**Chapter 2**

# System Integration

# and Modeling / Methodology

# 2.1 Introduction

The NASA Student Launch competition provides real world experience for teams of students working together to accomplish a common goal. This year’s goal is to autonomously load a payload into a rocket of our design and construction, launch to rocket with its payload to a predetermined altitude, and return the rocket and payload safely to the planet’s surface.

The portion of the project I am solely responsible for is the design and construction of the website and the programming of the Automated Ground Support Equipment (AGSE).

# 2.2 Technology

The technology involved in the project has been around for years, yet it is constantly evolving. This creates many new opportunities for innovation as well as a stable foundation to build upon.

## 2.2.1 Website Technologies

The technologies the website will use are HTTP (hypertext transport protocol), HTML (Hypertext Markup Language) PHP (PHP: Hypertext Preprocessor [1]), CSS (Cascading Style Sheets), JavaScript, and related technologies.

HTTP is the communications protocol which the webserver and web clients (web browsers) will use to communicate to display the webpages and content of the team’s website. PHP will be used to parse and prepare the HTML for display in users web browsers. HTML will be used to structure the content within the individual pages. JavaScript will be used for client side scripting. Most of the client side scripting will be for visual effects. Other technologies relevant to web development and display may be incorporated as needed.

## 2.2.2 AGSE Technologies

The technologies the AGSE will use are device trees, device tree overlays [2][3], C/C++, PCBs (Printed Circuit Boards), ICs (Integrated Circuits), solid state electronics, Electric Motors, and related technologies.

The Beaglebone black uses a generic device tree to define its components. The device tree overlay will be used to describe the components, such as the stepper motor controllers, and how the system will interact with them. C/C++ will be used to create the logic to control the motors; as well as to instruct the system to send signals to the stepper motor controllers. I am using commercially available PCBs, ICs, solid state electronics and electric motors to construct the AGSE’s control system. Other technologies relevant to embedded control systems may be incorporated as needed.

# 2.3 Components

## 2.3.1 Website Components

The website is being run on a shared hosting service. The server is running a variant of the GNU/Linux operating system. I am using the Apache webserver with the PHP module to serve the webpages. I am using the MySQL database as well as the servers file system to store the websites content, and I am using WordPress as the font end to manage and present the content. I am using \_s (under scores) as the base for my custom template to customize the look and feel of the website.

The code for the website will be edited using Notepad++ [4], a free text editor which offers syntax highlighting.

2.3.1

The AGSE will use a Beaglebone Black as the controller. There will be 3 stepper motor controllers driving the units 3 stepper motors. The stepper motor controllers will be Big Easy Driver model ROB-11876 [5]. It will have one indicator LED, to indicate the units current state (e.g. paused or running). It will also have a power switch as well as a push button to pause and un-pause the units operation.



Figure 1: Beaglebone Black

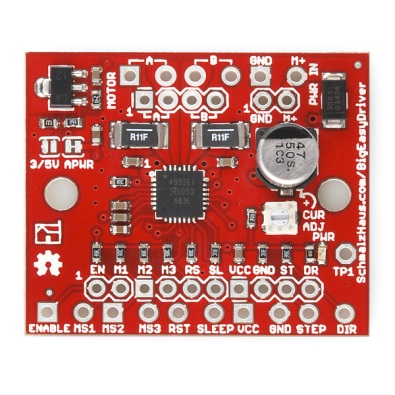


Figure 2: ROB-11876

The code for the AGSE will be edited in Eclipse, and cross compiled use GCC before the binaries are placed in the Beaglebone Black via a USB connection.

# 2.4 Initial Design

## 2.4.1 Website – Initial Design

The website will consist of 5 main section:

1. Background
2. Header
3. Menu
4. Content Area
5. Footer

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Header |  |
|  | Menu |  |
|  | Content Area |  |
|  | Footer |  |
|  |  |  |
|  |  |  |

Figure 3: Website page layout

The background will be consistent across all pages.

