**Lock and Key Group Final Presentation**

Alan Ngo

John Vetus

Nicolas Lowman

**Project Overview** 

Problem: Incorrectly slotted keys and foreign metal objects may cause shorts and misreads. In the current design when one slot shorts the entire drawer becomes unusable. This is a significant issue that makes up approximately one third of their daily support calls.

**Project Overview** 

Problem: Incorrectly slotted keys and foreign metal objects may cause shorts and misreads. In the current design when one slot shorts the entire drawer becomes unusable. This is a significant issue that makes up approximately one third of their daily support calls.

**Proposed Solution**

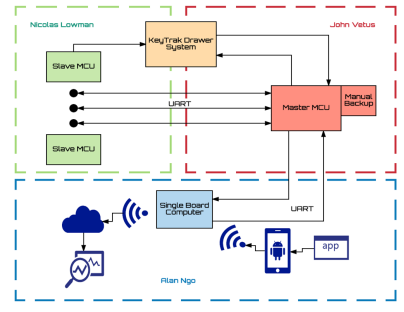
● Solve the shorting issue with the use of microcontrollers to isolate each iButton read line

● Upgrade current communication protocol (LPT) with more modern ones (USB)

● Implement backwards compatibility for legacy devices

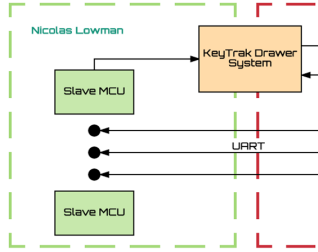
● Introduce new wireless communications and features for “next generation” key drawer

**System Overview**

****

**Slave MCU Subsystem Overview**

**403 Deliverables**

● Read iButton 

● Store iButton Serial

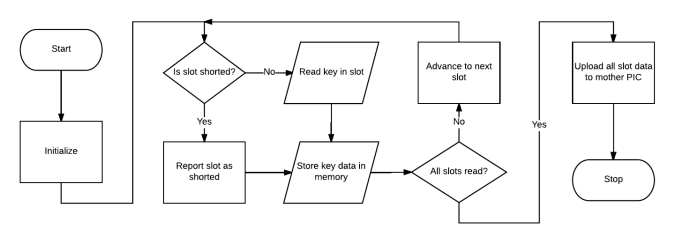
Number

● Solenoid drive system

● Power supply of entire

system

**Slave MCU Subsystem Flowchart**

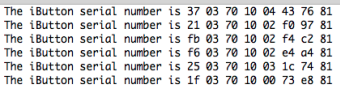
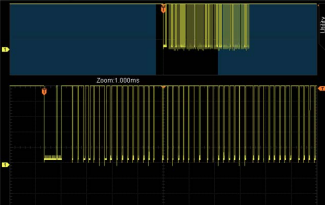
****

**Slave MCU Challenges**

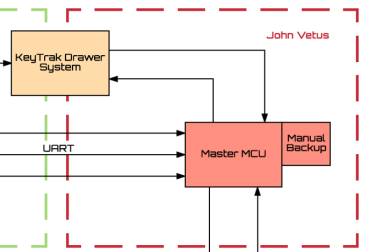
● ICD (In Circuit Debugger) no debugging operation with initial IDE software release

● Reading ibuttons

● Learning the IDE

**Slave MCU Subsystem Test/Validation Results**

**Master MCU Subsystem Overview**

**403 Deliverables** 

● Establish communication

protocols between MCUs

and SBC through UART

● Create software to

organize and track

iButton data

● Receive data from SBC

to operate drawer and

indicate iButton slots

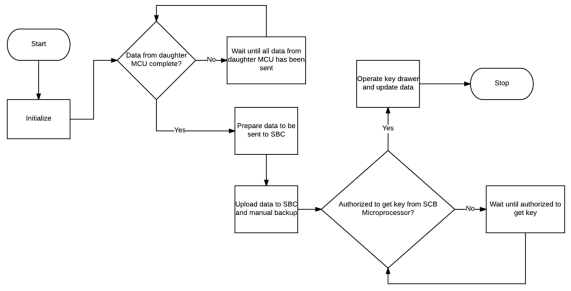
● Create a manual backup

file that has iButton

information and a timelog

of operation

**Master MCU Subsystem Flowchart**

**Challenges** 

● ICD (Debugger) was not functional until

patched in November

● Establishing the correct Baud rate between

each MCU

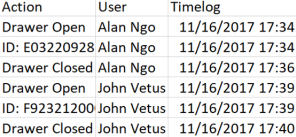
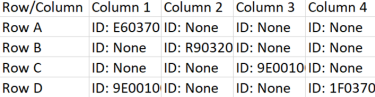
**Master MCU Subsystem Test/Validation Results **

Figure 1. Table showing where iButtons are located on the drawer (Partial screenshot)

Figure 2. Table showing timelog of user activity

**Master MCU Subsystem Test/Validation Results**

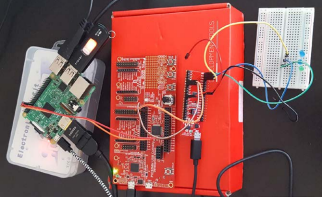
****

Figure 3. Master MCU connected to slave MCU and SBC

| Test | Actual | Met |
| --- | --- | --- |
| Validate  communication  between Master MCU and Slave MCU | Can read/write up to 16 bytes of data at a time through UART |  |
| Validate  communication  between Master MCU and SBC | Can read/write up to 16 bytes of data at a time through UART |  |
| Drawer operation and and key  retrieval/deposit | Can send a signal to operate drawer and flash an LED for  visualization of  where to take/place key |  |

**Single Board Computer Subsystem Overview**

**403 Deliverables**

● Create a cloud backup 

system for the status of

the keys and drawer

● Create an interface to

allow the user to open

the drawer from an

android phone

● Create a website to view

relevant information

about the drawer and

keys

**Challenges**

● Learning web dev and database manipulation

**Single Board Computer Subsystem Flowchart**

**Single Board Computer Test/Validation Results **

Figure 1: Terminal showing a key being added to the database.

Figure 2: Main project web page.

**Single Board Computer Test/Validation Results**

Figure 4: Graph showing how many times a 

certain person has taken keys.

Figure 3: Web page showing current keys in the 

database/drawer.

**Single Board Compute Test/Validation Results**

Figure 5: Successful drawer open signal sent to Raspberry Pi.

Figure 6: App 

interface for drawer control.

**Execution Plan Conclusion**

● Slave MCU read and store iButton Serial numbers ● Master MCU accepts serial numbers over UART ● Master communicates with SBC over UART

● Website with database created to indicate iButtons in drawer

**404 Goals**

● Fabricate full size drawer PCB

● Validate and test communication between subsystems ● Validate and test overall system performance

**Questions?**