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 will be implemented in a state-machine-like fashion. Firstly we have an idle state and a working state. Both STM32 (on the glove and robotic hand) will follow this logic flow. For the glove's side STM32 which acts as the master, the program will collect the ADC value from sensors and potentiometer on the exoskeleton and then transfer the value to the other microcontroller through SPI while the program is in a working state. Besides, it will also request SPI from the slave to receive data about the pressure applied on robotic fingers. This way the program will know when to restrain the movement on the exoskeleton.

When the program is in an idle state, the STM32 on the glove will send a flag to the STM32 on a robotic hand to indicate that the system should all be in the idle state. The condition for glove's microcontroller to switch state is determined by constantly checking if the glove is empty. This is achieved by using Real Time Interrupt(RTI) and there are some other software components that will be using RTI. These components are using RTI to detect the changes in pressure and temperature. When the program detects the glove is empty, it will switch to the idle state and switch to working state when the glove is being used.

For the microcontroller on the robotic hand's side, the program will listen to the master's signal to transfer the temperature and pressure sensors' value to the other microcontroller when the program is in working state. Besides that, the program will control finger's motion through PWM using the ADC value from the other STM32 which is transmitted through SPI.

However, the program is designed to reset the finger's position to default setting while a flag is received from the master indicating that the whole system should be in the idle state. The interrupt will once again being used to detect the flag from the master. Thus once user removes their hand from the glove, the slave will receive a signal through an interrupt and then restore the fingers' default position. A more detailed diagram will be showcased in the appendix section.

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

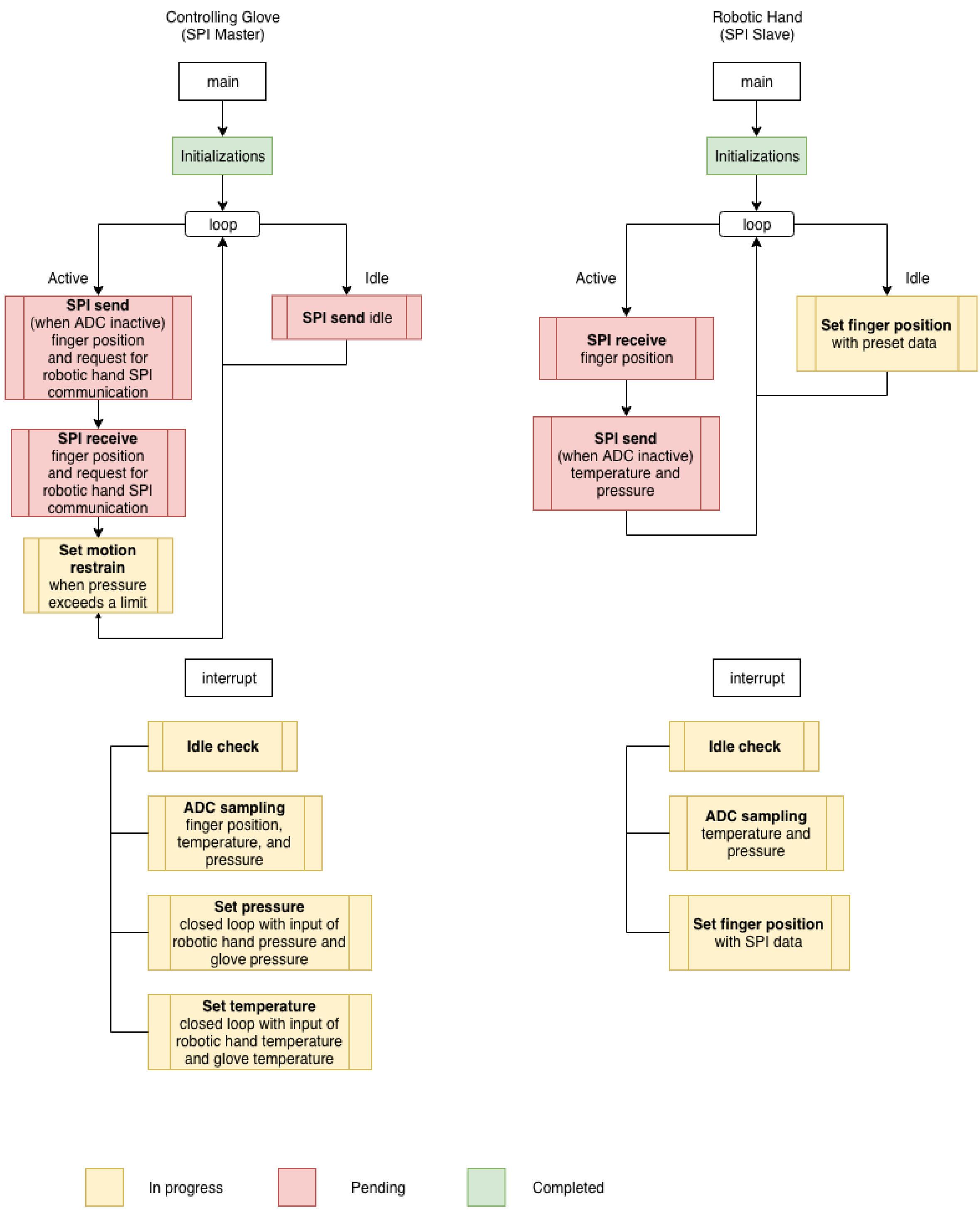
4.0 Sources Cited:

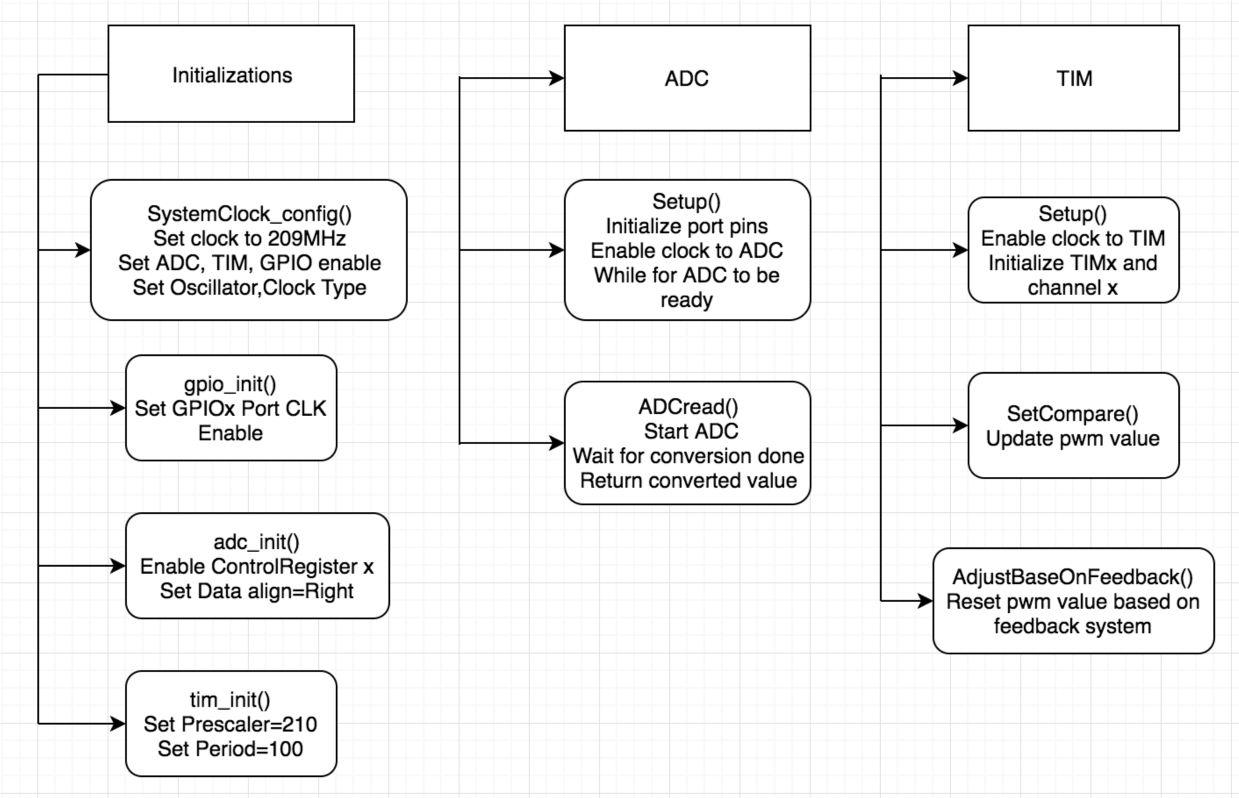
[1] D. Schaefer, "Eclipse CDT | The Eclipse Foundation", *Eclipse.org*, 2018. [Online]. Available: https://www.eclipse.org/cdt/. [Accessed: 06- Oct- 2018].
[2] “EAGLE | PCB Design Software | Autodesk", *Autodesk.com*, 2018. [Online]. Available: https://www.autodesk.com/products/eagle/overview. [Accessed: 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

Handi_glove Software Component Diagram




Initializations
SystemClock_config()
Set clock to 209MHz
Set ADC, TIM, GPIO enable
Set Oscillator,Clock Type
gpio_init()
Set GPIOx Port CLK Enable
adc_init()
Enable ControlRegister x
Set Data align=Right
tim_init()
Set Prescaler=210
Set Period=100
ADC
Setup()
Initialize port pins
Enable clock to ADC
While for ADC to be ready
ADCread()
Start ADC
Wait for conversion done
Return converted value
TIM
Setup()
Enable clock to TIM
Initialize TIMx and channel x
SetCompare()
Update pwm value
AdjustBaseOnFeedback()
Reset pwm value based on feedback system

Helper Functions