Dorm Door Locking Mechanism

Team 44 - Thomas Orr, Mason Hoppe, and Karan Usgaonkar
ECE 445 Design Document Fall 2017
TA: Zhen Qin

1 Introduction

1.1 Objective

The dorms on campus currently do everything that they can to prevent theft in each room. The measures that currently exist are icard scanners to enter each building and deadbolt locks on each individual student's room. Yet the University's police department's crime log[1] indicates there are still thefts in campus dorms every year. It is very easy to enter the building by simply following someone who lives there into the building, bypassing the first layer of security. Secondly, people tend to leave their doors unlocked as a personal preference or because they think that they'll be back quickly. In either case, this is when most thefts occur in the dorms.

In order to counteract this problem, we are creating a dorm door lock attachment which will automatically lock any dorm door that it is attached to. One problem that naturally arises is when a student leaves their room without their key on hand. In order to prevent this from being an issue, we will be creating this attachment with alternate entry methods in mind. We will be using a pin pad and an RFID scanner to allow the student to unlock the door without needing a key. The system will be wall powered with a battery backup in case of a power outage.

1.2 Background

Lock systems are ubiquitous in today's world and are used for a variety of purposes. Locks have been around for more than 6,000 years and were first seen in Egypt when a wooden post was affixed to the door, and a horizontal bolt slid into the post[2]. Today, locks are important to provide personal security and to prevent intruders from gaining access to unauthorized places; however, locks become inconvenient if the key is lost or forgotten inside the room.

There are many types of locks out there today which replace the typical deadbolt-key locking system. These include locks that take a PIN, fingerprint, and/or key fob (RFID). However, all of these types of locks must completely replace the deadbolt that currently exists on the door. Meaning, that if you are currently renting a space, or simply are not allowed to alter the door in question, you are unable to use any of the lock replacements that currently exist on the market. This is the niche that our product will fill in the market. Our product will be an attachment that does not alter the door permanently in any way. Thus, we will be able to reach a currently untapped market for these types of lock improvements.

1.3 High Level Requirements List

- We will implement two additional methods of unlocking the door: Numerical PIN keypad and RFID sensor.
- Use power efficiently by implementing a passive low power mode and an active high power mode.
- The door will automatically lock itself when it closes.

2 Design

2.1 Block Diagram

The block diagram below is separated into a couple main components. The outside interface is two input devices: the *Keypad* and the *RFID Scanner.* A user can either scan an RFID tag or enter a PIN from outside the door. Inside the door is two more input sources: a *Proximity Sensor* that detects when the door opens and closes, and a *User Interface* of 4 buttons and 3 switches that allow the user to control the system. A *Microcontroller* tests the input from outside the door. The *State Machine* uses all the inputs to instruct the *Lock Mechanism* to lock or unlock. The *Power Supply* provides power to the system and includes a battery backup in case of a power outage.

Figure 1: Lock Mechanism Block Diagram

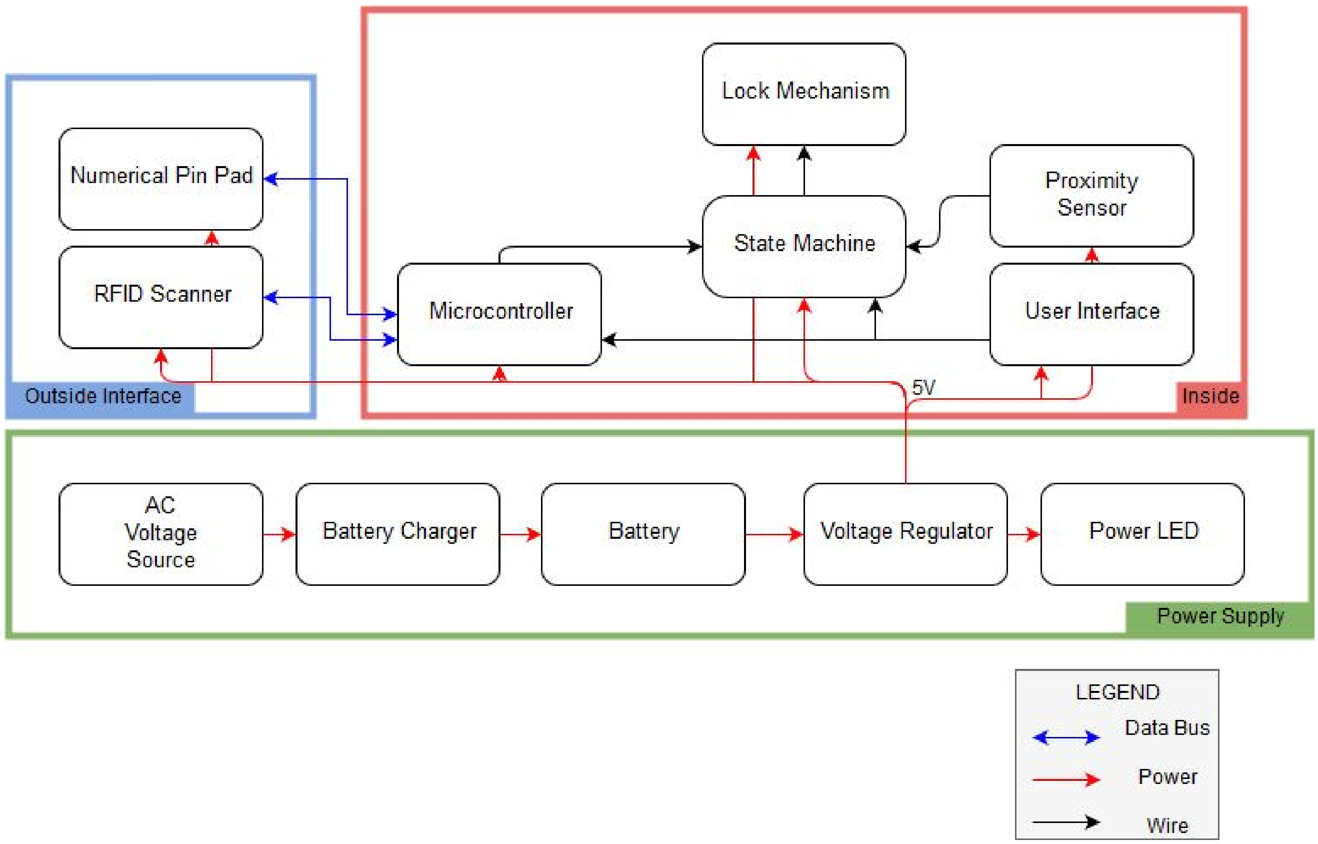
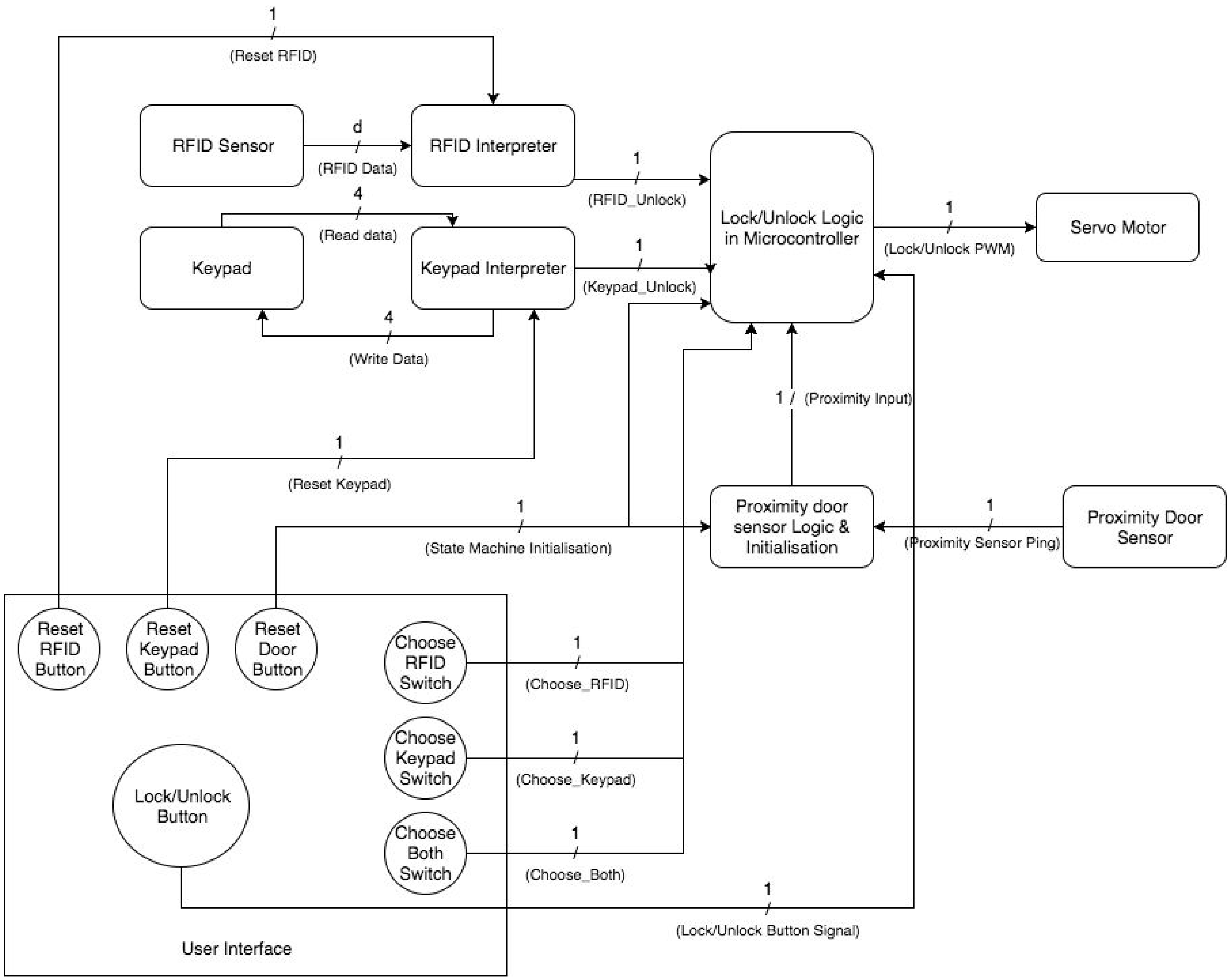


