Dual Authentication RFID Keypad (DARK) Lock

By Youri Lavoie (Student # 1273431)

Final Project II (Course # 243-B22-HR)

May 3rd, 2016

# Executive Summary

Not completed.

Contents

[Executive Summary 2](#_Toc450946612)

[Table of Figures 4](#_Toc450946613)

[Introduction 5](#_Toc450946614)

[Functional Description 5](#_Toc450946615)

[Circuit Design 7](#_Toc450946616)

[*Power Supply* 7](#_Toc450946617)

[*Microcontroller* 8](#_Toc450946618)

[*Peripherals* 9](#_Toc450946619)

[*Graphical User Interface* 11](#_Toc450946620)

[Circuit Layout 11](#_Toc450946621)

[Bill of Materials 15](#_Toc450946622)

[Timeline 16](#_Toc450946623)

[Conclusion 16](#_Toc450946624)

[References 17](#_Toc450946625)

[Appendix 18](#_Toc450946626)

# Table of Figures

[Figure 1 - Block Diagram 5](#_Toc450946540)

[Figure 2 - Power Supply Schematic 7](#_Toc450946541)

[Figure 3 - Microcontroller Schematic 8](#_Toc450946542)

[Figure 4 - Peripheral Schematic 9](#_Toc450946543)

[Figure 5 - GUI 10](file:///E:\Desktop\TheEnd\TheEnd\FinalProjectReport.docx#_Toc450946544)

[Figure 6 - JTAG Numbering Change 11](file:///E:\Desktop\TheEnd\TheEnd\FinalProjectReport.docx#_Toc450946545)

[Figure 7 - Final Revision (v1.2) Circuit Layout 12](#_Toc450946546)

[Figure 8 - Revision v1.2 Disassembled 12](#_Toc450946547)

[Figure 9 - Revision v1.2 Assembled 13](#_Toc450946548)

[Figure 10 - Milestones 14](#_Toc450946549)

[Figure 11 - ILI9340 Breakout Board Schematic 16](#_Toc450946550)

[Figure 12 - PN532 Breakout Board Schematic 17](#_Toc450946551)

[Figure 13 - v1.1 Schematic 18](#_Toc450946552)

[Figure 14 - v1.1 Layout 19](#_Toc450946553)

[Figure 15 - Gantt Chart 20](#_Toc450946554)

# Introduction

Individual door security systems are commonly deployed in business settings. Deployment of such systems at important checkpoints allows businesses to limit access to specific areas to relevant personnel, significantly increasing the security of company assets and personnel. This can also be applied in residential settings in a smaller scale to secure doors that grant access to the premises, like the front door. Many existing solutions typically include either a keypad or radio-frequency identification (RFID) as a means for authentication and usually include a light-emitting diode (LED), a screen, and/or a buzzer to communicate with the user. However, they often only include one of the authentication methods mentioned above and have few features beyond this. My final project, the Dual Authentication RFID Keypad (DARK) Lock, combines previously mentioned components to create a feature rich door security system.

# Functional Description

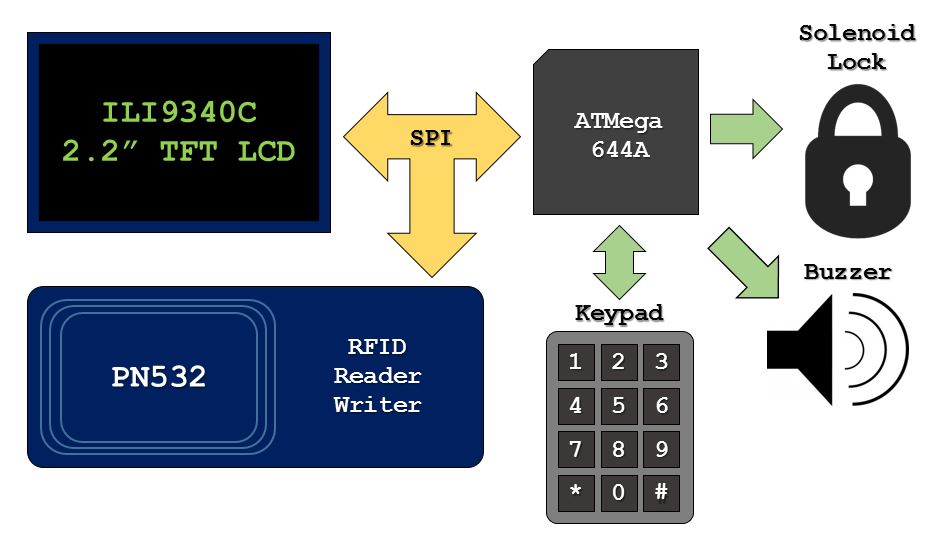


Figure 1 - Block Diagram

The DARK Lock has two methods of authentication and input to the user: a keypad and an RFID reader/writer (a PN532 breakout board). Both are connected to a microcontroller (an ATMega644A), which receives their inputs. The project also featured a display and buzzer as output devices and a solenoid lock.

The keypad in question is a 4 row by 3 column matrix keypad. The keypad allows the user to input their personal identification number (PIN) as a means for authentication. The fact it is a matrix keypad means that rather than each individual button having their own pin tying to the microcontroller, each row and column have one instead. This means that only 7 pins are needed to drive it rather than 13 (each individual button plus 1 for either high or low logic), which frees up several pins for other purposes. The rows act as outputs, while the columns act as inputs. They do not connect to each other unless a key is pressed. Only one of the rows output high at any given time while the rest are low. The output high cycles through each individual row several times a second. When the user presses a key, the row connects or intersects to the respective column of that key, causing the microcontroller to receive a high input on that column. The microcontroller then can deduce which button was pressed by looking up what value corresponds to the unique intersection of the high row and high column.

For RFID authentication, the PN532 v1.6 breakout board was used. It is an RFID reader/writer from Adafruit Industries and is powered by the PN532 near field communication (NFC) controller from Philips. This controller is extremely versatile and is ubiquitous; it can be found in most smartphones and other NFC enabled devices. It can read and write to compatible tags and devices, communicating at a frequency of 13.56 MHz. The PN532’s input supply range from its host interface is 1.6 V to 3.6V, its ambient operating temperature range is from -30 °C to 85°C, and it uses 3.3V logic. The breakout board supports universal asynchronous receiver transmitter (UART), serial peripheral interface (SPI), and inter-integrated circuit (I2C) communication. For this project, SPI was used to communicate between it and the microcontroller.

