Software Formalization

Year: 2018 **Semester:** Fall **Team:** 07 **Project:**Handi_glove
Creation Date: October 3, 2018 **Last Modified:** October 5, 2018

Member 1: Yaodong Shen **Email:** shen234@purdue.edu
Member 2: Jia En Chua **Email:** chuaj@purdue.edu
Member 3: Yao Chen **Email:** chen1748@purdue.edu
Member 4: Carol Lo **Email:** lo40@purdue.edu

Assignment Evaluation:

Item	Score (0-5)	Weight	Points	Notes
Assignment-Specific Items				
Third Party Software	5	x2	10	
Description of Components	4.5	X3	13.5	
Testing Plan	4	x3	12	
Software Component Diagram	4.5	x4	18	
Writing-Specific Items				
Spelling and Grammar	4.5	x2	9	
Formatting and Citations	5	x1	5	
Figures and Graphs	4.5	x2	9	
Technical Writing Style	4.5	x3	13.5	
Total Score	90			

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

General Comments:
Please check the comments.

1.0 Utilization of Third-Party Software

The major part of microcontroller coding uses the standard HAL libraries with the STM32F0CubeMX. Other softwares used are shown in the table below.

Name	License	Description	Usage
Eclipse IDE for C/C++ Developers [1]	Eclipse Public License (EPL) [7]	Eclipse is an integrated development environment used in computer programming	To be used as a base workspace for the project's software development.
EAGLE [2]	Autodesk EAGLE license [8]	Electronic design automation software for designing PCBs.	To be used for all our project's PCB design.
Fusion 360 [3]	Autodesk Fusion 360 license [9]	Cloud-based CAD/CAM/CAE tool for collaborative product development	To be used for our CAD designs to show physical layout of the project components.
STM32Cube MX [4]	SLA0048: Mix Ultimate Liberty+OSS+3rd-party V1 - SOFTWARE LICENSE AGREEMENT [10]	Graphical software configuration tool that allows the generation of C initialization code using graphical wizards.	To be used for configuration initialization.
Cura [5]	https://github.com/Ultimaker/Cura/blob/master/LICENSE [11]	Software used for 3D printing preparations.	To be used for preparing 3D printing.
Repetier-host [6]	https://github.com/repetier/Repetier-Host/blob/master/Repetier-Host-licence.txt [12]	Software used for 3D printing preparations.	To be used for preparing 3D printing.

2.0 Description of Software Components

The software system is made up of several major features for this project. First of all, the project is aimed to mimic hand gesture through robotic fingers, implement temperature and pressure feedback system, return robotic finger backs to default position when glove is empty and optionally wireless connection between glove and robotic hand. Therefore, we will break down each feature and talk about their detail implementation in software.

One of the features is to control robotic fingers through value from potentiometer. Therefore, we utilize HAL_DMA method on STM32L1 to read in multiple channels of potentiometers' ADC values. We carry out experiment to record the range of ADC values and use these values to formularize an equation and calculate the PWM duty cycle for servos. Then, we transmit these values from the glove's microcontroller to the robotic hand's microcontroller(micro). After that, we need to configure the timer PWM on the robotic hand's micro and utilize the equation to generate appropriate PWM duty cycle for the servo to turn correctly and hence mimicking the hand gesture.

Then, we need to read temperature and pressure sensor value on robotic hand's micro. Similarly, we utilize DMA method to read multiple ADC channels and transfer these values to glove's micro. For pressure feedback system, we need to set high voltage for linear actuator to extend and push syringe to expand air inside of airbag. Therefore, we need to utilize the sensors' values and calculate the time we need to set on linear actuator. However, there are five linear actuators in total and hence we can't do the setting one by one. Therefore, we figured out a way to move all five together and configure each by using timer interrupt.

For temperature feedback system, we need to generate heat/cooling effect by using the temperature sensor's values. Not only that, we will also modify the heat generated by using a secondary set of sensors on glove to reduce the error deviation. The Peltier module is configured by setting two PWM values to control the magnitude and polarity of temperature generated. We will have several timer interrupts which constantly run each of these features to make sure entire system work as expected.