Figure 2: More Specific Data & Logic Flowchart



The User Input can one of of 7 different kinds on the user interface:
4 buttons:

- Resetting the RFID
- Resetting the Keypad
- Resetting and Initializing the door sensor
- Locking and Unlocking the door

3 switches:

- Choosing the RFID for entry
- Choosing the Keypad for entry
- Choosing both the methods for entry

When a user touches their ID card to the RFID sensor, data (represented as d) is passed to the RFID interface from the RFID sensor. The RFID interpreter assesses the data passed to it. If the data matches the preset RFID tag code that was set by the UI as the only correct code, the interpreter sends an RFID_Unlock signal to the state machine.

Similarly, for the keypad, a user pushes in buttons on the keypad which completes the circuit in the keypad (explained earlier by completing the circuit between the 4 rows and 4 columns). For each of the buttons pressed, information is sent and queued in an interpreter. The interpreter assesses if the data contained in its queue is the correct combination set through the UI, and if so, a Keypad_Open signal is sent to the Lock/Unlock Logic in the Microprocessor.

The proximity sensor sends a pulse if the door has just been closed or opened. When the proximity sensor pings the intermediate hardware between the microcontroller and the proximity sensor, the lock/unlock logic of the microcontroller will receive a steady 1 signal if the door is open, and a steady 0 signal if the door is closed. The microcontroller's Lock/Unlock logic makes a decision of whether the door needs to be locked or unlocked, and sends a PWM to the servo motor to have it lock or unlock the door.

2.2 Physical Design

The premise for this design is that it is meant to be an attachment that fits over an existing deadbolt lock without altering it in any way. The deadbolt on the inside of the door will have a cog fitted around it, and the cog will connect the deadbolt to a servo motor. The servo motor will be house in a cover above the deadbolt and will not hide the deadbolt. This is to ensure that the user can operate the door normally if the entire system is powered off. This cover will also have all logic and interpretation components on the inside of the door just above the cover so that no one can tamper with the system from outside of the door.

There will then be an adjustable length bar going between the door frame and the door that will hold the outside components in place. This bar is adjustable to allow the user to attach and remove the cover with ease. The only things holding the lock attachment to the door is the tension in the bar which allows the cover to hold onto the lock. We also plan to use velcro if we determine that the friction and tension is not enough to hold it on.

The outside of the door will consist of only the input devices and wiring that send the signals back to logic inside the door. Doing this ensures reduces the risk of someone sending artificial signals to trigger an unlock. Ideally, the wiring will be contained within the frame of our mechanism to prevent damage to and tampering with the wiring. If time permits, we would like to replace the wiring with bluetooth receivers and transmitters. This would make the product more commercial.

