are transferring in our project. [1]

**2.4 Motor Drive Logic:** Motor drive logic is responsible for controlling two motors and their drivers. Upon receiving instructions from the Bluetooth, the logic instruct two motors to execute desired operation. For the linear actuator, the logic sent out ‘10’, ‘01’ or ‘00’ to move the stroke up, down or full stop. For the stepper motor, a PWM clock signal and 4 control bits are sent to L297. Then L297 could generate four control bits and it will be sent over to L298 for generating voltage signal. This part is fully developed by ourselves.

**2.5 Recipe structure and Data parsing:** The structure recipe has two fields: kind and data. Kind indicates the drink kind and the string “data” indicates the sequence of recipe. To minimize the total bytes of data being transferred, I schemed a system for formatting the data. The data will be sent in format of string. The format is as followed: "=>X[2,2,1,3...]\n". The "=>" basically tells the microcontroller start accepting and reading data. The following "X" represents one character that is the initial of the drink name. For instance, character "P" represents drink pina-colada, and "C" represents customized drink. It will be stored in “kind” field. This is for minimizing the string length. The numbers insides the bracket is the list of number ranged one to eight indicates the eight different available drinks. These numbers will be store as string in “data” field. The state machine for data parsing is as follows:

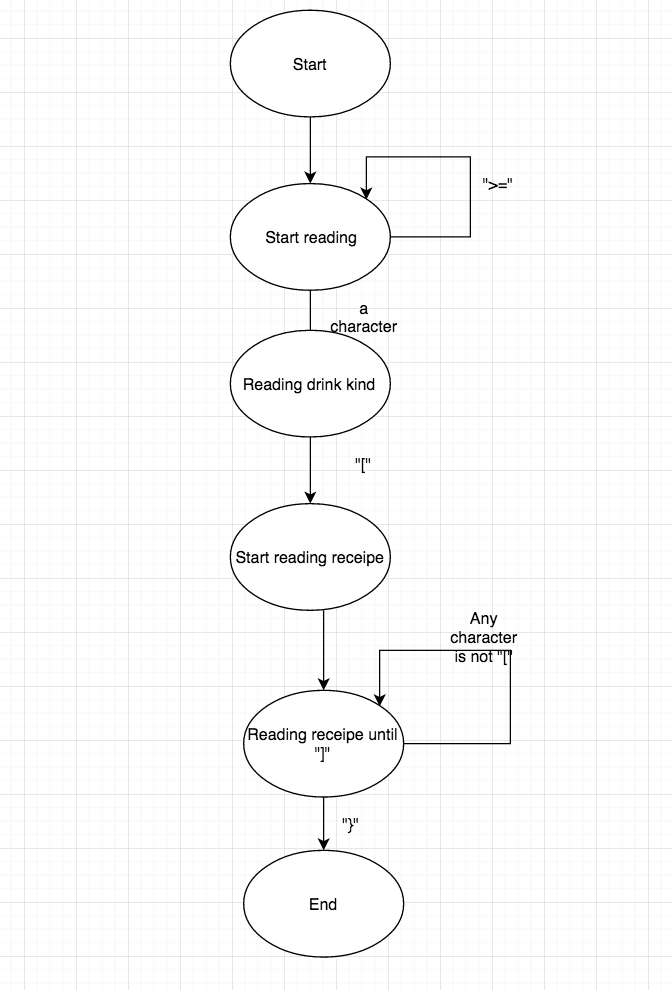


Fig 1 state diagram for data parsing.

3.0 Testing Plan

Each software component will be tested and debugged separately as follow.

3.1 LCD: It will display user’s choice and the status of drink mixing during the process. The test steps are:

1. Make sure LCD could display string, such as “Hello.” on the first line.
2. Clear the first line, and move the cursor to the second line.
3. Make sure LCD could display string, such as “World”, on the second line.
4. Make sure LCD could display strings on both lines simultaneously.

3.2 Timer: Timer is used to count time using interrupt. The test consist of:

1. Set the time limit for timer to countdown, such as 5 seconds.
2. Connect the Timer enable with one of the LED’s switch.
3. Manually activate the switch of LED, and using a stopwatch to count the time while the led lighting up.
4. Compare the result of stopwatch and the preset time limit.