Since the aforementioned breakout board only operates using 3.3 V logic, a 4050 hex non-inverting buffer was used on my main board to shift the 5 V logic to 3.3 V. The integrated circuit (IC) accepts a supply voltage (VDD) of 3 V to 15 V and steps down higher input voltages to the supply voltage. In this case, VDD is equal to 3.3 V, so the 5 V logic input into the buffer from the microcontroller is output at 3.3 V. Its operating temperature range is -40°C to 85°C

The TFT display used was a 2.2” thin-film-transistor liquid-crystal display (TFT LCD) breakout board from Adafruit Industries, which communicates via SPI to the microcontroller. It has a 320 by 240 resolution and through its use if an ILI9340 TFT driver, can display full-bit colour (262 144 shades). The ILI9340 driver has an operating voltage of 1.65 V to 3.3V and an ambient operating temperature range of -20°C to 70°C. The breakout board features an onboard 4050 buffer and RT9193-3.3 low drop-off (220 mV) voltage regulator which allows the breakout to be used with 5 V logic. It also includes a microSD card reader which also communicates via SPI.

The microcontroller used for my project was the ATMega644A running the Arduino bootloader, operating at 16 MHz. The chip features 64 kilobytes of flash, 2 kilobytes of electrically erasable programmable read-only memory (EEPROM), and 4 kilobytes of random access memory (RAM). Its operating voltage range is 1.8 V to 5.5 V. The microcontroller has a maximum operating frequency of 20 MHz and an operating temperature range of -40°C to 85°C. During the initial setup, joint test action group (JTAG) is used to load the Arduino bootloader and change the fuses. As changing the fuses disables JTAG, from that point forward the ATMega644A communicates with the PC via serial using a USB to serial converter.

The locking mechanism used is a solenoid deadbolt lock. A solenoid is simply a tightly wound coil that acts as an electromagnet. When 12 V and 1.5 A rushes through the coil, it generates a strong electromagnetic field which retracts the metal deadbolt. When no power is supplied to the coil, a spring deploys the deadbolt outwards. Because of this design, it is a fail-secure lock, meaning the deadbolt will secure the door in the event of a sudden loss of power. Naturally, since the microcontroller cannot output such high voltages and current from a single pin, this effect is achieved by tying the output pin to the gate of an IRLU024N metal-oxide-semiconductor field-effect transistor (MOSFET). When the microcontroller outputs high on the output pin, the MOSFET turns on, allowing large amounts of current to flow through it directly from the power adapter.

The buzzer used was a standard surface mount piezo buzzer connected directly to an output pulse-width modulation (PWM) pin of the microcontroller. It supports 0 Hz to 10 kHz square wave inputs to generate tones of the specified frequency. Volume is determined by the amplitude of the square wave, which can be from 0 V (no sound) to 12.5 V (max) peak to peak. Its operating temperature range is -40°C to 85°C.

# Circuit Design

The circuit schematic is composed of 3 major subsections. Therefore, this section will address each subsection individually. The graphical user interface (GUI) will also be covered in its own subsection.

## *Power Supply*

There are three power rails needed for the DARK Lock to operate its many modular peripherals. These are the 12 V, 5 V, and 3.3 V power rails.

The device’s input stage is powered via a 12 V, 2 A power adapter through a barrel jack. This ties in directly to the solenoid circuit to operate it. After passing through a diode used for reverse current protection, it also connects to the input of a LM1117-5 linear regulator with a voltage dropout of 1.2 V at 800 mA. This regulator generates the 5 V power rail necessary to power the microcontroller and the peripherals excluding the PN532 breakout board. Because of the steep drop from 12 V to 5 V, the regulator dissipates a lot of power which is output as heat, causing it to warm up significantly during normal operation. This however has no effect on performance, as the input voltage and operating temperature remains within acceptable ranges (0 V to 15 V, -40°C to 125°C respectively). A green light emitting diode (LED is tied to the 5 V power rail for diagnostics.

The 5 V rail then connects to a LM1117-3.3 linear regulator which generates the remaining 3.3V power rail needed for the PN532 breakout board. Being from the LM1117 family, it has the same specifications as the other linear regulator, except that it outputs 3.3 V rather than 5 V.

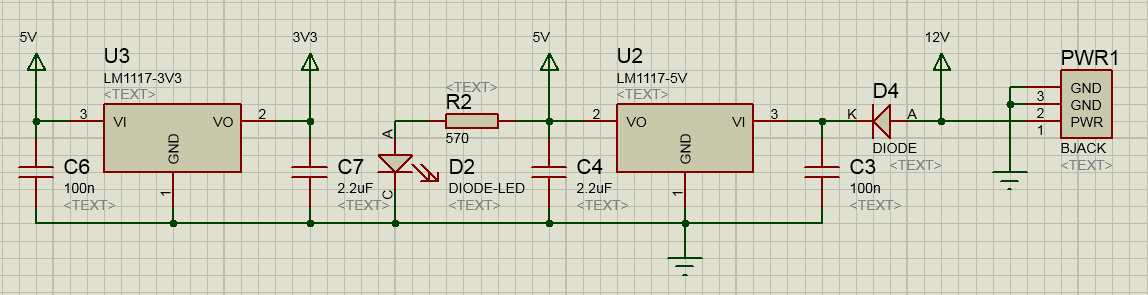


Figure 2 - Power Supply Schematic

## *Microcontroller*

The microcontroller used is the ATMega644A (referred to as ATMEGA644P in schematic below) operating at 5 V and at a frequency of 16 MHz generated by a crystal. There are two headers used to communicate with the PC tied into the microcontroller: a CONN-SIL6 used for the USB to serial converter and a CONN-DIL10 used for the JTAG interface. The JTAG header is only used once during initial setup to load the Arduino bootloader and set the fuses of the microcontroller. It needed to be included in the design due to the fact the microcontroller is surface mount, as is most of the layout. Because it is surface mount, the bootloader could not be loaded before soldering the chip on the board. Thereafter, the microcontroller communicates via the USB to serial adapter. Other supporting circuitry includes the reset pushbutton circuit.

The buzzer is connected directly to a PWM pin and is the only peripheral that is not modular; it is soldered on the board rather than connecting through a header. The rest of the pins output to headers connecting to the remaining modular peripherals. The buzzer plays distinct contextual audio queues. There are unique sounds for the following actions: when an RFID tag has been detected by the PN532, when access has been granted, when access has been denied, and when a tag’s unique identifier (UID) has successfully been sent to the PC via serial communication.