At last, we installed a touch sensor on glove to detect if the glove is empty and control the robotic fingers' movement. For example, if the touch sensor detects someone is using the glove, the glove's micro will send appropriate values to the other micro and work as usual. However, if the glove is empty, the micro will send default values to the robotic hand's micro to reset robotic fingers back to default position.

3.0 Testing Plan

The project follows a strict test-driven development pattern where all components need to be tested individually before integration. The specific testing plan is shown below.

3.1 Hardware I/O controller on robotic hand
There are four major peripherals involved with the robotic hand, ADC combined with TIM module to sample both the temperature and pressure perceived by the robotic hand, PWM module to control the movement, and finally UART module for data transmission. Besides, since the communication module applies the full-duplex mode, it also needs to be tested to ensure data can be transmitted in both directions.

1. Be able to drive the servo with angles
2. Be able to drive the linear actuator which is the fundamental factor for the pressure feedback system.
3. Be able to test the real temperature and check with
4. Be able to control the pressure according to the control glove
5. Be able to receive a data packet of N bytes value from STM32 on the glove
6. Be able to transmit N bytes value to STM32 on the glove

3.2 Hardware I/O controller on the glove
There are also four major peripherals involved with the glove, PWM module to reproduce temperature and pressure feedback, ADC module combined with TIM module to sample the movement angle from the glove, and finally UART module to transmit data. In order to mimic the glove movement on the robotic hand, the precise data of the movement metadata, in particular, the data on the sensors of the exoskeleton, needs to be sampled.

1. Be able to sample the movement metadata
2. Be able to reproduce the temperature feedback
3. Be able to reproduce the pressure feedback
4. Be able to receive N bytes value from STM32 on a robotic hand
5. Be able to transmit N bytes value to STM32 on a robotic hand

3.3 Feedback system
We have implemented standard policy to test both temperature and pressure feedback system. The ultimate goal for both feedback systems is to generate heat/pressure which are closest to the value emitted from sensors. We transmit the ADC value directly from sensors and utilize these values to generate heat or pressure. However, there might be slight deviation of error between expected and actual value due various reasons such as environmental temperature affection, heat causing air expansion and so on. Therefore, we have written code to do a secondary check on the heat/pressure that were generated on the glove using another set of sensors. In other words, we check values generated from Peltier Module and airbag and take these values to match with the expected values which are from robotic hand's sensor. Then we set a minimum range for two set of values to be different. If the difference exceeded our threshold, we modify Peltier Module and linear actuator to generate proper heat/touch effect.

Overall the software system is made up of several different features. Therefore, we need to set an appropriate timeline and priority for the success of this project. The list below contains a set of tasks and each of them is assigned a number to indicate the priority with '1' being the most prioritized.

1. Read in ADC value from potentiometer, touch sensor and temperature sensor.
2. Generate PWM duty cycle and control servo using potentiometer voltage value.
3. Move linear actuator by setting GPIO high time by using touch sensor's value.
4. Generate heat/cooling effect from Peltier Module by using temperature sensor's value.
5. Implement UART to transmit and receive values both ways.
6. Read in touch sensor value to check if glove is empty and set the robotic fingers back to default position.

4.0 Sources Cited:

[1] D. Schaefer, "Eclipse CDT | The Eclipse Foundation", *Eclipse.org*, 2018. [Online]. Available: https://www.eclipse.org/cdt/. [Accessed: 06- Oct- 2018].
[2] "EAGLE | PCB Design Software | Autodesk", *Autodesk.com*, 2018. [Online]. Available: https://www.autodesk.com/products/eagle/overview. [Accessed: 06- Oct- 2018].
[3] "Help", *Help.autodesk.com*, 2018. [Online]. Available: https://help.autodesk.com/view/fusion360/ENU/. [Accessed: 06- Oct- 2018].
[4] "STM32CubeMX - STM32Cube initialization code generator - STMicroelectronics", *St.com*, 2018. [Online]. Available: https://www.st.com/en/development-tools/stm32cubemx.html. [Accessed: 06- Oct- 2018].
[5] "Ultimaker Cura: Advanced 3D printing software, made accessible | Ultimaker", *Ultimaker.com*, 2018. [Online]. Available: https://ultimaker.com/en/products/ultimaker-cura-software. [Accessed: 06- Oct- 2018].
[6] "- Repetier Software", *Repetier Software*, 2018. [Online]. Available: https://www.repetier.com/. [Accessed: 06- Oct- 2018].
[7] "Eclipse Public License - Version 1.0", *Eclipse.org*, 2018. [Online]. Available: https://www.eclipse.org/legal/epl-v10.html. [Accessed: 06- Oct- 2018].
[8] "Getting Started with Autodesk EAGLE Licensing | EAGLE | Blog", *Eagle Blog*, 2018. [Online]. Available: https://www.autodesk.com/products/eagle/blog/how-to-get-started-with-a-new-autodesk-eagle-license/. [Accessed: 06- Oct- 2018].
[9] "Autodesk - Fusion 360", *Creative Tools*, 2018. [Online]. Available: https://www.creativetools.se/software/cad-software/fusion-360/autodesk-fusion-360. [Accessed: 06- Oct- 2018].

