Northern Arizona University

Team Radio Pi

Capstone Team	*Project Sponsor*	*Mentor*
Tyler Plihcik	**General Dynamics**	David Failing
Brandon Click	**Mission Systems**	
Jefferey Williamson	Randy Derr	
Kaelen Carling		
Saurabh Jena		

Prototype Plan

Version 1.0

November 6th, 2020

Table of Contents

Introduction

The world runs on multitudes of various IoT (Internet of Things) devices. These devices are anything from smart home devices, to wifi extenders, speakers, even cars and kitchen appliances. All these devices communicate via radio waves and run on what are called embedded chipsets, which are inherently small, in both footprint and power consumption. As our technology continues to advance, the need for smaller embedded systems becomes more prevalent each hour.

For nearly three decades software defined radios, or SDRs, have been revolutionizing the way IoT devices communicate. Leading the charge into this new era of devices and communication platforms is our project sponsor, General Dynamics Mission Systems. The need for newer, smaller, more efficient and more powerful systems is dire. Current solutions are often times built on legacy software, and although they still work, are definitively not the best solution given our technological advances since the late twentieth century. Because of this, Software Defined Radios and their current command and control systems are prone to:

- Failure
- System crashes
- Memory overloads
- General decay

Our envisioned solution for this rising issue is to completely redesign from the ground up, using modern technology in both hardware and software, the way we interface with SDRs today. We will be simulating this solution on a Raspberry Pi, a very small yet powerful, computer. We will establish a database and a web interface, connect the two, implementing these in the most efficient language(s) available, including networking protocols and database languages. We will then ensure that the end user has complete and total control, given they have the proper authorization, over the system. From querying the database, collecting, sorting, and storing data, controlling permissions and viewing system diagnostics, our envisioned solution will attempt to right all the wrongs of current command and control interfaces for embedded systems.

The following contents will serve as the plan presented to the client, General Dynamics Mission Systems, for Team Radio Pi’s solution prototype.

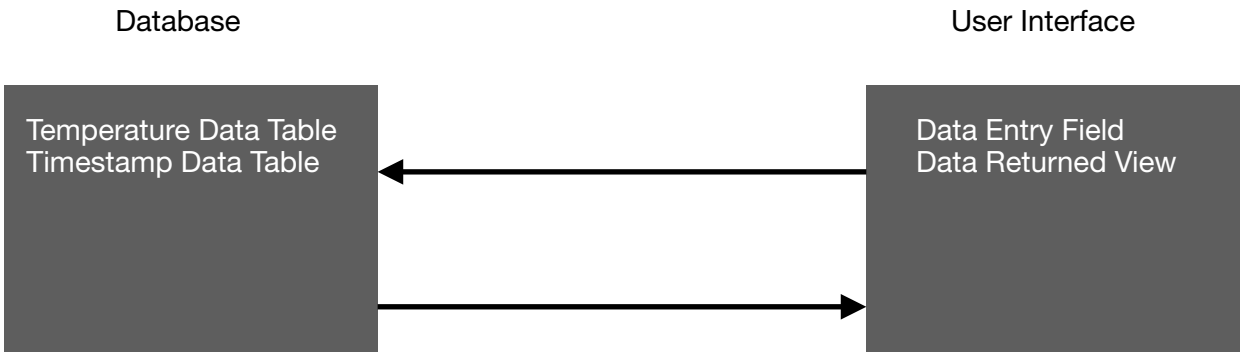
Prototype Plan

In this final phase preceding actual implementation, we plan to prototype the following, which will result in both validation of decisions made in our technological feasibility research, and client validation of our envisioned solution.

Our prototype will contain:

- A functioning database
  - Database will support two data types:
    - Mock temperature data
    - Time stamp data associated with the aforementioned temperature data
- A barebones interface to send and receive data from that database

During the creation of this database our primary focus will be on its resource consumption. This will serve as the proof that SQLite can function as the database to be used in an embedded product. We chose to use the temperature due to the fact that it is likely one of the more common data types to be transmitted from device to database, according to information provided to us by the client. Additionally, we have chosen timestamp metadata to go along with the temperature data because all data entries will have such data associated with them. The diagram below details visually our prototype design, containing a database and user interface. The user interface will contain a data input field, as well as a view for returned data. The database will hold two tables for our two selected datatypes to be used in this prototype, temperature data, and its associated timestamp data.



Conclusion

As technology continues to evolve, and the needs of everyday embedded systems continue to become more complex, and especially as SDRs become more prevalent we must strive to make our embedded systems run with less power consumption, require less memory, and efficiently and effectively execute the functions it was designed for.

General Dynamic's current command and control system for their embedded systems, and specifically their SDRs, is outdated, inefficient, and has been stitched together over the years. Our solution will offer a proof of concept for a ground up redesign of this command and control system, simulating GD hardware with a Raspberry Pi 3 B+.

Throughout the duration of this document we briefly discussed the deficiencies of the current solution, and offer our plan for the prototype. This prototype will both validate the choices we made previously during our technological feasibility research, as well as provide client validation for our envisioned solution.