2.3 Logic Design

Open/Close Logic In Microcontroller

Table 1: K-Map for Internal_Unlock Signal when the Choose_Both Switch is in the OR State

RFID_Switch/Keypad_Switch \ RFID_Open/Keypad_Open	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	1
10	0	0	1	1

Table 2: K-Map for Internal_Unlock Signal when the Choose_Both Switch is in the AND State

RFID_Switch/Keypad_Switch \ RFID_Open/Keypad_Open	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	1	0
10	0	0	1	1

Table 3: K-Map for the Locked State

Door_Closed/Proximity_Detected \ Internal_Unlock/Lock_Unlock_Button	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	0	0	0
10	0	0	0	0

Table 4: K-Map for the Proximity Sensor Input

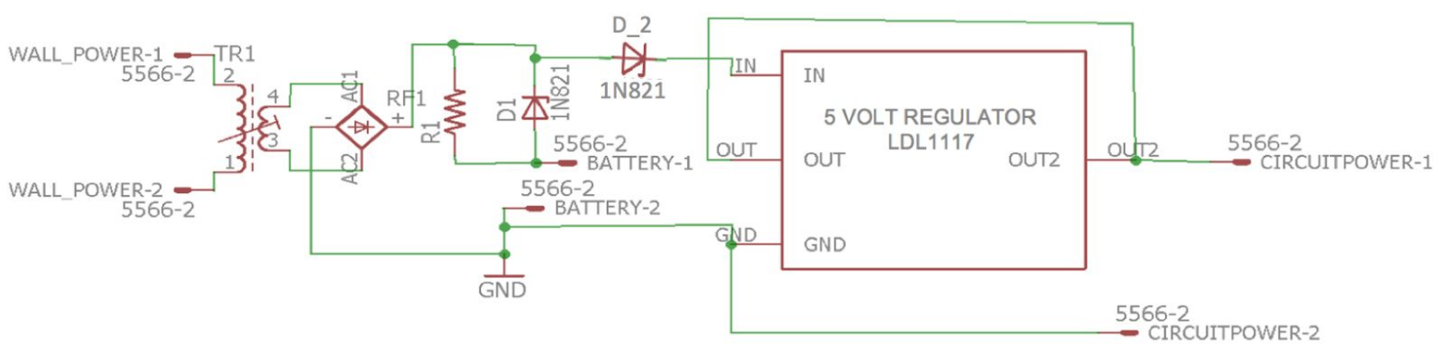
		Proximity_Sensor_Ping/State_Reset			
		00	01	11	10
Current State	0	0	1	1	1
	1	1	1	0	0

2.4 Schematics

In each of the following schematics, R is a resistor, D is a diode, RF is a rectifier, and TR is a transformer.

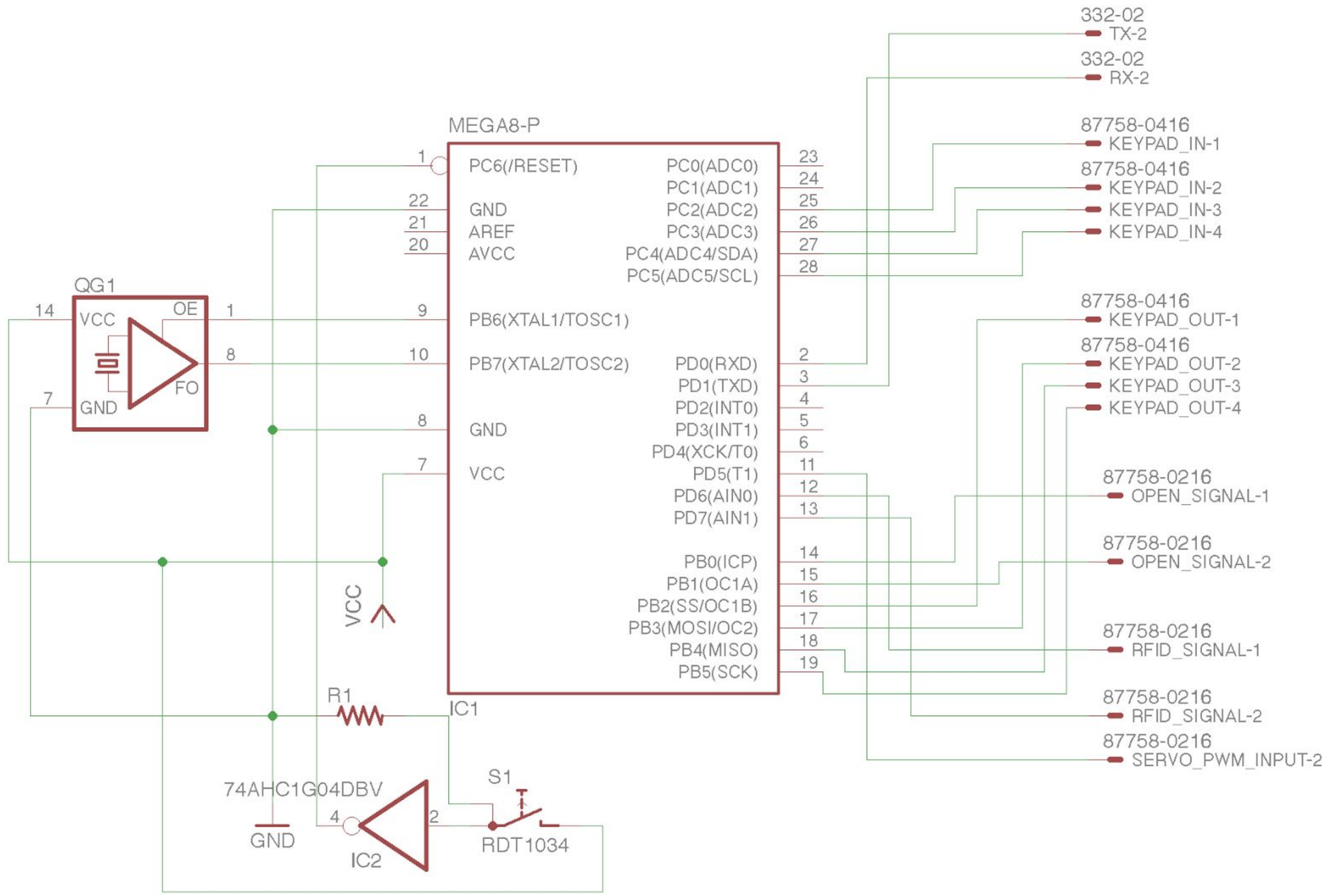
2.4.1 Power Supply

Figure 3: Power Supply Schematic



2.4.2 Microcontroller/RFID interpreter/Keypad Interpreter/Servo Output Schematic

Figure 4: Microcontroller-Hardware Interface



The above diagram illustrates how we plan to connect Atmega168 chip (which will be our microcontroller chip) to hardware components. Ports 1, 22, 9, 10, 8, and 7 (all on left side from top to bottom) are connected to components necessary for the Atmega168 chip to work. On the right side, ports 25-28 are connected across the inputs of the keypad, and ports 16 to 19 are connected across the outputs of the keypad - these 8 ports will be used to understand the combination pressed on the keypad (logic will be handled programmatically by software). Ports 2 and 3 (connected to TX and RX) are to connect to the USB adaptor (necessary to program the chip via software). Ports 14 and 15 are used to output open_signal which will be used to determine whether the servo motor needs to unlock the door. Ports 12 and 13 are used to provide the RFID with a input and output signals respectively (for read and write operations). Finally, port 11 is used to supply the servo motor with a PWM. The PWM supplied is also determined programmatically via software, and will control the angle at which the servo motor will hold the deadbolt steady (explained further in 2.5.8 - Lock Mechanism section).

2.4.3 Control Logic

The control logic is split into two sections. Figure 5 is the logic diagram for condensing the signals that control the unlocking of the door into a single signal. This then feeds into Figure 6 where this signal is used with four other signals to determine whether the door should be locked or unlocked.

