

CS CAPSTONE REQUIREMENTS DOCUMENT

NOVEMBER 2, 2017

PROJECT LOOM — DESIGN AN INTERNET OF THINGS RAPID PROTOTYPING SYSTEM.

PREPARED FOR

CHET UDELL

PREPARED BY

GROUP 36

PROJECT LOOM

WILLIAM SELBIE

TREVOR SWOPE

LUKE GOERTZEN

Abstract

This client requirements document outlines the deliverables that the Computer Science Senior Capstone Group 36 will be producing and held accountable for - with regards to client satisfaction and Capstone course evaluation. The document addresses what the end result will be expected to do and be, but not how it is implemented.

CONTENTS

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions, Acronyms, and Abbreviations	2
1.4	References	2
1.5	Overview	2
2	Overall Description	3
2.1	Product Perspective	3
2.1.1	User Interfaces	3
2.1.2	Software Interfaces	3
2.1.3	Hardware Interfaces	3
2.2	Product Functions	3
2.3	User Characteristics	4
2.4	Constraints	4
2.5	Assumptions and Dependencies	4
3	Specific Requirements	4
3.1	External Interfaces	4
3.2	Functions	5
3.3	Performance Requirements	5
3.4	Design Constraints	5
3.4.1	Reliability	5
3.4.2	Security	5
3.4.3	Maintenance	5
4	Appendix	6
4.1	Figure One: Dependencies Gantt Chart	6
4.2	Figure Two: License	6

1 INTRODUCTION

1.1 Purpose

This document aims to define the goals, scope, requirements, and development direction of the software and firmware side of Project LOOM. This document does not address in depth the requirements of the hardware portions of Project LOOM. The focus is instead only on the scope of the project with regards to the work to be done by Computer Science Capstone Group 36. Additionally, this document explains what the system will be expected to do, not how it will do so. As such, some details of specific technologies, protocols, or hardware will be deferred to subsequent documents. While this document will present the dependencies of the project, with regards of what must be completed in order for others to begin, a specific timeline will not yet be provided. This document is intended for the client of Project LOOM, Chet Udell, as well as the Senior Capstone course instructors to outline what can be expected of the Senior Capstone group to produce over the course of the project, and to provide a metric against which Group 36 will be evaluated.

1.2 Scope

With Project LOOM, we aim to create an open-source, plug-and-play, suite of modular building blocks, the extensible and easy programmability of which expands the demographic of people capable of implementing Internet of Things solutions. For users with limited technical expertise to create complex systems, we aim to build a system that abstracts out the more technical details, allowing them to focus on their system more than the implementation of the modules. The system should also be usable by higher level students and experts by allowing them to modify or write their own firmware, and create new modules. Project LOOM will be developed for university faculty demos of functionality, possible subsequent commercialization is currently outside of the scope of the Senior Capstone project.

1.3 Definitions, Acronyms, and Abbreviations

The Internet of Things (IoT): Entails an aggregate of systems (many of which are embedded) capable of reading, transmitting, receiving, processing, and acting upon data. This often takes the form of a network of remotely connected devices that can be used for the purpose of autonomous data collection, or remote control and automation of various systems.

Module: An interchangeable, self-contained device that can be connected to an IoT network with minimal configuration. Connects to other modules to combine behaviors (i.e. a sensor attached to a WiFi adapter becomes a sensor that transfers data over WiFi).

Gateway: A wired or wireless node connecting multiple devices on a network (e.g. WiFi, Ethernet)

1.4 References

- 1) **Figure One:** Gantt Chart
- 2) **Figure Two:** License

1.5 Overview

The remainder of this document will provide an overview of major functions of Project LOOM, followed by more specific information regarding these functions and constraints to their design, as well as characteristics of its users and the assumptions and dependencies that are inherent to its inception. A Gantt chart is provided to illustrate dependencies of the stages of the project.

2 OVERALL DESCRIPTION

2.1 Product Perspective

The scope of the work that Group 36 will be responsible for is a subset of that of the larger Project LOOM team. This subset includes the embedded firmware that will be run on the devices themselves, as well as the development of the graphical interface that will be used to program and monitor the connected devices.