[10] "STM32CubeMX - STM32Cube initialization code generator - STMicroelectronics", *St.com*, 2018. [Online]. Available: https://www.st.com/en/development-tools/stm32cubemx.html. [Accessed: 06- Oct- 2018].
[11] "Ultimaker/Cura", *GitHub*, 2018. [Online]. Available: https://github.com/Ultimaker/Cura/blob/master/LICENSE. [Accessed: 06- Oct- 2018].
[12] "repetier/Repetier-Host", *GitHub*, 2018. [Online]. Available: https://github.com/repetier/Repetier-Host/blob/master/Repetier-Host-licence.txt. [Accessed: 06- Oct- 2018].

Appendix 1: Software Component Diagram

Figure 1: Overall software components

STM32L152ZET6's Peripheral Modules

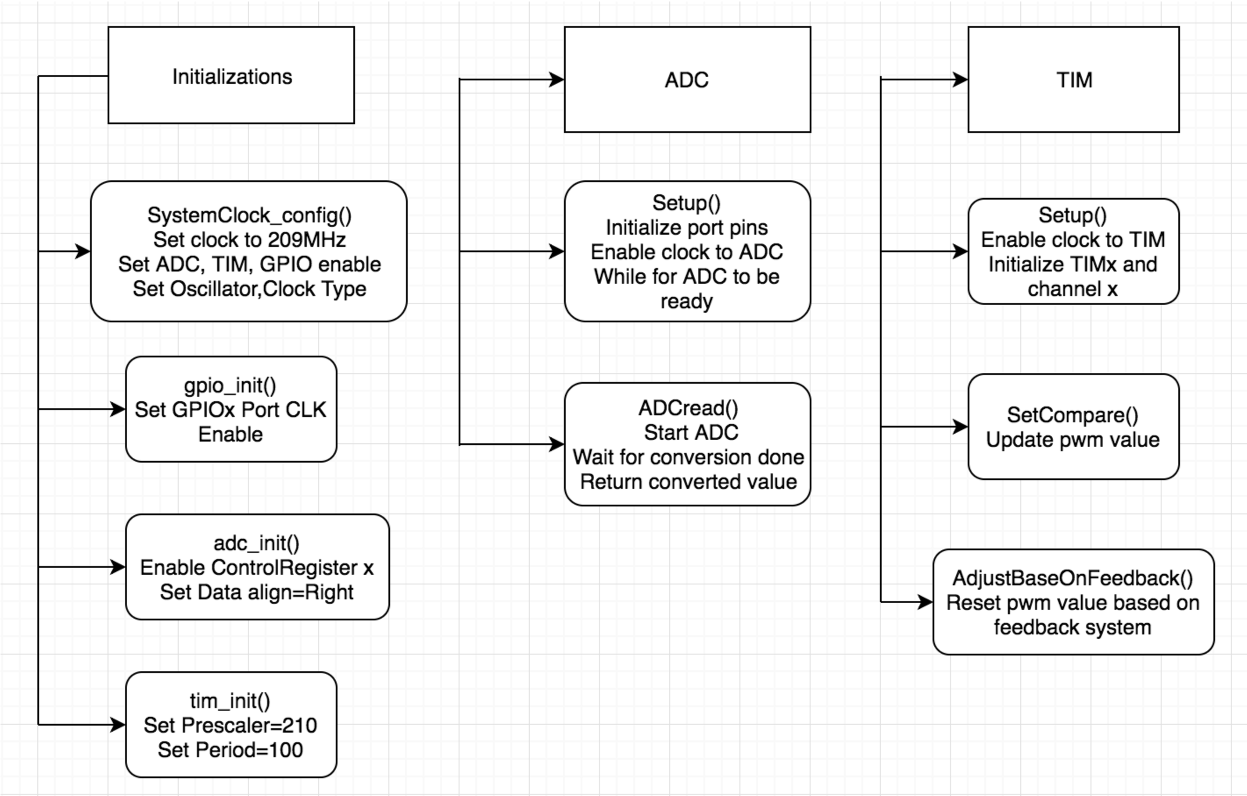

Figure 2: Each module's software formalization

Helper Functions

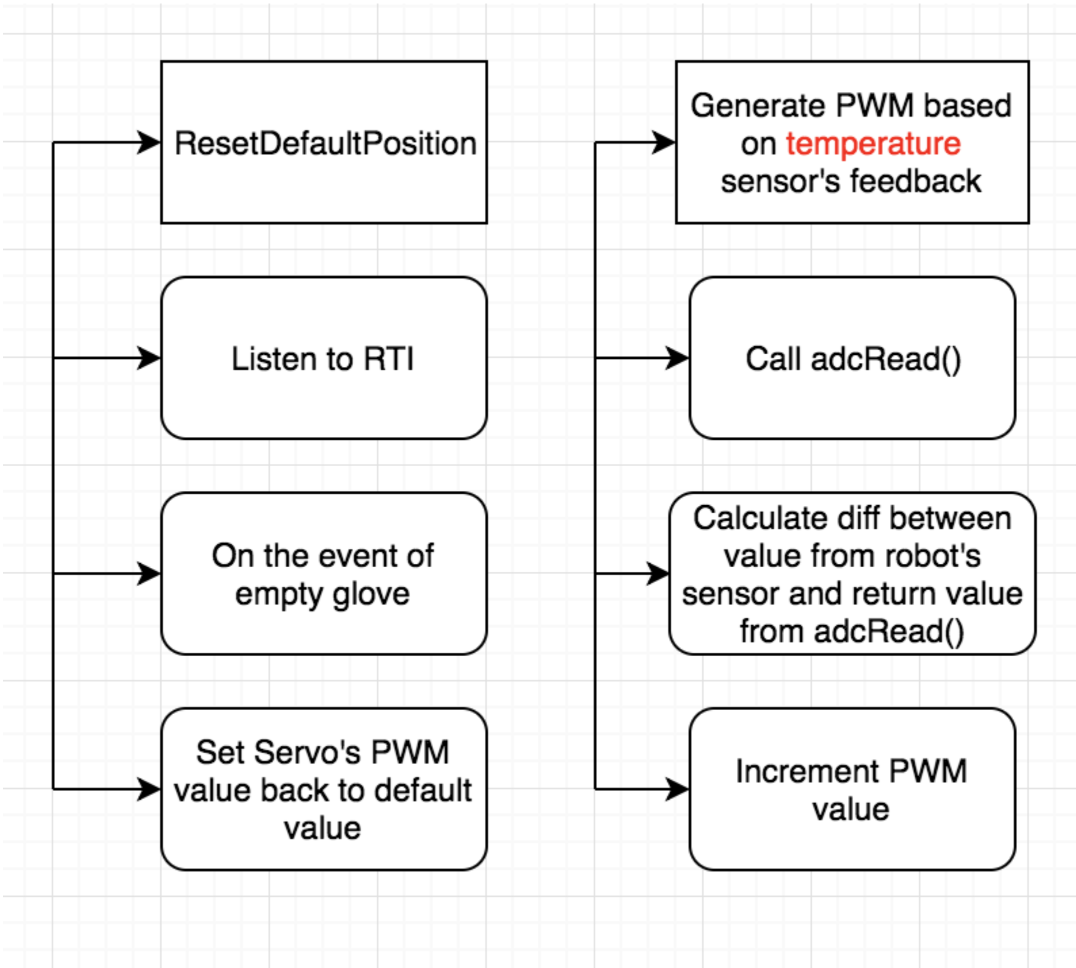

Figure 3: Each function's functionality and purposes