Figure 5: Logic for Determining a Control Logic Input

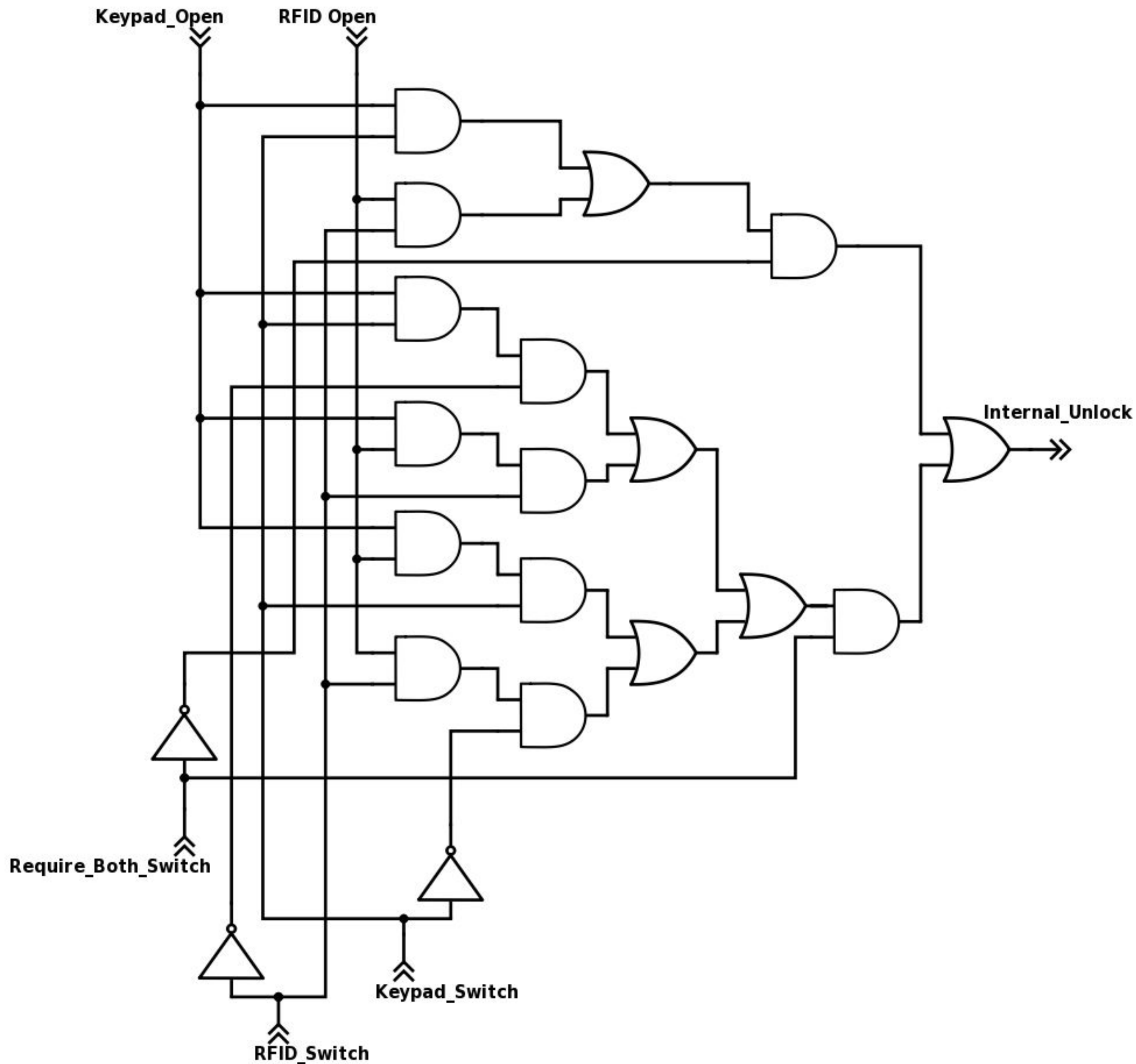
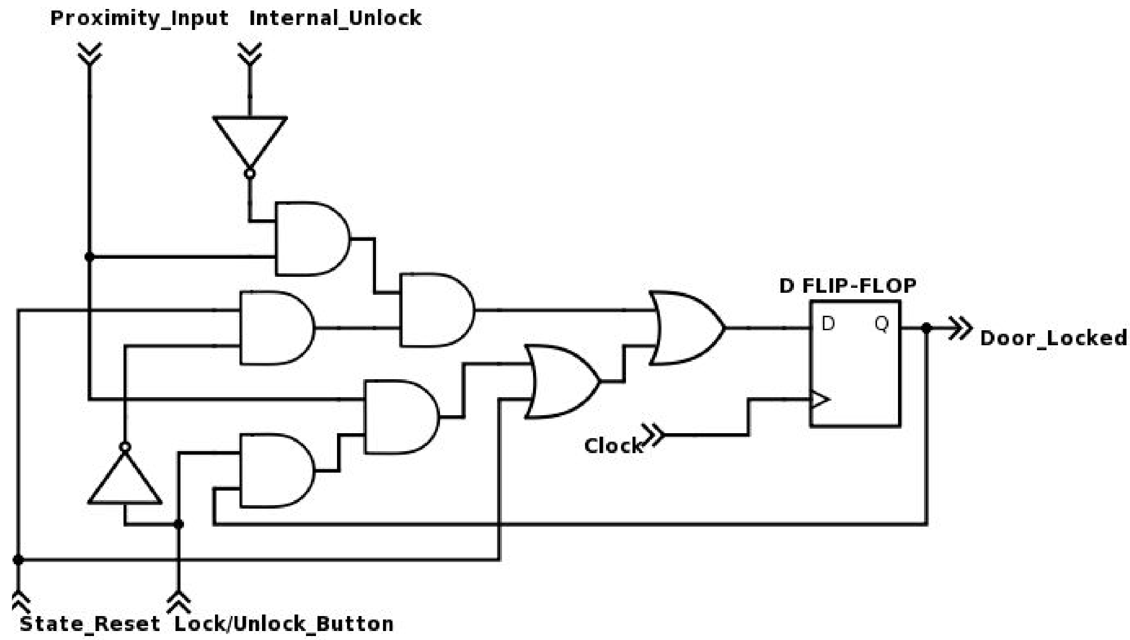
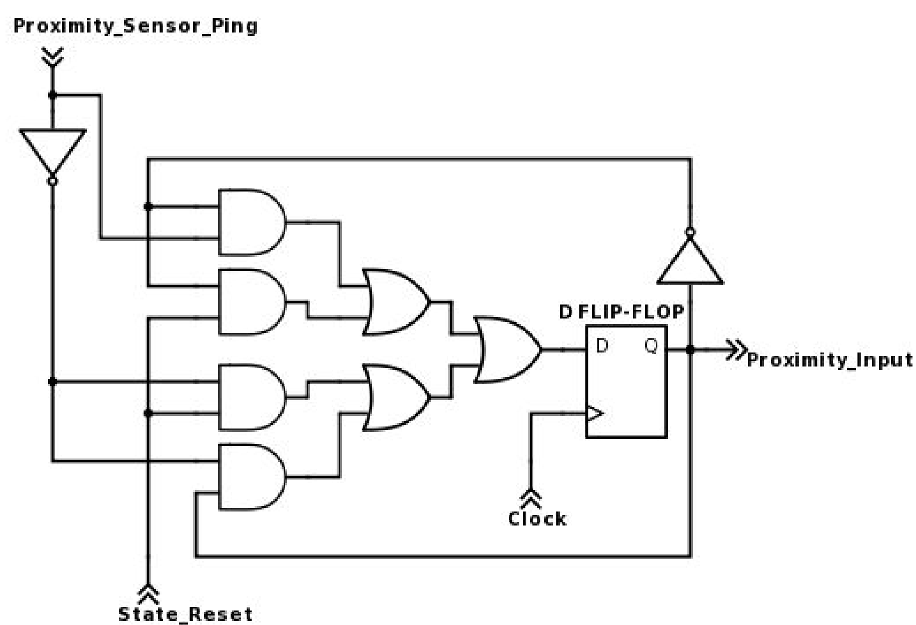


