Wireless RFID Locking System

CS122A: Fall 2018

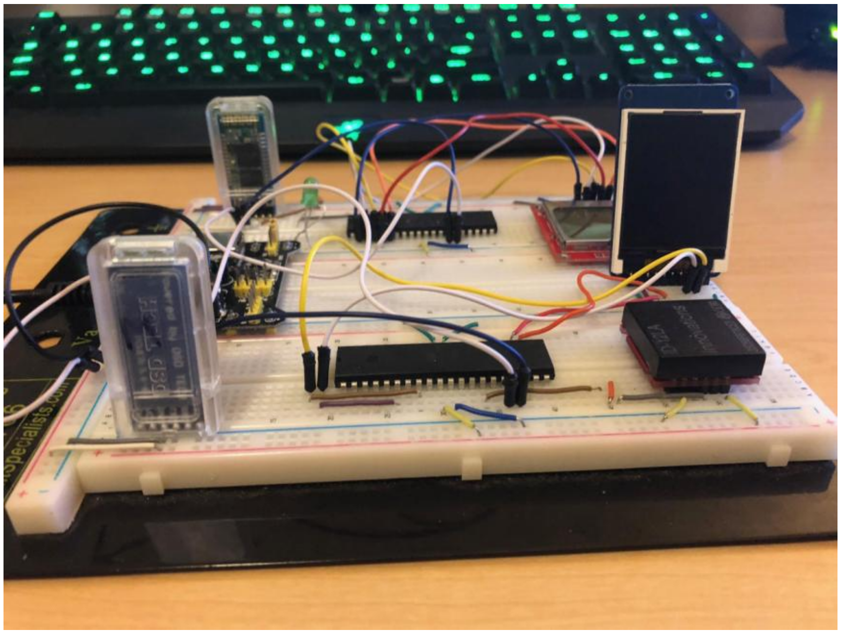
Thomas Shiels

Table of Contents

Introduction

My project simulates a wireless locking system that utilizes bluetooth and RFID technology. The concept may be extended to any secure system that requires secure remote access. The project also utilizes two LCD screens to display relevant information.

The system initializes in a locked and disconnected state and waits for the user to scan his or her ID tag with the RFID reader. The system then validates the tag on the first ATmega and displays the ID tag and whether or not it is a valid card. Finally, a signal is sent to the other ATmega over bluetooth either to unlock or lock the remote system.



