

NFC Windows Login System

Kyle Badskey



Instructor: Paul Lin
Advisors: Hongli Luo
Harold Broberg
Date: April 29, 2016

Presentation Outline



- | | |
|------------------------|---------------------|
| ❧ Executive Summary | ❧ Keyboard Commands |
| ❧ Problem Statement | ❧ How to Use |
| ❧ Project Requirements | ❧ Evaluation |
| ❧ Project Code | ❧ Conclusion |

Executive Summary



- ❧ My project's goal was to allow a user to have an NFC chip with a unique code attached to it. This unique code would be linked with a set of keyboard commands that would sign in the user automatically thus making logging into a computer system a lot faster and easier without having to memorize any passwords

Executive Summary Cont.



- ❧ Four major operating requirements
 - ❧ Have a pre registered Near Field Communication (NFC) tag that is assigned to a given user
 - ❧ Detect the NFC tag by using an NFC tag reader
 - ❧ Authenticate the tag using an Arduino Leonardo with pre programmed user tags registered
 - ❧ Access the password prompt, enter the user's password, and hit 'Enter' to grant the user access to their account.

Overview



4

Problem Statement

- ❧ In 2012, a company called Janrain conducted a study about how many passwords the average American adult had to remember to login to online profiles.
- ❧ The study found that 58% of American adults they asked had to remember 5 or more passwords.
- ❧ 30% of those adults had to remember ten or more passwords.

5

The Solution



- ✧ Cut down on what we have to remember with material objects.
- ✧ A card with a unique code that you don't have to remember is a lot easier to work with than a password that could have to change every other month.
- ✧ NFCs are becoming a more common thing being used in the world around us.
- ✧ Example: Mobile Payments

6

Goals of the Project

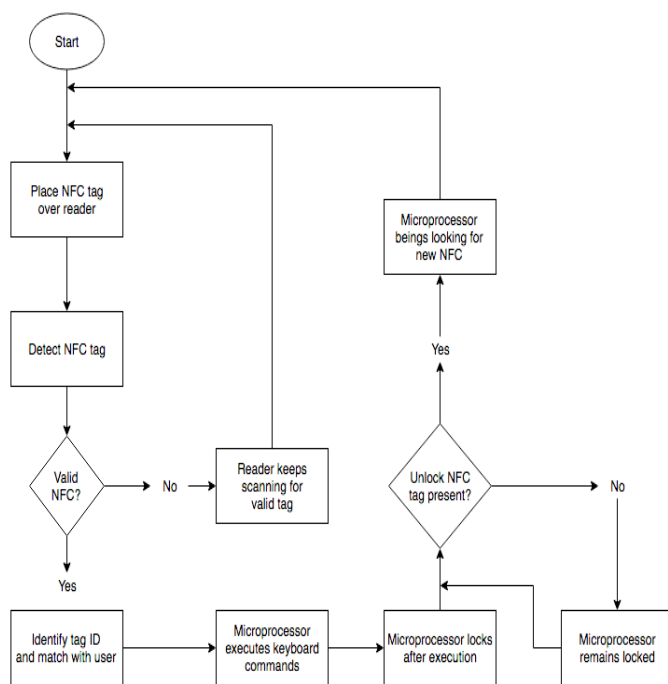


Requirement Type	Requirement (Shall or Should statements)	Verification Method	Date Verifie	Verification Report
Operational	System shall login users without use of a keyboard interface	Demonstration		
Operational	System shall keep out unwanted users from accessing unauthoized computers	Test		
Functional	System shall identify users with unique tags	Demonstration		
Performance	System shall log users in within 10 seconds of detection	Test		
Performance	System shall detect wireless keys within 3 inches of receiver	Test		
Physical	System shall be portable for the user	Inspection		
Physical	System shall use a Micro USB connector	Inspection		
Performance	System shall store at least 20 users	Demonstration		
Enviornmental	System shall have a master database with a master key	Inspection		

7

Operational Flowchart

8



Key Goal 1



- ☞ System shall login users without the use of a keyboard interface.
- ☞ Main objective of the system
- ☞ Eliminate the need for having to memorize password

9

Key Goal 2



- ❧ System shall log users in within 10 seconds of detection
- ❧ Must be fast to save time and be efficient
- ❧ A user will find system annoying if slow

10

What's Needed?