2.1.1 User Interfaces

The main user interface will be presented via modules in MaxMSP that receive and process data in a readable way. Readability will be informally evaluated by whether or not faculty or other users are able to set up and understand the displays of program during our demos.

2.1.2 Software Interfaces

Project LOOM will interface with the following software in order to visualize data or to use the software as an output of the system:

- MaxMSP
- IFTTT
- Google Docs via PushingBox
- AdaFruit IO
- Plot.ly
- Anything else that accepts TCP/MQTT or UDP/OSC packetized data

2.1.3 Hardware Interfaces

The following hardware interfaces (not necessarily exhaustive) will be used to make the modules. The construction itself is not a part of Group 36's portion of Project LOOM:

- AdaFruit FeatherM0
- AdaFruit Feather32u4
- MPU6050

2.2 Product Functions

The system must be able to:

- Receive and transmit data with physically and wirelessly connected modules.
- Perform actions based on received data.
- The firmware should be modular and configurable via a user interface.

Users must be able to:

- Select and visualize incoming sensor data on a network via MaxMSP
- Remotely trigger events on actuators and monitor these actions via a user interface
- Re-calibrate devices remotely
- Process or analyze this data for events, thresholds, feature recognition, remap to useful control signals
- Program the devices with the Arduino development environment by altering or replicating the open-source code on which the firmware is based

2.3 User Characteristics

There are a range of possible users for the LOOM devices and software; the target audience is meant to span from novice hobbyists to experts. Anyone wanting to learn about or implement an IoT solution in a streamlined fashion should be able to pick it up out of the box and get a simple system working without any programming. On the other end of the spectrum, there are users with more technical expertise, who simply want well-designed sensors and actuators that work together in ways that make sense. Since Project LOOM is open-source, these users will be able to write their own software and firmware for the chips if they choose to do so. Project LOOM is currently being developed with the intent to be tested by university professors and their students.

2.4 Constraints

Hardware Limitations: The software must run on the following hardware: AdaFruit FeatherM0, AdaFruit Feather32u4, MPU6050i.

Parallel Operation: Sending and receiving should be possible at the same time, especially to and from any central computer.

Reliability Requirements: The system will send a notification if system has detected that a module is broken, offline, or otherwise less than functional. However, the project is not currently being held to any standards and reliability will not be quantified.

Safety and Security: Both will be addressed to a reasonable effort, but not to the point that it restricts progress of the rest of the project.

2.5 Assumptions and Dependencies

Our work will be built off of the framework of existing hardware and software of Project LOOM. We are assuming that our firmware and software will be run on functional hardware developed by a sister team of Project Loom. The firmware for the modules in Project LOOM will be developed with the Arduino development environment, and the interface for reading the data will be a plugin for MaxMSP.

3 SPECIFIC REQUIREMENTS

3.1 External Interfaces

Inputs:

- Open Sound Control packets giving instructions
- Commands from the gateway
- Requests for data
- Data from sensors

Outputs:

- Actuator behavior (servos, motors, lights, etc.)
- Open Sound Control digital/analog
- Data to the gateway

3.2 Functions

Modules will each have their own functionality but should be able to connect to the software and network of other modules without difficulty. The user should be able to use software to view data reading or to cause behaviors of actuator modules via outputs. Software can be used to translate outputs from one module into a usable input to another module, or simply process and display the data for the user. A comprehensive list of modules intended to be developed is not yet defined.

3.3 Performance Requirements

There are not currently requirements regarding the speed of the modules, the number of modules, etc. The focus is currently to get the system operational, performance improvements can be incorporated later. However, some performance related points are:

- Code has a lightweight modular structure, based on pre-compiler conditionals
- Programmable and modifiable via Arduino IDE and MaxMSP
- Live data configurable over UI

3.4 Design Constraints

Budget: Products must be low-cost relative to other off-the-shelf products on the market.