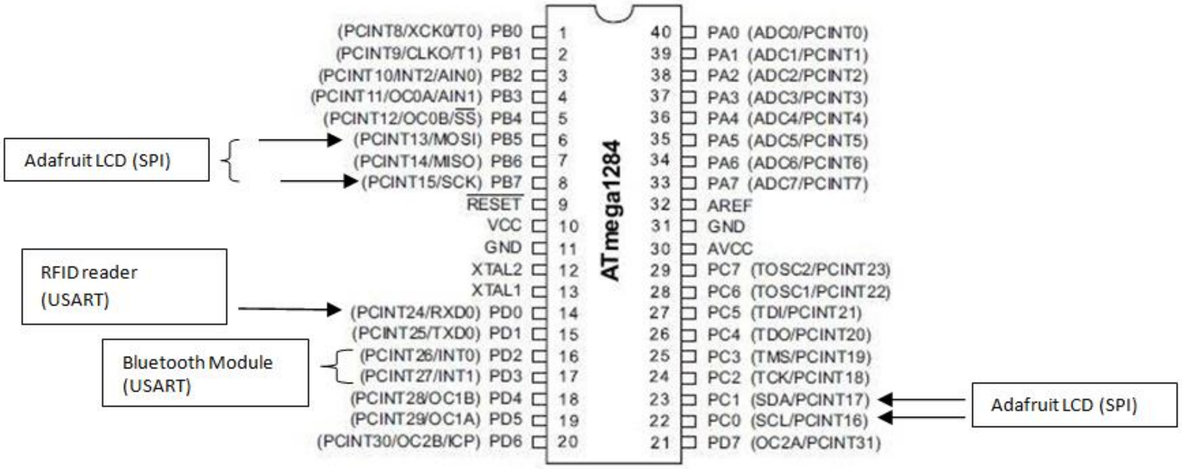
Hardware

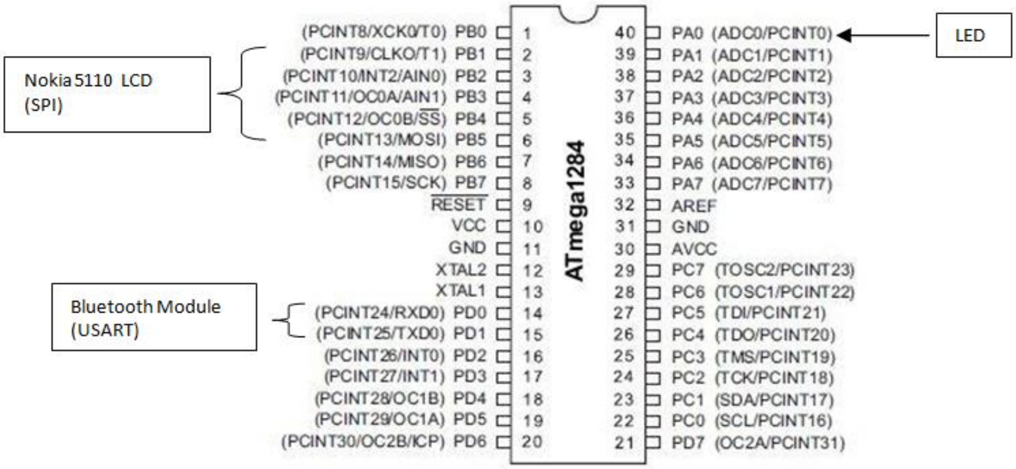
Parts List

Part	Part #	Quantity	Price (optional)
ATMega1284	ATMega1284	2	

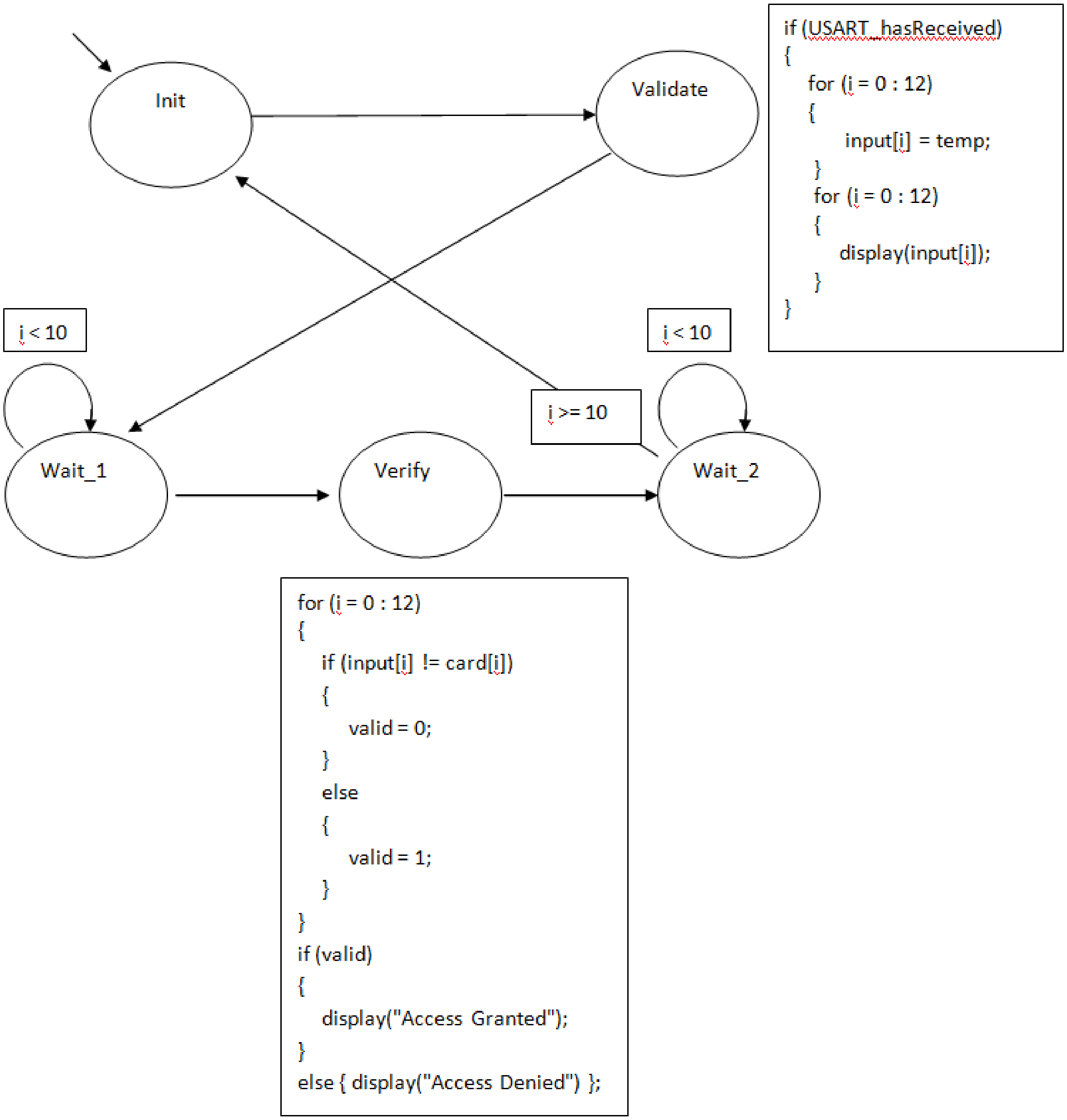
LCD	Adafruit TFT 1.8”	1	
LCD	Nokia 5110	1	
Bluetooth Module	DSD hc-05	2	
RFID reader	ID-12LA	1	