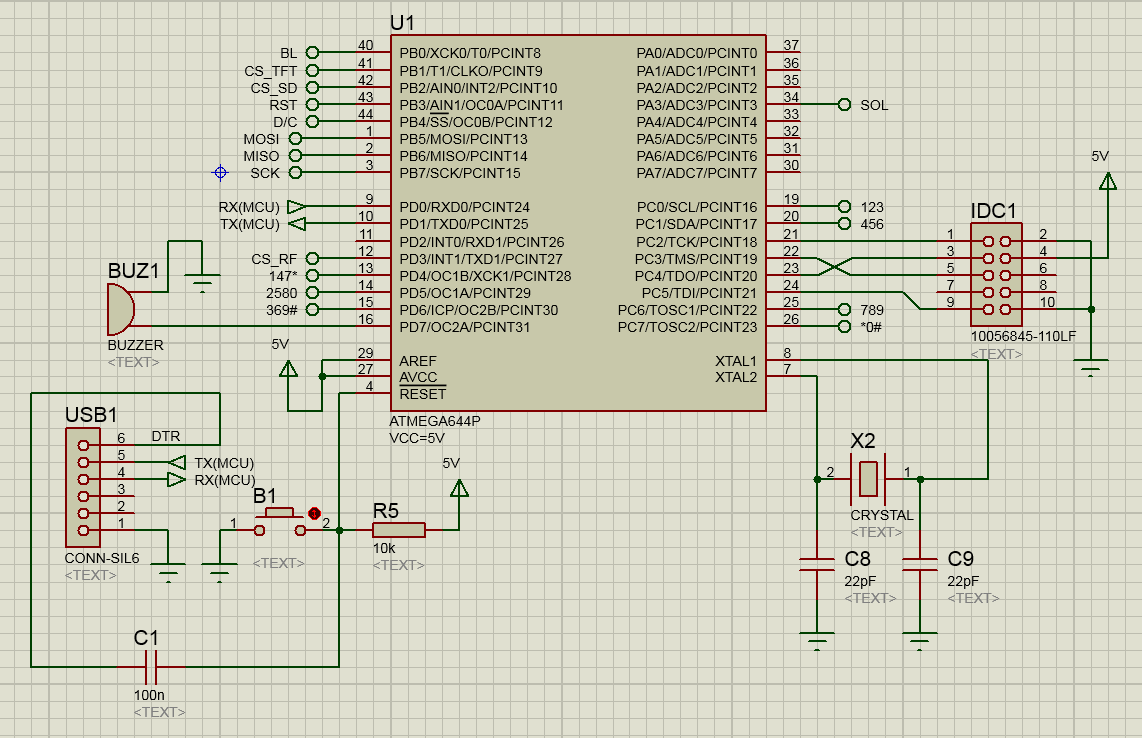


Figure 3 - Microcontroller Schematic

## *Peripherals*

As previously mentioned, all peripherals used in the project are modular with the exception of the buzzer. The matrix keypad, solenoid, ILI9340, and PN532 breakout boards, connect to their respective headers. This has the advantage of providing the DARK Lock with a lot of versatility. If any of the aforementioned peripherals fail, they can easily be swapped out.

Because both the keypad and PN532 use blocking polling to detect an input, in order to use them both concurrently, a timeout was implemented in firmware to both polling attempts. If no keys have been touched yet on the keypad, after a few milliseconds the PN532 activates and tries to scan for a RFID tag. If no tag was found after a few milliseconds, the PN532 gives up and lets the keypad try again. This work around causes an infinite cycle which happens a several times a second that is transparent to the user and allows for both authentication methods to be used.

To enter a PIN using the keypad, the user simply needs to press the desired string of keys. As soon as a key is pressed, the DARK Lock ceases the scan for RFID tags and waits for the user to finish dialing his code. There is no set length for PINs. A user can dial as many or as little digits as they wish as long as they confirm their selection by pressing pound (#). Each key press appends a character to string used as a buffer. Once pound is pressed, ‘#’ is removed from the string and it is converted to an integer. The PIN integer that was input is then compared to all known PINs in the database. If it positive match is found, the system unlocks the solenoid.

To scan an RFID tag, the user simply needs to pass a compatible tag a few centimeters in front of the PN532 breakout board. The PN532 will then obtain the 4 byte unique identifier (UID) associated to the tag, store it as an unsigned 8-bit integer array, and pass it on to the microcontroller. The microcontroller will then compare the scanned UID with all known UIDs in the database. If it positive match is found, the system unlocks the solenoid.

If the GUI sends a SCANxx (0xC1) command to the DARK Lock via serial, the device will enter scanning mode: the DARK Lock will wait for a tag to enter the field of the PN532 breakout board to read it indefinitely. Once a tag is read, the UID is sent to the PC via serial and the lock returns normal mode after a few seconds.

The solenoid lock operate at 12 V, 1.5 A. Since most other solenoid lock solutions operate similarly, different lock types can be used. One could easily replace the deadbolt lock with a striker lock solenoid instead and the device would still function as intended. It simply unlocks for 3 seconds after the gate of the MOSFET is opened by a high signal from the microcontroller. This happens only when a valid PIN has been entered on the keypad or a tag with a valid UID has been detected.

Schematics for the ILI9340 and PN532 breakout boards are included in the Appendix.

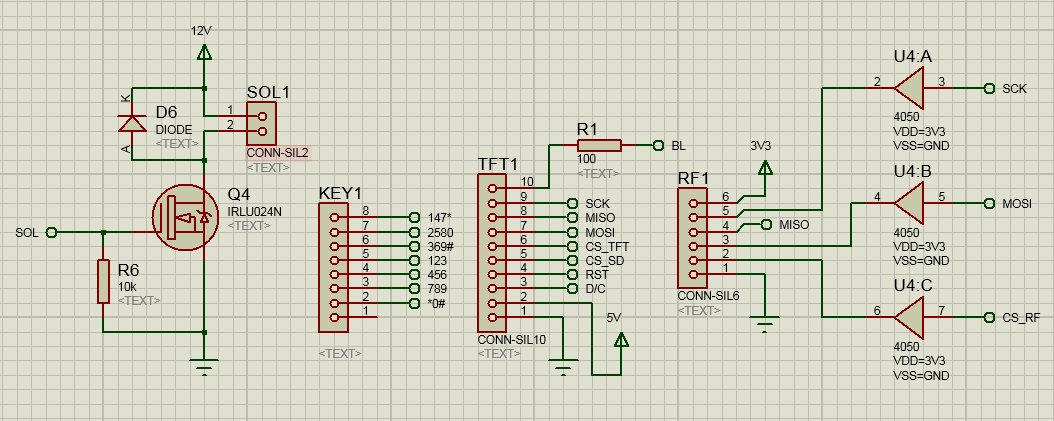


Figure 4 - Peripheral Schematic

## *E:\Desktop\TheEnd\TheEnd\guiSnip.JPGGraphical User Interface*

The GUI, as seen on the left, is incomplete and thus its main feature is not functional. Its only functional feature is that it can send the previously mentioned SCANxx (0xC1) command to the DARK Lock to obtain a UID in plain text within the plain text editor.