- ❧ Arduino Leonardo
- ❧ The Leonardo model is best mostly for this for its keyboard and mouse emulation.



11

What's Needed? Cont.



- ☞ Arduino NFC/RFID Shield

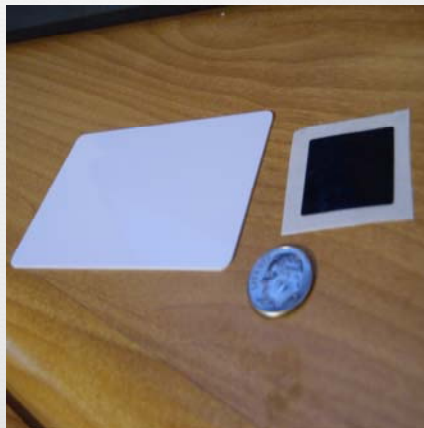


12

What's Needed? Cont.



- ☞ NFC Tags
- ☞ Ones shown hold
 - 1K bytes of data
 - 144 bytes of data



13

What's Needed? Cont.

- ☞ A computer running Arduino's software version 1.6.8 or above along with the code.



```

Login NFC | Arduino 1.6.8

// Project Name: "NFC Windows Login System"
// Date: April 24, 2016
// Prepared For: Professor Paul 2nd Year CSE Dept. of Electrical & Computer Engineering Technology

// This program emulates a keyboard and mouse emulation to log into a computer system
// for a user using an NFC (Near Field Communication) reader and NFC tag.

// Operational Description
// 1. Place a valid NFC tag with 3 bytes of the reader.
// 2. When the reader identifies the user, the card will be switched to a user.
// 3. When the match is made, the software emulation will commence logging the user.
// 4. Once the user is logged on they may perform what they need to do and may log out.
// 5. at the end
// 6. Whenever the beginning of a new session the result tag must be presented on the result
// 7. Button must be pushed or else the next tag will not be read.

// =====

#include <Arduino.h> //allows communication with I2C device
//https://www.arduino.cc/en/reference/links
#include <Arduino_NFC.h> //Shield library
#include <Keyboard.h> //Supports keyboard emulation library
#include <Mouse.h> //Supports mouse emulation library
#define DIG 6 // Sets DIG to pin 6
#define RESET 8 //Defines reset

Arduino_NFC nfc(DIG, RESET); //sets up Arduino Shield V2.0

int Round = 0; //Integer variable for Lock Feature
//Lock Feature allows the NFC tag to be held over the device or placed for long durations so
//movement must be over duration

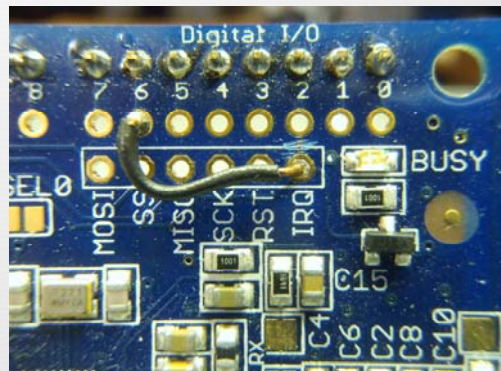
// =====

// =====
Sketch uses 10,408 bytes (30%) of program storage space. Maximum is 32,767 bytes.
Global variables use 99 bytes (0%) of dynamic memory, leaving 1,901 bytes for local variables. Maximum is 2,048 bytes.
  
