Slang Embedded Systems Interfacing  
CIS 598 – Computer Science Project  
Initial Writeup and Feature List

Your Name: Jacob Legg

Your Advisor: Dr. Hatcliff

# Overview

The goal of this project is to provide a system to write code for embedded systems in a high level, type safe, and semantically verifiable environment. This can be done in two ways. The first way this can be accomplished is by sending Firmata commands over serial ports to be translated by a microcontroller using a command translator. This way will involve using Firmata4j for the sole purpose of using its build in serial port connection code and Slang for code development and system design. The second way is to transpile the Slang code into C to be run directly on a microcontroller. The second approach for this project is not the focus of the project, but it will influence certain design decisions to allow for easier implementation later.

The main purpose of the tool for this project is that of an educational tool. This means the other section of this project will be to standardize the tool to be easily usable in a classroom setting by providing documentation for the tool and by making the library as easily usable as possible. The tool should also leave a specific interface setup so teachers can plug in different backends that can be used to demonstrate code functionality (e.g., arduinos, simulations, state machines, etc.). The design of the tool will focus on simple syntax with complex functionality for education and for industry.

## Problem/Solution

The problem I am trying to solve is that writing code for embedded systems can be unsafe in a programmatic sense. I plan on solving the problem by moving control over an embedded system to a higher-level language that can interact with microcontrollers using standard format Firmata commands that can be translated by a microcontroller into onboard instructions, by using Slang to provide semantic verification using Logika, type safety using Tipe, and safe code outlines for components using HAMR.

## Algorithmic Functionality

My project is algorithmically complex since it depends on the use of multiple languages, serial port translation, and semantic contracts to provide control over microcontrollers using higher-level languages.

## Qualifications

I am qualified to work on this project since I have worked with the SANTOS Lab, the Slang language, and the SIREUM tool library for 8 months now. This means that I am aware of the tools that are available to me and how to use and comprehend the tools functionality. I am also qualified to work on this project since I have taken ECE 241 and CIS 450, so I know how to write code for and use embedded systems.

# Feature Lists

## Minimum Viable Product (MVP)

* The project will have low-level GPIO interfaces that can be used to interface with core fermata libraries
* The project will allow for the board to be turned on and off and setup port connections in slang.
* I will be able to use the basic control flow and GPIO classes to control basic examples of embedded systems using pre-generated HAMR examples to provide basic code outlines.

## Version 1.0

* This project will have Slang libraries on top of the bade GPIO interfaces to provide abstraction way from common devices
* I will clean the code to be more usable by students and teachers in an educational setting.
* I will write basic documentation for usage of the tools to explain the usage of each of the individual tools and how to set up the framework using with preexisting HAMR generation.
* I will create example hardware to demonstrate usage for the tool.

## Version 2.0

* The project will allow for the component controlling code to be pushed onto the microcontroller using C transpolation from Slang Embedded so the code does not have to run on JVM.
* The project will provide a backend of C libraries to be used instead of fermata commands to control components.