Its intended purpose was a database of 10 users that could be synced to the hardware via serial. The DARK Lock would then retain the synced database even if there was an unexpected loss of power. This is not implemented at this time. Therefore, pressing “Sync” yields nothing.

Other than the “Scan” feature, the current version of the GUI features serial controls to connect to or disconnect from the DARK Lock from a list of available serial ports. The “Enumerate” button refreshes the list of available ports.

Figure - GUI

User details are selectable from a dropdown menu currently limited at 10 users; this number can be expanded if modifications are done in firmware and software. Each user’s details are stored in individual text files following a specific format: line 1 is the user’s name, line 2 is whether or not the user is active or not, line 3 is the user’s PIN, and line 4 is the user’s RFID tag UID. The plain text editor displays the contents of the user selected in the dropdown menu. Changes can be saved using the save button.

# Circuit Layout

A particularity of my layout is that every component other than the headers (including the barrel jack), crystal, and pushbutton are surface mount. This design choice was taken to minimize the space taken by the DARK Lock. Thanks to this, the entire design (excluding peripherals) fits on a printed circuit board (PCB) that is smaller than a credit card, the final revision (v1.2) measuring only 2.6” by 1.9” (66.5 mm by 48 mm). The width of the PCB is also identical to that of the ILI9340 breakout board which sits above it. Therefore, when the display peripheral is connected, all of the circuitry of the DARK Lock hides neatly below it, protecting it from tampering. This effect did require the display header and keypad header to be placed strategically. Besides this, other layout considerations include the traces used to power the solenoid being thicker as they required more current to run through them.

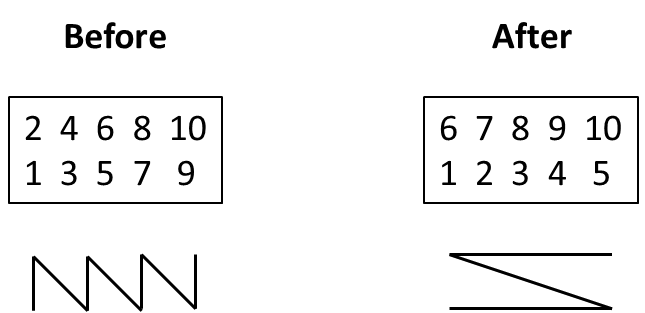
There only exists one minor issue with v1.2: after changing the package of the JTAG header, the numbering of the pins changed. This meant that a conventional ribbon cable could not be used to configure the chip via JTAG and individual female-to-female Dupont cables would need to be used instead to map the mismatching numbering correctly on the connecting devices. Since the JTAG interface was only used once and was disabled thereafter, this was a non-issue.

Figure 6 - JTAG Numbering Change

The first revision (v1.1, see Appendix) had many issues, all of which were addressed and solved in the final revision. Firstly, due to a design oversight, the buzzer in v1.1 was not connected to a PWM pin at all and therefore did not function. Secondly, the packages used for the horizontal headers for the PN532 breakout board, the USB to serial converter, and the solenoid had the wrong pitch, which ended being slight narrower than 0.1” (2.54 mm). This meant that none of the horizontal headers purchased would fit and required the peripherals to be connected via dangling wires instead. Thankfully, this issue had no impact on the device’s operation and its effect was thus merely aesthetic.

Finally, the v1.1 was expected to use a different display than v1.2. The original design was based around a generic version of the Adafruit breakout board from China that was supposed to be functionally identical to it. One difference between the two was the pin configuration; the Adafruit version used 10 pins to the left of the breakout board for both the display and SD card reader, while the generic used 9 pins to left for the display and 5 pins to the right for the SD card reader. After assembly, it was discovered that the generic simply did not work with my design, but the official Adafruit breakout board did work. This meant that to have a functioning display, the Adafruit breakout board had to be connected with male-to-female Dupont cables to the header as the pin locations did not match.

Overall, other than the buzzer not working, revision v1.1 was a success and functioned normally. However, because all the wires protruding from the unusable headers, it was a huge eyesore. This ultimately led the creation of v1.2.