```

14

Board Modification

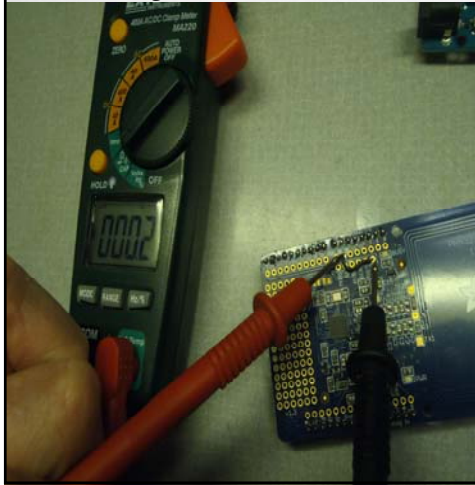
- ☞ In order for the IRQ to talk with the Leonardo the I/O needs rerouted to digital I/O 6.



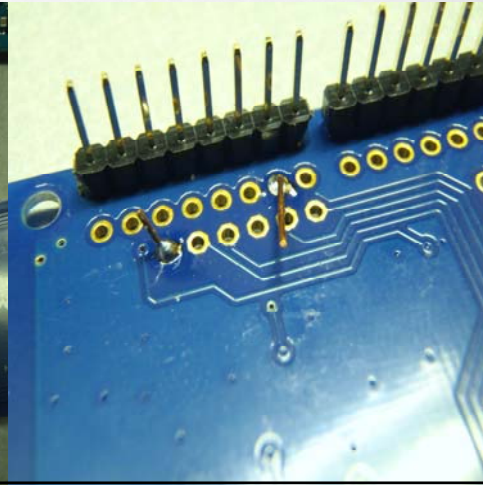
15

Board Modification Cont.

Checking for solid connection



Underside of NFC Board
(Wires have been trimmed)

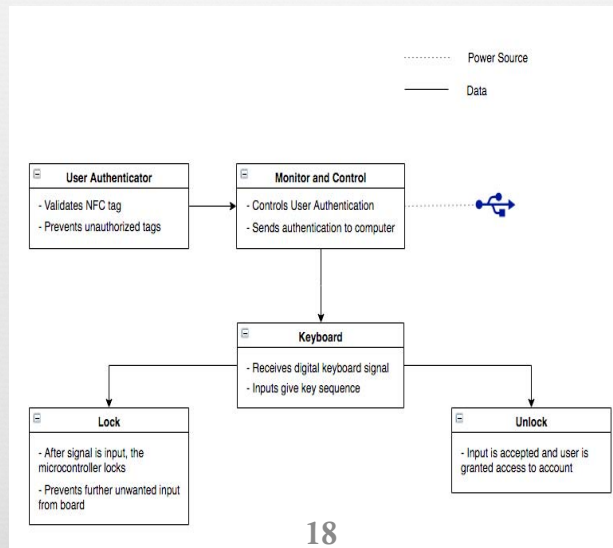


How to Use



1. A user places a valid NFC tag within four inches of the NFC Shield that is connected to the Arduino Leonardo
2. The shield sends the tag ID to the Leonardo and if it matches an ID, the password code is sent to the attached computer.
3. The computer takes the code coming from the Leonardo through the USB and reads it as ASCII character inputs and executes those key commands.
4. The user will then be logged into the system within 10 seconds (pending speed of the computer or connected network)
5. The NFC shield will then lock so no further input is brought in by the tag.

Functional Block Diagram



Project Code (Libraries)



```
#include <Wire.h> //allows communication with I2C
device

// (https://www.arduino.cc/en/reference/wire)

#include <Adafruit_NFCShield_I2C.h> //Shield Library

#include <Keyboard.h> //imports keyboard emulation
library

#include <Mouse.h> //imports mouse emulation library
```

19

Project Code (Setup)



```
void setup() {
  // set up Serial 9600 bps
  Serial.begin(9600);
  // find Adafruit RFID/NFC shield
  nfc.begin();
  // configure board to read NFC tags
  nfc.SAMConfig();
  Keyboard.begin(); //initiate the Keyboard
}
```