3.3 UART: UART helps RN4020 communicate with microcontroller PIC24FJ128GA010. The test should follow:

1. Use the microcontroller to control the LCD to display the information sent from RN4020.
2. Connect mobile device with RN4020 and start choose a drink from mobile application.
3. Check the LCD display. The first line of LCD should display the name of drink, and the second line should display the recipe of drink as a string.
4. Repeat step 2,3 again and choose another drink, and check the information displayed on the LCD.

3.4 Motor Drive Logic: There two motor drives in Barbot. One is to drive linear actuator, and the other one is to control stepper motor. The test steps are similar for both two:

1. Build the driver circuit for both linear actuator and stepper motor. Connect the input pins with microcontroller’s I/O.
2. Programming microcontroller to set input for both driver logic.
3. Check whether both motors working as expected.

After all the individual tests are conducted, it can be make sure that all the components work as expected and they are ready for final project assembling.

4.0 Sources Cited:

[1] JIMBO, "Serial communication," in Sparkfun. [Online]. Available: https://learn.sparkfun.com/tutorials/serial-communication. Accessed: Feb. 23, 2017.

[2]“AN1861,” *Microchip Technology Inc*. [Online]. Available: http://www.microchip.com//wwwAppNotes/AppNotes.aspx?appnote=en572728. [Accessed: 23-Feb-2017].

Appendix 1: Software Component Diagram



Fig 2 Overall software component of our design

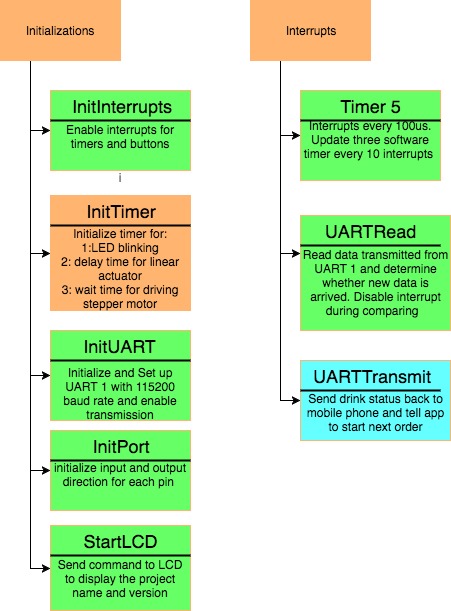


Fig 3 Software component for initializations and interrupts routines.

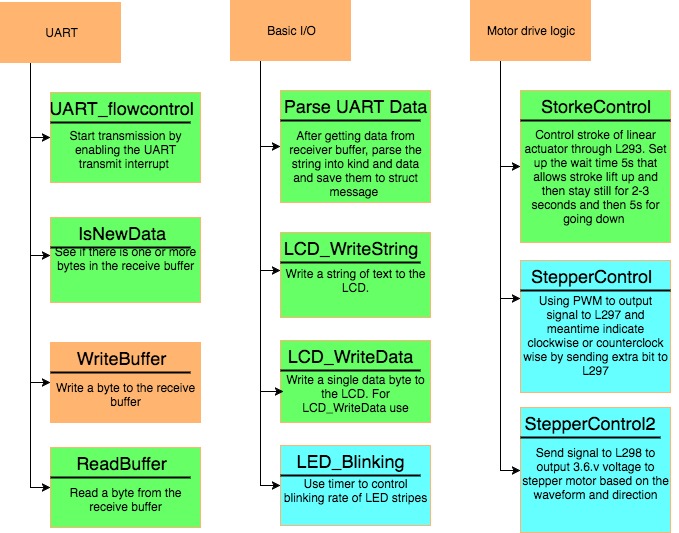


Fig 4 Software components for UART protocol, basic I/O and motor drive logic

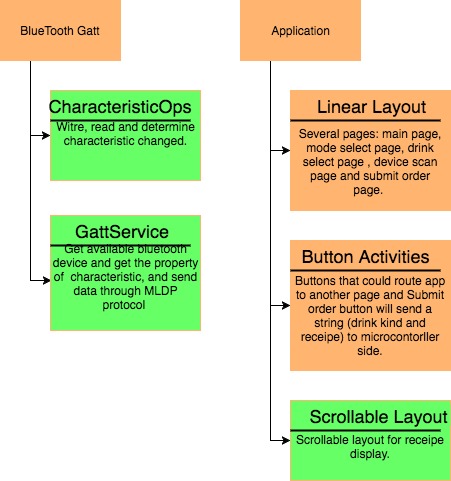


Fig 5 Software component for Android Application