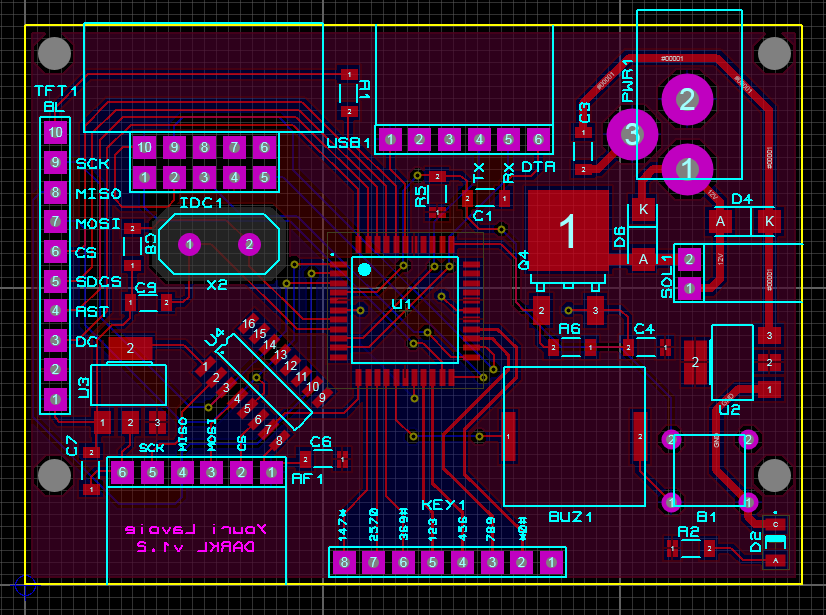


Figure 7 - Final Revision (v1.2) Circuit Layout

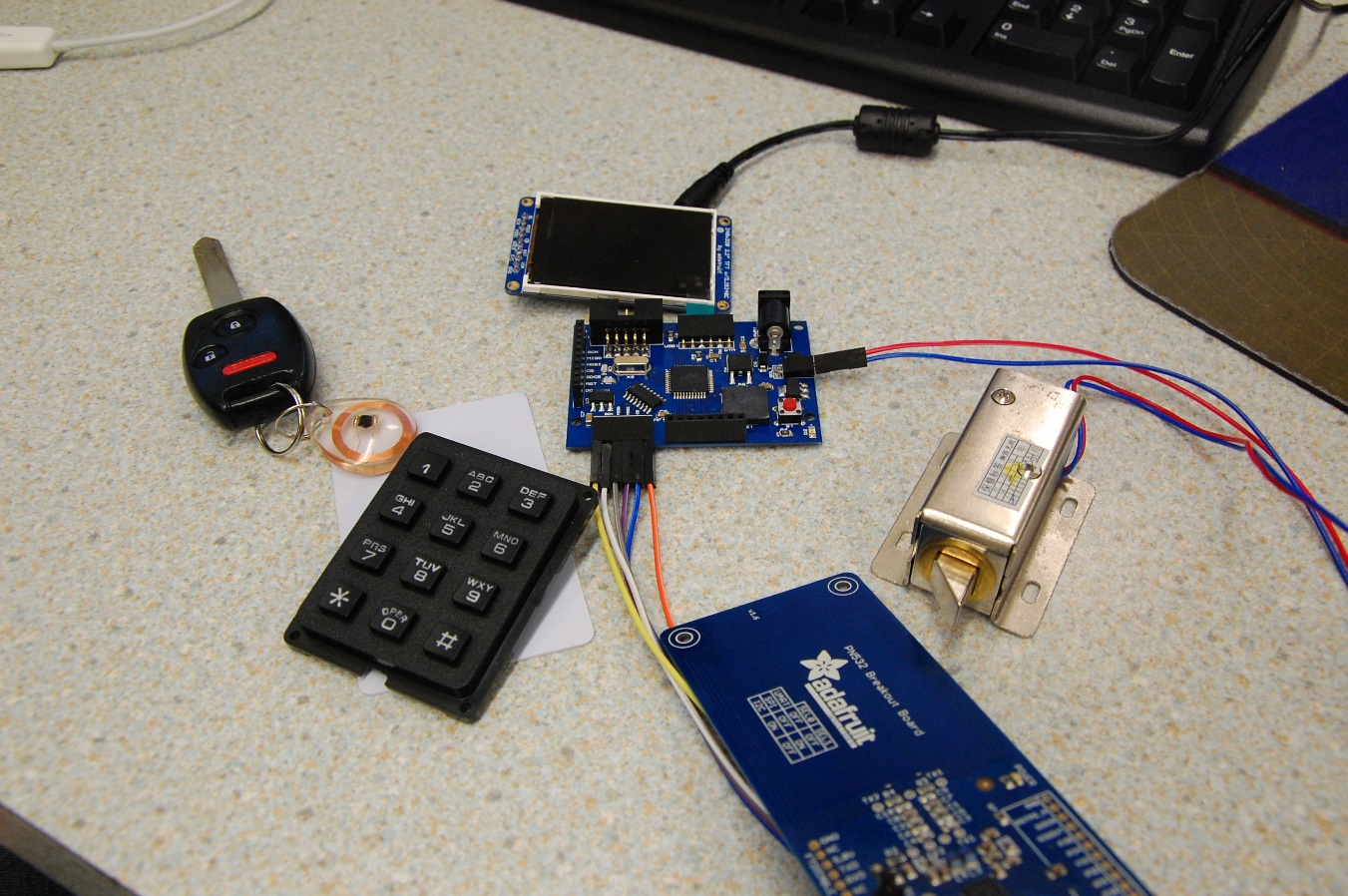


Figure 8 - Revision v1.2 Disassembled

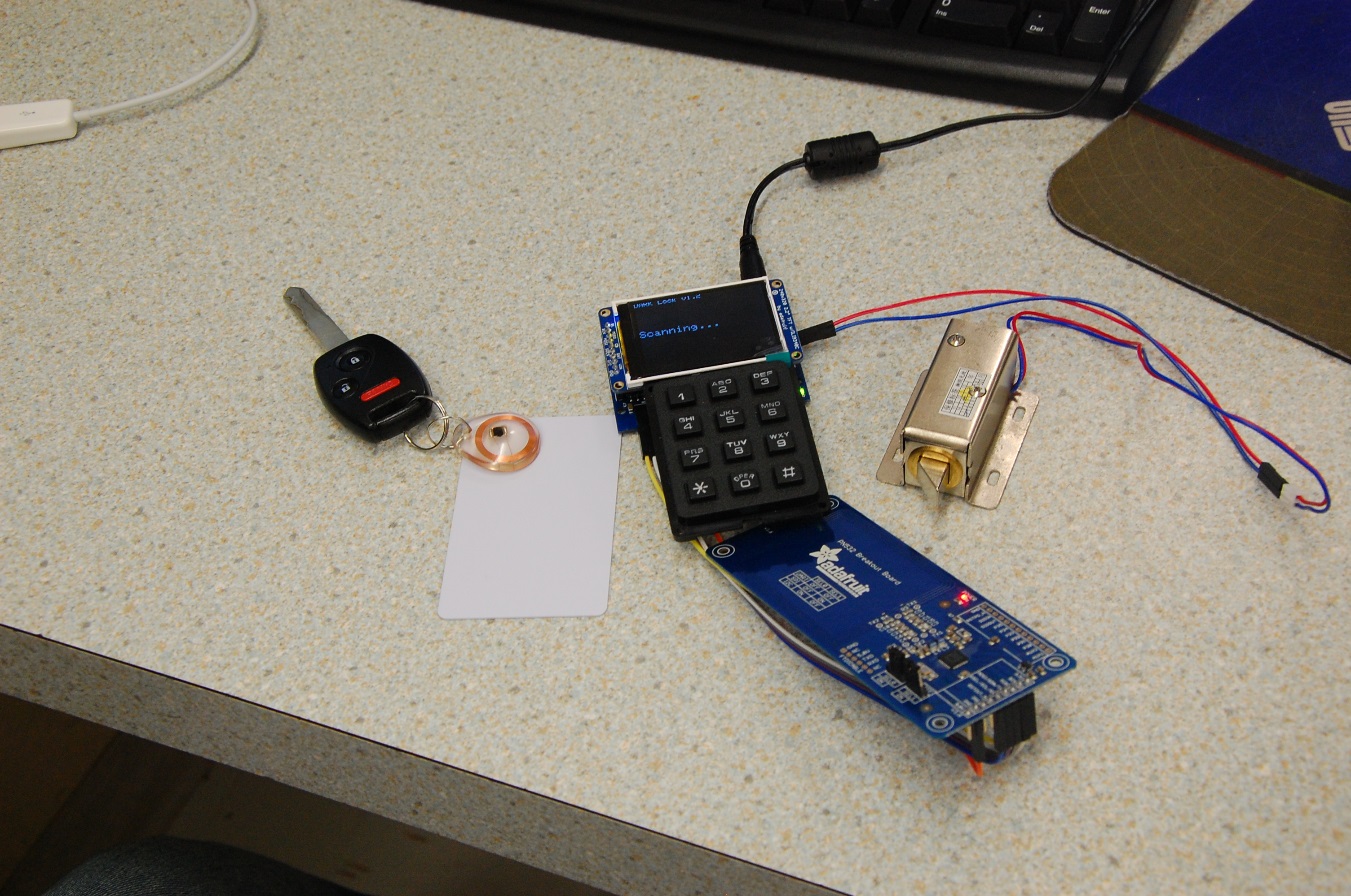


Figure 9 - Revision v1.2 Assembled

# Bill of Materials

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Quantity** | **References** | **Value** |
| Capacitors | 3 | C1,C3,C6 | 100n |
| Capacitors | 2 | C4,C7 | 2.2uF |
| Capacitors | 2 | C8-C9 | 22pF |
| Resistors | 1 | R1 | 100 |
| Resistors | 1 | R2 | 570 |
| Resistors | 2 | R5-R6 | 10k |
| Integrated Circuits | 1 | U1 | ATMEGA644A |
| Integrated Circuits | 1 | U2 | LM1117-5V |
| Integrated Circuits | 1 | U3 | LM1117-3V3 |
| Integrated Circuits | 1 | U4 | 4050 |
| Transistors | 1 | Q4 | IRLU024N |
| Diodes | 1 | D2 | DIODE-LED |
| Diodes | 2 | D4,D6 | DIODE |
| Miscellaneous | 2 | B1,KEY1 |  |
| Miscellaneous | 1 | BUZ1 | BUZZER |
| Miscellaneous | 1 | IDC1 | CONN-DIL10 |
| Miscellaneous | 1 | PWR1 | BJACK |
| Miscellaneous | 2 | RF1,USB1 | CONN-SIL6 |
| Miscellaneous | 1 | SOL1 | CONN-SIL2 |
| Miscellaneous | 1 | TFT1 | CONN-SIL10 |
| Miscellaneous | 1 | X2 | CRYSTAL |
| Miscellaneous | 1 | - | PN532 BREAKOUT |
| Miscellaneous | 1 | - | ILI9340 BREAKOUT |
| Miscellaneous | 1 | - | 12V-2A POWER ADAPTER |
| Miscellaneous | 1 | - | USB TO SERIAL CONVERTER |

# Timeline

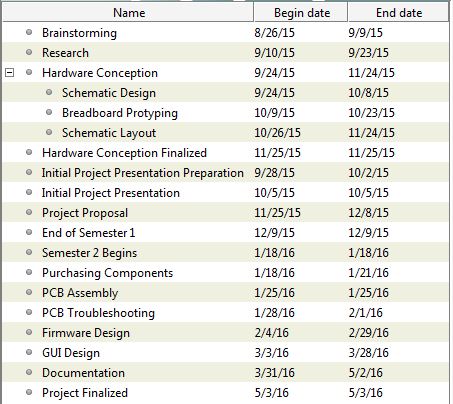


Figure - Milestones