20

Project Code (Variables)



```
✎ unsigned digit = 0; //set value of digit to zero temporarily
✎ int found = 0; //Integer variable for lock feature
✎ int user = 0; //Integer variable identifying which ID was last used.
✎ uint8_t identify; //Will hold ID of the NFC tag
✎ uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 }; // Buffer to store the returned UID
✎ uint8_t uidLength; // Length of the UID (4 or 7 bytes depending on
  ISO14443A card type)
✎ identify = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
  &uidLength);
```

21

Project Code (Main Loop)



```

❧ void loop() { //Main loop function
❧   uint8_t identify; //Will hold ID of the NFC tag
❧   uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 }; // Buffer to store
    the returned UID
❧   uint8_t uidLength; // Length of the UID
❧   Serial.println("Waiting for valid NFC tag!"); //Prints
    on prompt "Waiting for valid NFC tag!"

```

22

Project Code (Main Loop Cont.)



```

❧ identify =
    nfc.readPassiveTargetID(PN532_MIFARE_ISO14443
        A, uid, &uidLength);
❧ uint32_t cardID = 0; //Will be used to print out the
    card number that will be used as ID

```

23

Project Code (Main Loop Cont.)



```

❧ if (identify) { //activates when identity has information, aka an
  Serial.print("NFC tag #");
  //converts the UID into a variable number
  cardID = uid[3];
  cardID <=< 8; cardID |= uid[2];
  cardID <=< 8; cardID |= uid[1];
  cardID <=< 8; cardID |= uid[0];
  Serial.println(cardID); //Prints the new cardID variable
  Serial.print(" detected"); //Prints the word "detected" at the end
  to complete the prompt

```

24

Keyboard Commands



```

❧ //This code can be duplicated and manipulated for about
  140 users, pending on password lengths
❧ //if (found == 0){ Constant
❧ //if (cardID == 34445572) { change the value of cardID
  to card value given by Arduino Serial Monitor
❧ // if (user == #) Will use to determine where to put
  selection
❧ // Mouse.click(MOUSE_LEFT); Use this for Windows
  10
❧ // Keyboard.press(KEY_LEFT_CTRL); Use this for
  Windows 7
❧ // Keyboard.press(KEY_LEFT_ALT);
❧ // Keyboard.press(KEY_DELETE);

```

25

Keyboard Commands Cont.

```

❧ // delay(1500); Constant
❧ // Keyboard.write('p'); Change the letters in the apostrophes
  to your password
❧ // Keyboard.write('a');
❧ // Keyboard.write('s');
❧ // Keyboard.write('s');
❧ // Keyboard.write('w');
❧ // Keyboard.write('o');
❧ // Keyboard.write('r');
❧ // Keyboard.write('d');
❧ // Keyboard.write(KEY_RETURN); Have this at the end
  after all characters have been entered
❧ // found = 1; Constant
❧ // user = #; # = User ID

```

26

Evaluation

- ❧ From what I wanted to achieve in the beginning on my original requirements table, my key design requirements have been tested and verified.

27

Requirement Results



Requirement (Shall or Should statements)	Verification Method	Date Verified	Verification Report
System shall login users without use of a keyboard interface	Demonstration	20-Apr-16	Passed
System shall keep out unwanted users from accessing unauthorized computers	Test	20-Apr-16	Passed
System shall identify users with unique tags	Demonstration	10-Apr-16	Passed
System shall log users in within 10 seconds of detection	Test	20-Apr-16	Passed
System shall detect wireless keys within 3 inches of receiver	Test	10-Apr-16	Passed
System shall be portable for the user	Inspection	15-Jan-16	Passed
System shall use a Micro USB connector	Inspection	15-Jan-16	Passed
System shall store at least 20 users	Demonstration	20-Apr-16	Passed
System shall have a master database with a master key	Inspection	20-Apr-16	Passed

28

Conclusion



- ☞ This project was a success and I was able to create a way for users to be able to login to a computer system without having to use a keyboard at all or have to memorize a password.
- ☞ From this project I learned how to take an idea, develop the idea, and execute the idea to make the idea an actual product.

29



Any Questions?

30



Demo

31