Figure 6: Logic Determining Whether to Lock or Unlock the Door



2.4.4 Proximity Sensor Interface Logic

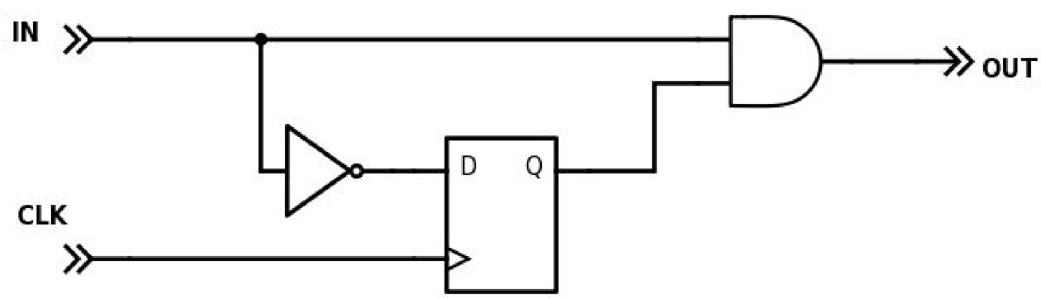
Figure 7: Proximity Sensor Interface



2.4.5 UI Interface: Edge Detectors

Pushed buttons propagate a high signal for as long as the button is held down. Therefore the signal will remain high for many clock cycles when sometimes we only need it for one. An edge detector will turn a continuous signal into a pulse with a width of one clock cycle.
(See 2.5.3: UI for application)

Figure 8: Button Edge Detector



2.5 Block Design

2.5.1 RFID

The RFID scanner acts as one of the two input options. The scanner will read a nearby tag and send the tag ID bytes to the microcontroller which will interpret the data and compare it to the acceptable values to determine if the given RFID tag has permission to unlock the door. (See the microcontroller block for more details about interpretation).

Table 5: RFID Scanner R&V

Requirement(s)	Corresponding Verification Method(s)
1] Scan RFID tags at a maximum range of 2.5-3 inches 2] Go into a low-power mode when not in use to conserve power (less than 0.05 W). Active power should be between 0.5W and 1.0W	1a] Set up an test arduino to read the tag data. The circuit must be able to read 4 bytes of data from the tag. 1b] Start with tags far from the scanner. 1c] Move a tag 3 inches from the scanner. This might scan. 1d] Move the tag 2 inches from the scanner. This and anything closer should definitely scan. 2a] Attach the scanner to an oscilloscope in order to measure power consumption 2b] Record the power consumption when no tags are in range. This is the idle power, compare to expected value of < 0.05W 2c] Scan a tag multiple times and record the power consumption. Verify the active power spikes against the expected 0.5W - 1.0W

2.5.2 Keypad

The second input method is a keypad that is the interface for a user to enter a 6 - 8 digit PIN. There are 16 buttons, and pushing a button will connect the wires associated with that particular button's row and column. The keypad's input/output interface is 8 bits (4 columns and 4 rows) and the whole byte is sent inside the door for analysis and verification. A simple hardware circuit will interpret which key is pressed and send that info (4 bits) to the microcontroller. The microcontroller will keep a queue of the most recently pushed buttons to compare to the acceptable PINs. (See the microcontroller for more details about interpretation).

Table 6: Numerical Keypad R&V

Requirement(s)	Corresponding Verification Method(s)
1] Must be able to output high V signal indicating which button was just pressed. 2] Must be easy to push the buttons	1a] Hook up the keypad to an oscilloscope or multimeter 1b] Power the row pins with 5V 1c] Push every button and verify the correct column pins have between 4.5 - 5.5V 2] Ensure the buttons are easy to press without strain or error

2.5.3 UI

The user interface on the inside of the door is a set of switches that the user can use to manage the system. Both the RFID and keypad will have a corresponding switch (that marks that method as “in-use” or “not in-use”. A third switch will turn on an AND mode that would require both a correct PIN AND RFID card to open the door. When this switch is turned off, the circuit is considered to be in OR mode. In this mode, either a correct PIN OR RFID card can be used to enter the door. These three bits of info are sent to the state machine to determine when to send the “open” signal to the lock mechanism.

There will also be reset buttons (one per input method) that the user will press to change the acceptable PIN and RFID codes. We will require that the correct corresponding authentication is received before the reset buttons will work. There will be a initial reset button that will align the state machine with the actual physical state. Pushing this button will set the state machine to the closed and locked state. The fourth and final button (lock/unlock) will let the user lock and unlock the door, making it easier for the system to maintain the state machine and for the user to interface with the device.

When pushed, the buttons connect the output pin to the 5V pin and output current until the button is unpressed. We only want the buttons to send short pulses so they will each pass through an edge-detector.

Table 7: User Interface R&V

Requirement(s)	Corresponding Verification Method(s)
1] Every switch and button should alternate between low (0V - 0.7V when off or unpressed) and high (4.5V - 5.5V when on or pressed) 2] Buttons passing through the edge-detector should create a pulse as long as the clock cycle. 3] Must be intuitive for the user and properly labeled.	1a] Test each switch and button individually 1b] Start the item in its low state (unpressed or off). Verify the output voltage is between 0V and 0.7V. 1c] Change the item to its high state (pressed or on). Verify the output voltage is between 4.5V and 5.5V. 2a] Build a button in series with an edge detector. Use an oscilloscope to measure the output of the edge detector. 2b] Use a function generator to make a square wave clk signal. 2c] Push the button down (don't let up). The output should be a single pulse with the same width as the clk signal. 2d] Test with many clk signal widths 3] Show the UI to people not involved in the project and ask what each part does. If the majority answers incorrectly, we need to redesign that part.

2.5.4 Proximity Door Sensor

The proximity sensor will be placed at the top of the door on the inside of the room. It consists of a magnetic tag which will be fixed on the frame of the door and a sensor which will be fixed on the door itself. Whenever the magnetic tag comes in range of the sensor (at least 0.25 to at most 0.70 inches[3]) or leaves the range of the sensor it sends a pulse signal to the state machine.