Unfortunately, due to the switch from a through hole to a surface mount design, layout conception took longer than expected in the 1st semester, with the PCBs only arriving in the final few weeks of the semester. Additionally, due to the aforementioned issues with v1.1, a substantial portion of the 2nd semester reserved for firmware and graphical user interface (GUI) design was instead used to troubleshoot and revise hardware. This shortened the available development time for software significantly.

Unfortunately, in the end, it shows. Though hardware is excellent and the firmware is solid, accomplishing all intended functionality, the GUI pales in comparison. Ideally it would have been a modifiable database for users on the PC that could sync up with the DARK Lock. The lock would then update its own client side database and would retain the updated information even if it lost power. I was not able to achieve this within my given time frame. The only function my GUI serves at this point is to allow the user to scan a RFID tag and transmit its UID to the PC in plain text. Ultimately, this experience was a lesson in time management for me.

Gantt Charts for both semesters are included in the Appendix.

# Conclusion

Not completed.

# References

<http://pcbheaven.com/wikipages/How_Key_Matrices_Works/>

<https://www.adafruit.com/product/1480>

<https://cdn-shop.adafruit.com/datasheets/ILI9340.pdf>

<https://raw.githubusercontent.com/adafruit/Adafruit-2.2-SPI-TFT/master/Adafruit_2.2_SPI_240x320.png>

<https://cdn-learn.adafruit.com/assets/assets/000/026/820/original/rfid___nfc_pn532_16.png?1438272672>

<http://www.murata.com/~/media/webrenewal/products/sound/sounder/vppt-buzj083-d.ashx>

<http://www.atmel.com/images/atmel-8272-8-bit-avr-microcontroller-atmega164a_pa-324a_pa-644a_pa-1284_p_datasheet.pdf>