The Header will provide basic information about the rocketry team, and will be consistent across all pages.

The Menu will provide the main navigational links of the site, and will be consistent across all pages.

The Content Area will contain the information relevant to a specific page.

The footer will contain information about our sponsors, and will be consistent across all pages.

The web design for the website will be separated into a number of stages or stories.

* DNS Configuration
* WordPress Installation
* Template – Create the basic structure of the pages
* Template – Customize the look and feel of the pages
* Template – Add special features (parallax scrolling star field)
* User training: WordPress content management

Writing the logic to use the AGSE to acquire and deposit a payload into the rocket.

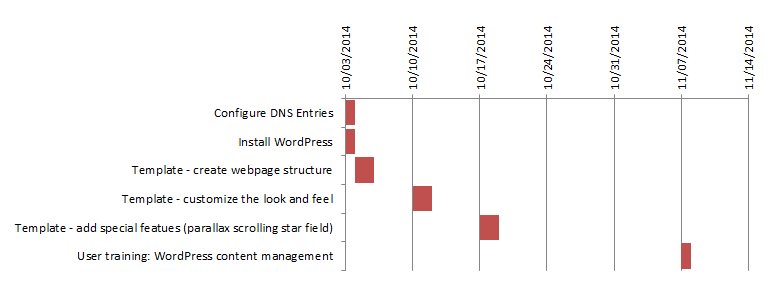


Figure 4: Website gantt chart

## 2.4.2 AGSE – Initial Code Design

The code design for the AGSE will be separated into a number of stages or stories, and each stage will be researched and developed independently.

* Device Tree and Device Tree Overlay
* Reading from and writing to the GPIO
* Controlling an external LED
* Communicating with the stepper controllers
* Creating the Controller Classes for the stepper motors.
* Writing the logic to use the AGSE to acquire and deposit a payload into the rocket.

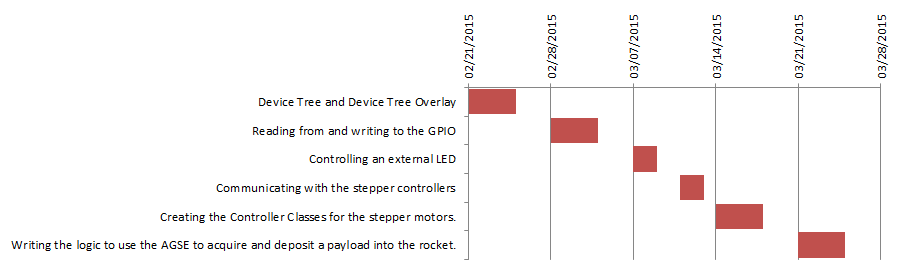


Figure 5: AGSE software development gantt chart

# 2.5 Agile Code Development

The code for the project is being subdivided into smaller manageable sections of which measurable deliverables can be created. During development, there will be validation tests created for each section, and unit testing will be performed throughout the process. As portions are completed, they will be presented to the UAA Rocketry Team for user acceptance testing. If a section does not pass the user acceptance testing, it will be re-worked, re-tested, and then acceptance tested again. I will be using this process to control the quality of the code being written for the project.

# Appendix A: References

[1] PHP, http://php.net/manual/en/faq.general.php#faq.general.acronym, retrieved 2/23/2015.

[2] Adafruit, https://learn.adafruit.com/introduction-to-the-beaglebone-black-device-tree/device-tree-overlays, retrieved 2/23/2015

[3] FDTWiki, http://devicetree.org/Device\_Tree\_Usage, retrieved 2/23/2015

[4] Notepade++, http://notepad-plus-plus.org/, retrieved 2/23/2015.

[5] Sparkfun, https://www.sparkfun.com/products/retired/11876, retrieved 2/23/2015

# Appendix B: Code and Documentation

WordPress theme code:

https://github.com/jadempsey/UAARocketry/tree/master/WP%20rocketry%20theme

Doxygen Documentation:

https://github.com/jadempsey/UAARocketry/tree/master/doxygen