When the tag comes in range we know the door is closing and the state machine will send the "lock" signal to the motor block. When the tag goes out of range of the sensor we know the door is open so the door will not lock. The proximity sensor is important to add the automatic locking feature (i.e. when the door is closed for a certain period of time, it will automatically lock).

Table 8: Magnetic Proximity Sensor R&V

Requirement(s)	Corresponding Verification Method(s)
1] The sensor outputs a short 0.5ms pulse every time the two parts separate or meet	1a] Start with the tag far from the circuit. 1b] Move the tag towards the sensor, the sensor should output a short pulse between 4.5 - 5.5V 1c] Take the tag away from the sensor and it should output the same pulse: 4.5 - 5.5V 1d] Each pulse length should be 0.4 - 0.6ms long

2.5.5 Controller

The controller acts as an interpreter of the raw input module data. The inputs are the keypad module byte and the RFID module bytes, and also the reset_rfid and reset_keypad bits from the UI. The controller outputs two bits: rfid_correct and keypad_correct.

The keypad buttons are pushed sequentially so the controller will store in a queue the most recent 8 button presses. After each press it will check the queue against the correct PINs. If it matches, keypad_correct will become a high signal. All RFID bytes are taken simultaneously so they can immediately be compared to the correct values.

Lastly, the controller uses the two reset bits from the UI module to update which PIN and RFID tags can open the door. After a reset button is pressed, the next input in the corresponding method will be added to the correct values list. For example, if one presses “reset_keypad”, the next PIN they enter will now be a valid PIN to open the door. After 5 seconds of no input, the reset will be stopped and remain unchanged.

Table 9: Controller R&V

Requirement(s)	Corresponding Verification Method(s)
1] Chip outputs low signal (0V - 0.8V) when no or incorrect output is given 2] Chip outputs high signal (4.5V - 5.5V) when a correct input is given	1a] Set up a test circuit with saved values for PIN and RFID numbers that are correct 1b] Input incorrect PIN and scan the wrong tag. Output should be low (0V - 0.8V) 2a] Input correct PIN. Output should be high (4.5V - 5.5V) 2b] Scan correct tag. Output should be high (4.5V - 5.5V)

2.5.6 State Machine

The state machine is what makes the decision to unlock or lock the deadbolt. Input data comes from the RFID Sensor, the keypad, the UI (all switches and the state_reset and lock/unlock buttons) and the proximity sensor. The output of the block is a single bit that is sent to the servo interface. High indicates lock, low indicates unlock.

The UI bits will determine which of the inputs must be correct for the door to open. A correct PIN won’t open the door if the keypad_switch is off. During the AND mode operation (when the keypad and the RFID must both be entered) when the microcontroller determines that one of the two inputs is correct, it will need to remember that decision for 5 seconds. During this time, if the second input gets an affirmative signal the door will open, otherwise it will reset everything back as if nothing was entered.

In total, the state machine is broken down into two sections, the unlock signal, and the unlock/lock state. The unlock signal is determined through a combination of the switches, and the input data from the RFID and keypad as shown in Figure 5 and Table 1 and 2. The resulting unlock signal is then output to the unlock/lock state circuit. Using a combination of the unlock signal, the state_reset and lock/unlock buttons, and the proximity sensor output, we determine whether the circuit should next be in the lock or unlock state.

Table 10: State Machine R&V

Requirement(s)	Corresponding Verification Method(s)
1] RFID and Keypad Inputs output a 2.7V-5.5V signal when they the corresponding switches are turned on and between -0.5V-0.8V when an incorrect combination is input. 2] The Unlock/Lock state must be between 2.7-5.5V when the bit is high and -0.5V-0.8V when the bit is low. 3] Use proximity sensor to lock the door	1a] Set up the circuit as shown in Figure 5. 1b] Apply 5V inputs in every combination shown in Tables 1 and 2. 1c] Ensure that between 2.7-5.5V are output for combinations where a 1 appears in Tables 1 and 2 and that between -0.5-0.8V are output where there is a 0. 2a] Set up a circuit as shown in Figure 6. 2b] Test every combination of signals in Table 3. Ensure that the end voltage is 2.7-5.5V where a 1 shows up in the table and between -0.5-0.8V where a 0 shows up on the table. 3a] set the circuit created in 2a to a state where having the proximity sensor input at 0V puts the state to unlocked, and changing that same input to 5V would put the state to locked without changing any other inputs. 3b] Ensure that the state changes from -0.5-0.8V output to 2.7-5.5V output.

2.5.7 Power Supply

The power supply will consist of five components. These are the AC Voltage Source, a Lithium Ion Charger, a Lithium Ion Battery, a voltage regulator, and a power led. The AC Voltage source is simply a wall plug to provide a recharging source to the battery. It is not used at all to power the circuit. The lithium ion charger is used to recharge the battery supply within the the system. This is to ensure that the battery can last indefinitely. The lithium battery supply is used to power the circuit. This ensures that there are no security concerns from the wall power being cycled and creating errors in our system. The voltage regulator works as the name suggests and ensures that the circuit receives the correct voltage input. The power led will only turn on when the power is low to signal to the user to either plug it into the wall, or change the batteries.

We are currently planning to use 5 NiCD D batteries to power this circuit. Each of these have a voltage of 1.2V, so 5 of them would be adequate to cover our voltage needs. These can have approximately 5000mAh of capacity to them. The length of time that we can run our circuit assuming a 23 mA current while idle and without the power supply is as follows. The 23 mA is an estimate of idle power consumption based on the components we currently have:

Equation 1

$$5000mAh / 23mA = 217\ hours$$

From Equation 1 we see that we can power our circuit for about 9 days on idle without wall power.

The value of the resistor in the recharging circuit is set such that the difference in voltage between the maximum for the battery and the power supply divided by the capacitance of the battery over 10. This equation is listed below.

Equation 2

$$(9V - 6V)/(5/10) = resistance = 6\Omega$$

So for the 6V battery, 9V power supply, and 5000mAh power supply, we get a needed 6 Ohm resistor to limit the current into the battery. This value must be adjusted for the allowed manufacturer variance as described later in the Tolerance section.