<http://www.ti.com/lit/ds/symlink/lm1117.pdf>

<http://www.nxp.com/documents/data_sheet/HEF4050B.pdf>

# Appendix

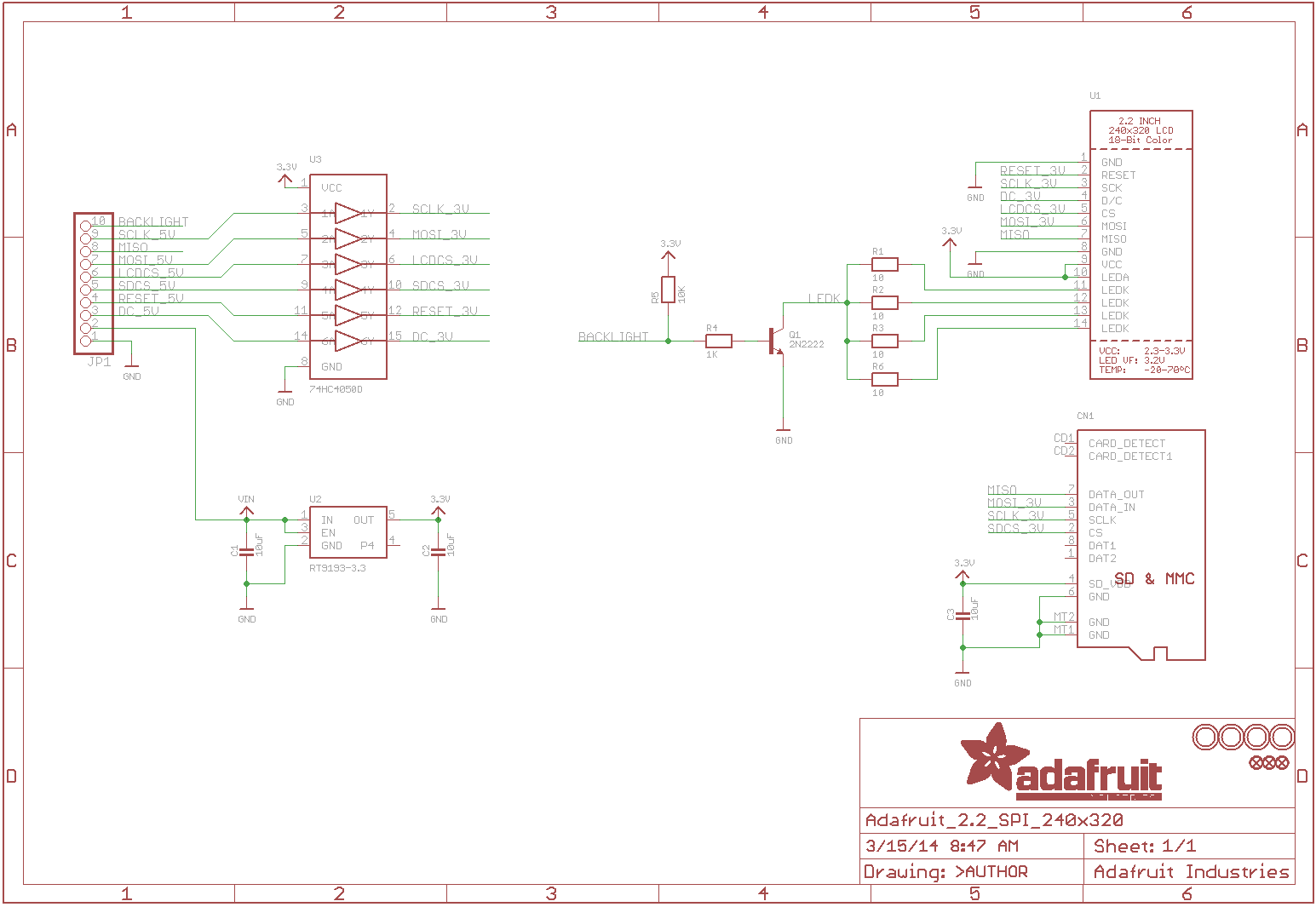


Figure 11 - ILI9340 Breakout Board Schematic

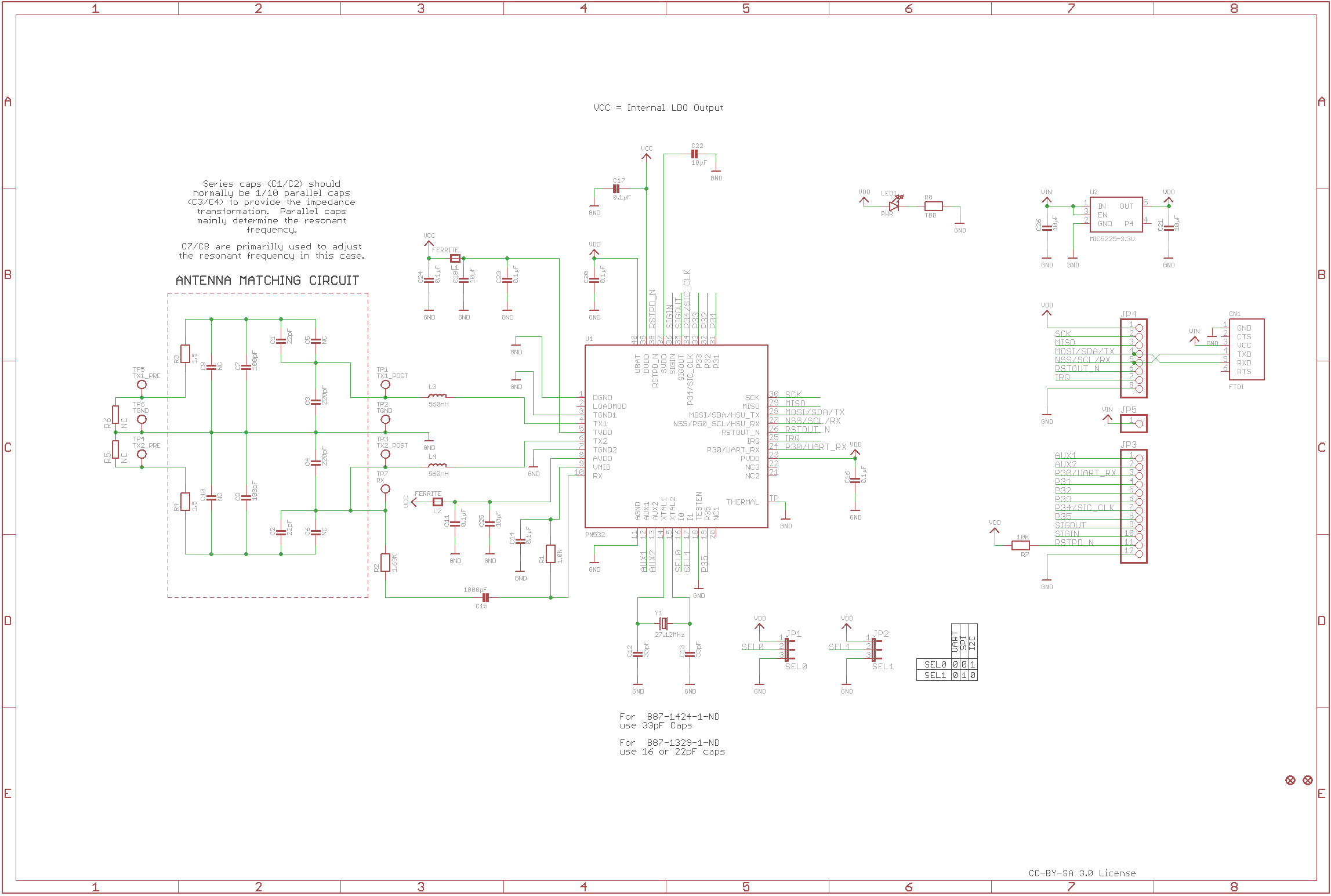


