EMC Project Report

Callum O'Brien

October 21, 2015

1 Overview of Project

Luke Bowden and I are creating a system by which students will be able to check books in and out of the EMS library in an easier and cooler way; using NFC tags in the books and in our already existing student ID cards. The end result of this will be an open hardware, open source system that will (hopefully) actually be used, probably implemented as a daemon running on a Raspberry Pi, as the school already has a few of those lying around. Our objectives are:

1. To allow students to check books in and out of the library easily and conveniently,
2. Create output when a student attempts to check out a reference book informing them that that book should not be removed from the library,
3. Provide a log of all check ins and check outs of books by students,
4. Provide a interface by which people can check if a book is in the library or not,
5. Do all of the above using only free/libre and open source software.

2 Progress Made

So far, we have mostly been researching different competing standards for NFC. There are quite a few of these. We have decided upon the OpenNFC standard, as it provides the best resources regarding interfacing chips with Linux including a full, open source software stack and excellent documentation. Other standards may require proprietary drivers, which would void the fifth objective of this project on the list above.

We have also done some low-level analysis of the problem, identifying inefficiencies in the current system used in the library and outlining how to overcome these, as well as beginning to consider the design of the proposed system. At time of writing, our code is currently limited to a generic shell of a daemon in C; for a completely up-to-date view of things, see our GitHub repository[1].

[1] `https://github.com/henceforthingly/emslibrary`

3 Future Work & Direction

The proposed system is yet to be fully designed; this is currently the item of highest priority as it will be much easier to implement the system with a clear idea of its structure. After the design is finished, we can fully devote our time to implementing and testing the proposed system; testing this will involve manual tests using actual hardware and unit tests using hardware input emulation to rapidly and automatically test any new code. Upon finishing this, we will install the system in the library and will then be finished.