Block Diagram





Software



```
if (USART_hasReceived)
{
    for (i = 0 : 12)
    {
         input[i] = temp;
     }
     for (i = 0 : 12)
     {
         display(input[i]);
     }
}
```

```
for (i = 0 : 12)
{
    if (input[i] != card[i])
    {
        valid = 0;
    }
    else
    {
        valid = 1;
    }
}
if (valid)
{
    display("Access Granted");
}
else { display("Access Denied") };
```

- This SM first checks to see if USART is ready to receive from the RFID Reader. It then runs a loop that accepts the number of characters that make up the ID and displays it on the screen. Finally it validates the card by comparing the input to the correct card value and displays the result on the screen.

Implementation Reflection

I am mostly satisfied with the outcome of my project. I am relieved that I was at least able to get all my parts working together to form a working system. The process was frustrating at times, so if I were to do it again I would ask for help earlier and more often when I get stuck.

My favorite part of the project is the RFID sensor and bluetooth modules. I also found using a TFT LCD screen very interesting to use, but I had a hard time finding a decent library for it that wasn’t designed for Arduino. It made the screen much more difficult to use, and would have asked for advice on how to go about finding a library.

My first milestone was to get my bluetooth and Nokia LCD drivers working. I was able to get the LCD working relatively quickly because I used it for CS120B, but I had a hard time getting the Bluetooth module to work. I was trying to pair them using a fake Arduino board that would not work. I couldn’t tell if I was doing something wrong, or if it was the board that wasn’t working. Eventually I borrowed a USB serial converter which worked perfectly the first time.

My second milestone was to get the RFID reader and TFT display drivers working. The RFID was a bit confusing at first, but I was able to figure out how to read the tags relatively quickly. I had completed the RFID driver but not the LCD driver by the time I was checked off. The LCD was extremely frustrating because the library it came with was designed for Arduino, and I was not sure how to modify it to get it working on the ATmega. I found a small library for it after coincidentally finding a classmate who had also used the same screen. It isn’t the best library, but I am grateful that I still found it, and it was good enough for my project.

Completed components

I completed about 80% of my project.