Figure 12 - PN532 Breakout Board Schematic

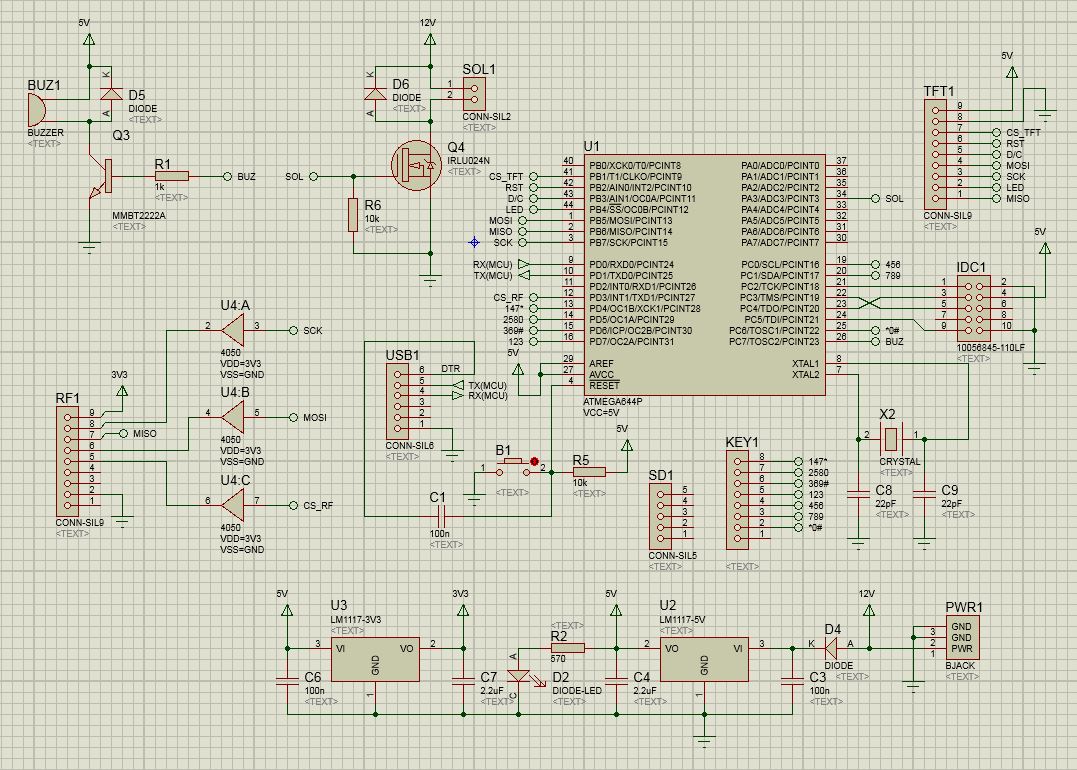


Figure - v1.1 Schematic

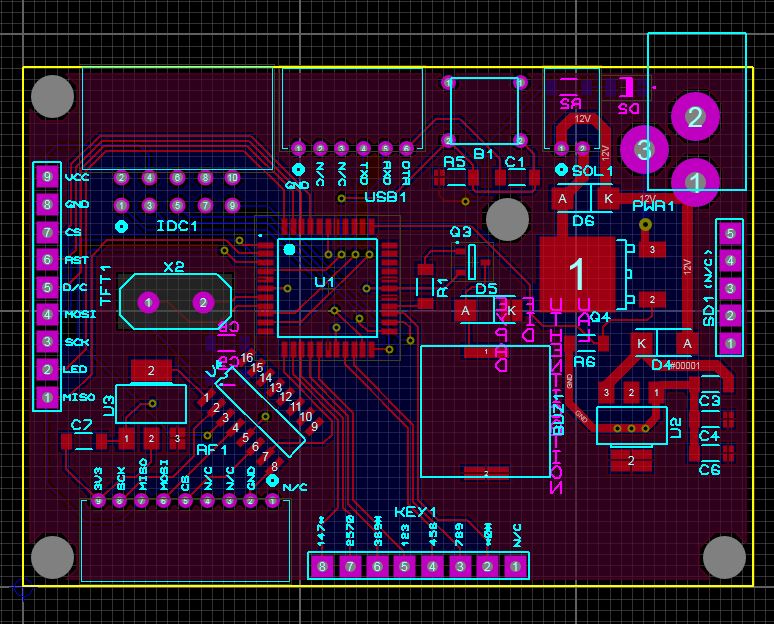


Figure - v1.1 Layout

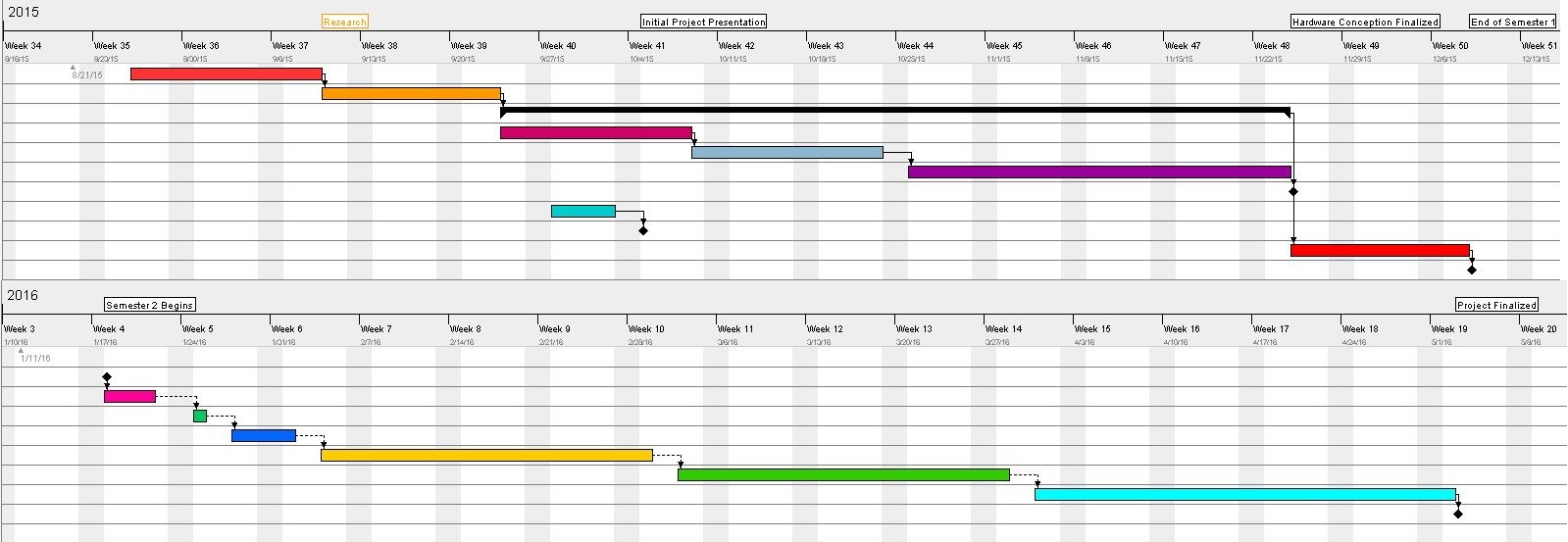


Figure - Gantt Chart