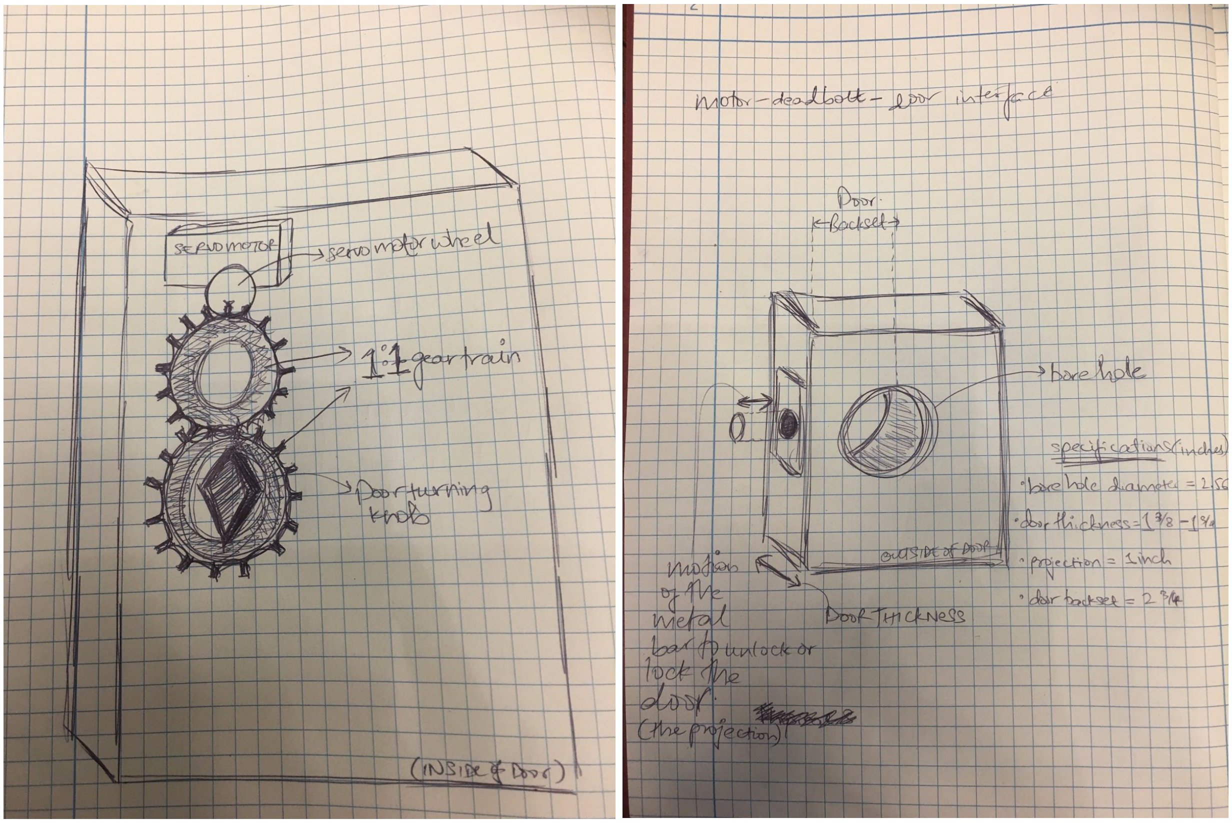
Table 11: Power Supply R&V

Component	Requirement(s)	Corresponding Verification Method(s)
Power Supply	1] A function generator supplying 220V AC, similar to that provided by a wall unit, powering the circuit must be able to output 9V DC from the terminals of the voltage regulator (minus the voltage drop across the LED).	1] Use a function generator to apply the 220V AC circuit across the power supply, and measure the voltage across the voltage regulator to ensure it is at least 8.3V DC (0.7 V is the estimated voltage drop from the LED).
Battery Charger	1] Voltage applied across the entire combination of batteries is greater than 6V when plugged into the wall outlet.	1] Use a voltmeter to estimate the voltage across the batteries when the battery charger is plugged into the wall. The reading must be 6V or more.
Battery	1] Should be able to maintain a voltage of at least 4.8V to the circuitry inside and outside the door 2] Should be able to suppress any errors from a battery inserted backwards. These batteries should have no impact on the system	1] Use voltmeter when battery is being drained and measure potential difference across ends. Reading must be 4.8V or more. 2] Insert a battery backwards and use an ammeter to test current flowing from diode D2 in Figure 3 to input pin of the voltage regulator in Figure 3. Ensure that the ammeter reading is 0 A in this situation.

2.5.8 Lock mechanism

The lock mechanism is the physical part that unlocks and locks the door using a servo motor (model 31311S HS-311) fixed to the deadbolt on the inside of the door using a 1:1 gear train. One of the gears will be fixed to the servo, and the other will fit on top of the deadbolt The system takes in two bits, one to lock and one to unlock the door. The servo motor is capable of turning 180 degrees and turns based on the supplied current. At 4.8 V, the torque output is 3kg/cm and the required torque to rotate the deadbolt is ~1kg/cm. The motor runs on PWM (pulse wave modulation) to set itself in different positions or angles or rotation. The motor needs a continuous supply of any voltage between 4.8V and 6V. Our circuit will supply 5V to the motor, and this will be constant (regardless of whether the motor is moving or not.) Another port of the motor will need the PWM as input and will be between a range of 575 and 2460 µseconds. The PWM time period needs to change by approximately 10.5 µseconds to increase the motor’s rotation angle by 1 degree.

Figure 9: Sketches to Describe the Mechanical Component of the Door Lock
(left: inside of door, right: outside of door)



The above 2 drawings illustrate the mechanical aspect of the door lock and its interface with the motor. On the inside of the door, we can see that the servo motor is connected to the turn knob by 2 gears of the same size (1:1 gear train). This will enable the user to freely operate the turn knob without having to physically touch the servo motor. On the outside of the door, there will be a hole for the locking unit to fit into, and the projection will protrude out of the door by about 1 inch.

Table 12: Lock Mechanism R&V

Component	Requirement(s)	Corresponding Verification Method(s)
Lock Mechanism, associated gear train, and servo motor	1] Motor is capable of moving 90 degrees which would need a range of PWM from 575 to 1517 µseconds time period to execute the rotation and the motor does not rotate back to it's original position. Motor must move to the correct position when a function generator creates these waves as PWM inputs at a voltage of 5V. 2] Power is conserved when the motor is not in motion. In our case, the power consumed by the servo motor when not in use is at most 0.0375 W.	1] Using an oscilloscope, measure the PWM passing through the port for the servomotor controlling the angle of rotation. At 575 µseconds PWM the motor should be is in its neutral position (0 degrees or unlocked). At 1517 µseconds PWM, the motor should be in a locked position (or 90 degrees rotated). 2] Use a voltmeter and an ammeter, measure the voltage across the 2 power ports of the servo motor and the current flowing into the servo. The motor does not draw more than 7.5 mA when powered with 5V when the motor is stationary - the numbers used for our power calculations in Equation 1.

2.6 Tolerance

2.6.1 Battery

The battery that we are using requires that we have a charging current of less than c/10 where c is the capacitance in Ah. As such, we will need to select our resistor that limits the current to be at a value such that our battery accepts it. Say we have a wall DC converter that outputs a 9V dc voltage. Also assume that we have a 6V, 5000mAh battery string. In this case we would want to limit the current to 500mA at a maximum.

Equation 3

$$(9V - 6V)/.5A = 6\Omega$$

As such, we would want a 6 Ohm resistor to limit current. Let’s say that this resistor has a 20% variance from the rated values. We can have a lower current than 500mA and have the circuit work still, so lets say that the resistor was 20% below rated resistance. This would put us at having bought a 7.2 Ohm resistor. So knowing the variance in product would give us the minimum acceptable resistance for the resistor here. We must use this calculation to ensure that the current that charges the battery is not in excess of 500mA.