Software: Software and firmware produced must all be open source. The software must be editable via Arduino IDE to ensure it is accessible for the community of arduino-based open source user community and clientele the product is being made for.

Hardware: The firmware must function on the AdaFruit Feather 32u4 and AdaFruit Feather M0 processors, while the shields that will be supported will be determined over the duration of the project.

3.4.1 Reliability

Within the scope of Project LOOM as an open source project, it will not be held to any standards, as is dictated in the license that is provided with it. Within the scope of the Project LOOM as Group 36's Senior Capstone Project, Group 36 is expected to produce code that is robust and modular enough to be compiled onto a variety of combinations of boards and shields. As Group 36 is strictly responsible for software and firmware, the reliability of the physical product is disjoint from our responsibilities.

3.4.2 Security

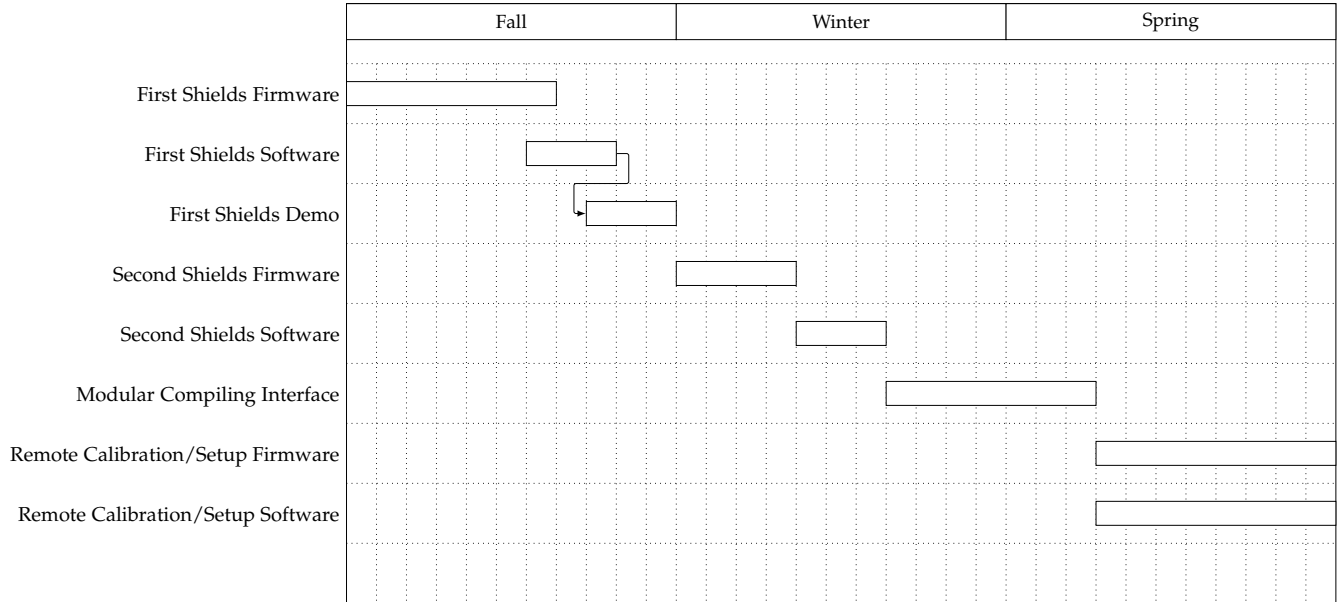
Rigorous attention to the security of the modules is not a primary focus within the scope of the Senior Capstone project - anyone with physical access to the module would be able to overwrite the behavior of the module. However, to begin with, the chips will have a hard-coded list of known connections which it will query when receiving a new connection.

3.4.3 Maintenance

As mentioned under section 3.5.1 Reliability, the interface should give an alert to the user if one of the modules that is meant to operate on the long-term is no longer transmitting its data in the expected interval. This will allow a user to troubleshoot the problem.

4 APPENDIX

4.1 Figure One: Dependencies Gantt Chart



4.2 Figure Two: License

Uses the following license and any Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.