Incomplete components

I was unable to complete the 2-way chat portion of my project. It proved to be a bit too ambitious, because I wanted to simulate a texting platform that displayed previous texts stacked on top of each other that moved upwards as the conversation progressed. I had worked for awhile on trying to get the full chat to work, but ran out of time and had to change my project to a locking system. The library was too frustrating to use to get the text to appear properly, and my state machine kept resetting flags for the other state machines, so I had to abandon that idea due to time constraints.

If I had to go back and work on it, I would implement a simpler texting process that sends or receives only one text at a time, so there are no interrupts, and I would not fuss with displaying stacked messages on top. Because the first thing I worked on was the Bluetooth module with a fake Arduino board that did not work, I wasted a lot of time that could have been spent on the rest of the development process.

Youtube Links

Make sure they are publicly viewable!

- Short video:
    - https://youtu.be/le9lqEhSdYQ
- Longer video:
    - https://youtu.be/MVEEUAqA3iY

Testing

Bluetooth Module

- Testing consisted of reusing lab 2 to create synchronized blinking LEDs on two separate ATmegas using bluetooth modules and USART.

RFID

- Tested by reading tags and displaying on TFT LCD screen. I validated the numbers by first reading the card on the USB serial converter on a terminal on my PC, and then comparing the numbers to what is displayed on the LCD.

LCD Screens

- Tested by displaying various text in different locations on screen.

Known Bugs

- RFID reader does not display ID tag immediately on startup. After reading the first few times, the display will start showing the ID.
    - I believe it could either be a timing issue with the SM, or I am not using USART correctly

    - I have tried to change the SM period, and alter the for loops that capture the data.
    - I would continue to modify the USART code and SM code.
- SM keeps resetting flag that triggers other SM’s.
    - I believe this also has something to do with timing, and the periods of the SM’s.
    - I have tried changing the period of the SM’s.

Resume/Curriculum Vitae (CV) Blurb

I created a secure wireless locking system that utilizes RFID and Bluetooth technology. The system reads an RFID tag, displays the ID on an Adafruit TFT 1.8” LCD and validates it. It then sends a signal to a remote system for locking and unlocking. The system utilizes two hc-05 bluetooth modules that communicate via USART, and an ID-12LA RFID reader which captures ID tags also through USART. If the device becomes unlocked, an LED lights up and a Nokia 5110 LCD displays the current state of the system.

Future work

If I were to continue working on this project, the next feature I would implement would be a simple chat system. I would add an SM that captures a string of characters and then sends it via USART to the other LCD to display. I do not know how to create a custom PCB, but I would implement it if it became feasible in the future.

References

- https://github.com/LittleBuster/avr-nokia5110

  ```
  /* Nokia 5110 LCD AVR Library
   *
   * Copyright (C) 2015 Sergey Denisov.
   * Written by Sergey Denisov aka LittleBuster (DenisovS21@gmail.com)
   *
   * This library is free software; you can redistribute it and/or
   * modify it under the terms of the GNU General Public Licence
   * as published by the Free Software Foundation; either version 3
   * of the Licence, or (at your option) any later version.
   *
   * Original library written by SkewPL, http://skew.tk
   */
  ```
- TFT: Experiments interfacing ATmega328 to an ST7735 1.8" LCD TFT display //

// Author : Bruce E. Hall //
Website : w8bh.net //
Version : 1.0 //
Date : 04 May 2014 //
Target : ATmega328P microcontroller //
Language : C, using AVR studio 6 //
Size : 3622 bytes

- https://github.com/Matiasus/ST7735

```
/
```

```
    * LCD driver for controller st7735.h / 1.8 TFT DISPLAY /

    *

    * Copyright (C) 2016 Marian Hrinko.

    * Written by Marian Hrinko (mato.hrinko@gmail.com)

    *

    * @author      Marian Hrinko

    * @datum       08.01.2016

    * @file        st7735.h

    * @tested      AVR Atmega16

    * @inspiration
       http://www.displayfuture.com/Display/datasheet/controller/ST7735.pdf
    *              https://github.com/adafruit/Adafruit-ST7735-Library

    *              http://w8bh.net/avr/AvrTFT.pdf

    *

    */
```