Term Project Proposal

Smart Thermostat Control Programming Language

Johnathan Picek

Lewis University, College of Engineering, Computing, and Math

CPSC 46000-003

Dr. Eric Chou

March 20th, 2022

Project Idea and Rationale

Nowadays, embedded systems can consist of small computers like Arduino, Raspberry Pi, or even have a built-in SoC (System on a Chip) capable of running C++, Python, Java, and more. This makes almost any device incredibly capable and powerful, but these capabilities also present a problem as they may open systems up to potential security vulnerabilities by increasing the available attack surface.

The examples provided in Week 8 Office Hours were exactly the spark I needed to get excited about a project. Recently, my family has remodeled our kitchen with new appliances including microwave, oven, a new refrigerator, and new thermostat control. These newer devices have controls that are significantly more complicated than those of our old models, so I have recently read many instruction manuals to help my parents learn to operate the new devices.

In my project, I intend to mock-up and prototype a streamlined and lean programming language specific to a Smart Thermostat control in line with one of your suggestions. To limit attack surface and therefore contribute to enhanced device security, the final implementation would ideally be isolated entirely from other languages and operate at a low enough level to avoid having the system run a separate interpreter or VM. Programming and operation would be done through the limited number of buttons available to the user, specifically:

* Up and Down arrow or + and – Buttons for manual adjustments
* Menu and Enter/Confirm Buttons
* no more than 2-3 auxiliary keys to assist more complex operations

Expected Project Deliverables

1. **Design Document of the Language** – the equivalent of the public documentation information for languages like Java, Python, etc. but for the embedded system’s language instead. This would include detailed information on the various functions
2. **Prototype** – An interpreter prototype of the embedded system’s custom programming language operating in Python, showing operation of all designed functions, based on a small number of recognized keywords (estimated 5 - 10)
3. **Hardware Controls Map** – The hardware control map relates the software functions to the physical hardware buttons. It will include a mock-up diagram of a device control along with instructions for each command, and which language keywords and functions those relate to. A version of this would eventually be included in an actual device’s user manual (likely without the programming language details).
4. **Presentation –** A PowerPoint presentation informing on the results, and successes and challenges faced during the project, which would accompany the Report
5. **Technical Report –** a written Report informing on the results, successes, and challenges faced on the project

Tools and Materials

I expect to use:

* Python to code the Prototype
* PowerPoint for the Presentation
* Vizio and/or draw.io for the diagrams and Hardware Control Map
* Word for the Technical Report and the Language’s Design Document

Schedule

With 6 Weeks remaining in the class, the following Sundays remain, so my preliminary schedule is as follows:

**3/27** 4/3 4/10 **4/17 4/24** **5/1**

HCM . . . work on DD and P . . . DD and P All Done Submit

There are 5 deliverables, with the Design Document and Prototype likely to be the most elaborate pieces. As such, I will begin by using the Hardware Controls Map to brainstorm the necessary functions of the device and begin to populate the keyword list necessary for the design document and prototype. The Map should be completed by 3/27.

The Design Document and Prototype must go hand in hand. They will also require the most significant amount of time spent. These should be progressing steadily through 4/3, 4/10, and either complete or nearly complete by 4/17 with only slight adjustments remaining.

The Presentation and Technical Report will follow and should be completed by 4/24, leaving the final week necessary as a buffer to handle any unexpected difficulties or complications that arise, as they tend to do.