2.6.2 Servo Control

If the PWM changes due to electrical disturbances or voltage fluctuations on the microcontroller, the degree maintained by the servo might be affected. For instance, a short fluctuation of the time period of the PWM will cause the servo to change its position momentarily. It would be easy to identify by watching the servo motor for an extended period of time, and by passing the PWM through a circuit which has a capacitor connected in parallel.

Ar 575 μseconds time period of the PWM, the neutral or 0 degree position should be maintained. If due to manufacturing faults, the corresponding PWM for the 0 degrees position is not the same, our microcontroller would need to be calibrated accordingly. Similarly, according to the data available, a PWM with a time period of 1517 μseconds would correspond to the 90 degree angle, and any deviations from this would need to be adjusted for.

At 575 μseconds PWM, the motor is in its neutral position (0 degrees or unlocked). At 1517 μseconds PWM, the motor is in a locked position (or 90 degrees rotated).

3 Cost and Schedule

3.1 Cost Analysis

Using a rough estimate of $35 per hour salary based off of the average salary of graduated ECE students, 10 hours worked per week, 3 people, and about 60% of the weeks for this class spent on our design, we arrive at the following manufacturing costs for our design:

Equation 4

$$3 * \$35/hour * 10\ hours/week * .6 * 16\ weeks * 2.5 = \$25,200$$

So, our estimated cost to design this product is $25,200. The costs for the prototype are listed below:

Table 13: Parts Table

Part	**Cost (Prototype)**	**Cost (Bulk)**	**Voltage (V)**	**Idle Current(mA)**
RFID Reader	$49.99	$44.99	4.5-5.5	9.4
Lock for the door (for testing purposes)	$12	$0	N/A	N/A
Proximity Sensor	$2.92	$2.02	4.5-5.5	0
Keypad	$19.92	$12.99	4.5-5.5	5
Assorted resistors, capacitors, wires, etc	$10	$0.40	2-5.5	10μ
Battery	$16	$16		
Battery charger	$18	$18		
Battery Casing	$2.45	$2.45	N/A	N/A
Servo Motor	$15	$12	5	7.5
Wall power unit	$7.44	$7.44	N/A	N/A

This comes out to a total of $153.72 for the prototype, and $116.29 when this product is made in bulk. This is a total of $25,353.72 for the total development cost.

3.2 Schedule

Table 14: Weekly Schedule

Week	Tasks to Handle
10/2	- Purchase lock/deadbolt combination and servo motor for the machine shop (Karan) - Turn in design document - Collect all other components and make necessary orders (Mason)
10/9	- Visit machine shop to finalize the mechanical design for the machine shop to work on and provide the lock to the machine shop (Thomas) - Receive all remaining pieces and begin connecting the various components (RFID, pin pad, and servo motor, logic controls, etc.) to understand the devices and PCB requirements. Understand power requirements, and source a suitable battery. (Karan, Mason) - Design review - Begin building blocks (All)
10/16	- Work with TA to solve any identified challenges with the current design. (All) - Begin working on developing our own Arduino, but continue using the commercial Arduino for prototyping and testing purposes. (Mason) - Make a mock design for the PCB based on data sheets of the components and the interfacing needs for the different components. Test with commercial arduino and breadboard, and then move to testing with our own Arduino when complete. (Karan, Thomas) - Check PCB mock design on breadboard with TA and develop finalized PCB design for first prototype. (All)
10/23	- PCB must pass audit by this week - Test all components with PCB and identify problems or remaining requirements to fulfill. Begin to design second PCB to order if necessary. (Thomas) - Test extensively to satisfy core functionality issues and modify if necessary. (Karan) - Begin to fit components onto the actual door frame to begin mechanical robustness and environmental testing. Try to tamper-test the device to understand limits on the physical robustness of the device. (Mason)
10/30	- Send out PCB order and confirm design with TA (All) - Test power requirements, robustness, and stability in times of power-loss. Explore other options for powering the system if necessary. (Karan) - Make changes to the physical design (if more or less space is needed, gear train between servo and deadbolt needs to be changes, etc) and revise the design with the machine shop if necessary. (Thomas, Mason)
11/6	- Final round of PCBway orders go out - Get final mechanical work done and integrated with our devices. (All) - Last day of revision to the machine shop
11/13	- Complete final environmental testing, define power ranges, test maximum life of device without wall socket power supply, and any other metrics of robustness or functionality. (Karan) - Begin working on final project report (Mason) - Begin working on final presentation for project (Thomas)

11/20	- Thanksgiving Break - Send final project report and presentation to TA for revision (All)
11/27	- Complete final presentation after checking with TA for revisions. (All) - Make final edits to report based on TA recommendations. (All) - Mock demo

4 Ethics and Safety

We will be powering our circuit primarily by an AC outlet, with a lithium-ion battery acting as backup. We plan to wire it in parallel with the circuit and AC source so that the battery only draws power when it is not fully charged and only discharges when the AC source is for some reason off or unavailable. Lithium-ion batteries are very useful but they come with a number of considerations. We must never allow the battery to be short-circuited because it can easily overheat. Li-ion batteries also run the risk of thermal runaway if their temperature gets too high; we expect our low-power design to minimize the battery requirement and keep the temperature at a reasonable level. Finally we must make sure that the batteries are not exposed to a voltage outside of the acceptable voltage range by carefully designing the circuit and battery enclosure.

As this project is at its core a lock, we must carefully consider IEEE Code of Ethics 9: "to avoid injuring others, their property, reputation, or employment by false or malicious action."[4] Users will be trusting our design to work to protect themselves and their property from harm and theft. Poor design on our part can expose users to both. We are working to add security to our design in a number of ways. First, only input signals will exist on the outside of the door so the "open" signal cannot be injected anywhere. Next, we will limit inputs such that the controller only takes inputs about once a second. This will prevent anyone on the outside from spamming, for instance, every possible keypad combination until the door opens. We can also lockout devices after a number of failed attempts to further reduce the chance of brute-forcing combinations.

Another concern is malicious resets. For instance, if one could enter the room they could hit the "reset" button and quickly program their card or a PIN so they could access the room later. Unfortunately, access to the reset button is hard to prevent because that access requires one to be inside the room, indicating the user has some level of trust in that person. The most we can do is make the reset require verification first so the only ones that can update the codes is the ones who know the current codes. This system of course still has flaws, but we have to trust that the user will not lose their card, key or write down their PIN anywhere.

5 References

[1] University of Illinois Police Department, ‘Daily Crime Log’, (2017,September 20). [Online] Available: https://illinois.edu/blog/files/7512/557491/118800.pdf [Accessed: 21-September-2017]

[2] Historyofkeys.com,‘History of Locks’, (2017). [Online] Available: www.historyofkeys.com/locks-history/history-of-locks/ [Accessed 20-September-2017]

[3] Arrow Electronics, Inc., ‘Actuator For Magnetic Sensor In Mini Flangemount Housing’, (2017), [Online] Available: https://www.verical.com/datasheet/cherry-electrical-products-unsorted-as201801-24621.pdf [Accessed 20-September-2017]

[4] IEEE.org, ‘IEEE Code of Ethics’, (2017). [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: 21-September-2017]