

# nocLock System Design

T02 – Nov 11th, 2014

---

Travis Berger, Cameron Tribe, Jaime Rodriguez, Sean Koppenhafer

# Table of Contents:

<b>1 Introduction .....</b>	<b>2</b>
<b>2 Block Diagram .....</b>	<b>2</b>
• 2.1 Level (0) .....	2
• 2.1.1 Level (0) & (1) Table.....	3
• 2.2 Level (0) Breakdown .....	5
• 2.2.1 Level (0) Table .....	6
<b>3 Unified Modeling Language Views .....</b>	<b>8</b>
• 3.1 State Machine View .....	9
• 3.1.1 State Table/Description .....	9
• 3.2 Use-Case View .....	10
• 3.2.1 Program Mode .....	11
• 3.2.1 Unlock Mode .....	12
• 3.2.1 Reset Mode .....	12

# 1 Introduction

The Purpose of this document is to give a visual description of the system design from high to low level diagrams and model the behavior seen in the system using a Unified Modeling Language, in different views.

## 2 Block Diagrams

### 2.1 Level (0) and (1):

In the Diagram below we see two different views of the system, Level (0) shows a high-level view with its various inputs and outputs, while Level (1) shows the same with its embedded sub-components.

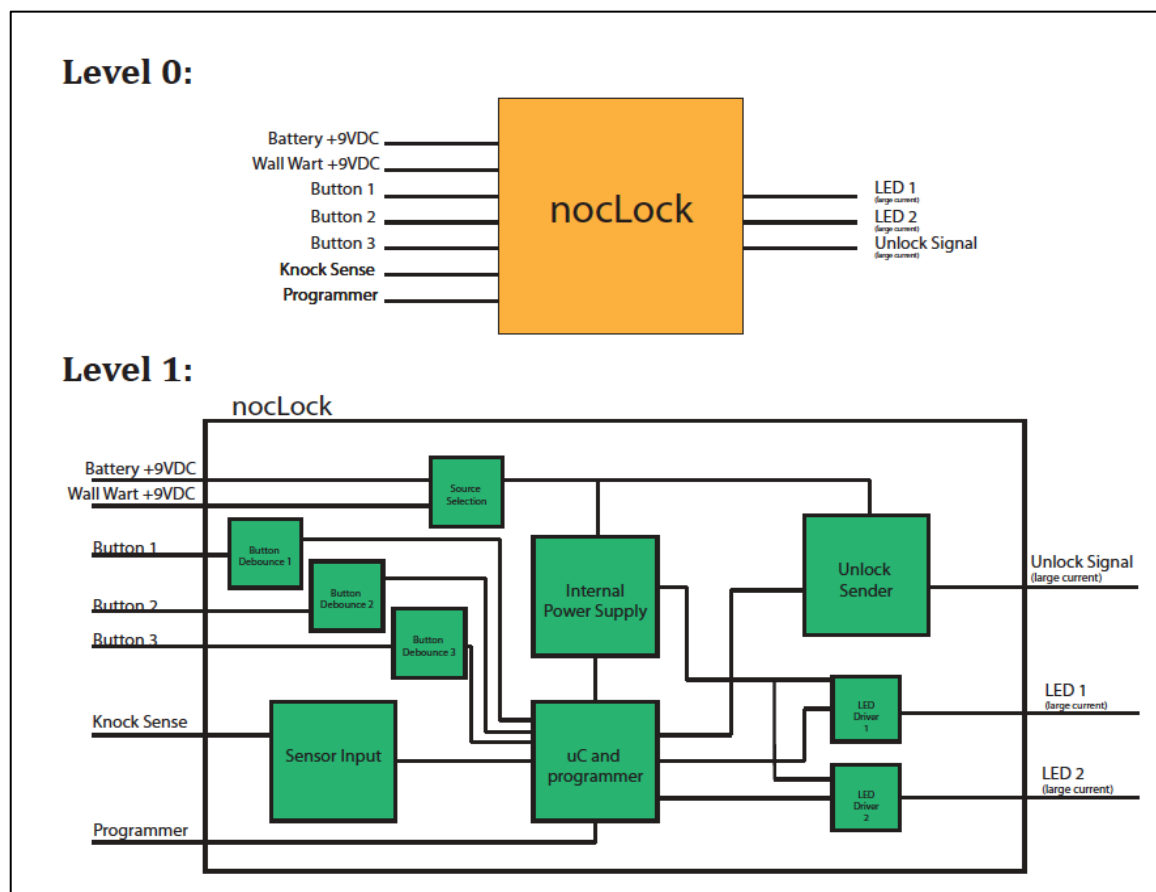


Figure 1. Level (0) and (1) Block Diagram of nocLock System

### 2.1.1 Level (0) Table:

The table below gives a brief description of the functionality of each corresponding input and output described in the Level (0) and Level (1) block diagram.

Level (0)			
<b>Inputs:</b>			
Block Name	Input Name	Input Type	Description
nocLock	Battery +9VDC	Power	A standard 9V battery can be used as a backup in case of power outage.
nocLock	Wall Wart +9VDC	Power	In normal operation, the nocLock uses either a 9VDC AC adapter with a current rating of at least 1500mA.
nocLock	Button 1	Digital	Button 1 will indicate that the user intends to enter a sequence of knocks; the user also uses it to tell the system the end of a knock sequence.
nocLock	Button 2	Digital	Button 2 will indicate that the user intends to write a new knock sequence. It is also used to navigate through the new knock write sequence.
nocLock	Button 3	Digital	Button 3 will indicate the user intends to completely erase all stored knocks and start over. This is intended for debugging purposes as a back door to start over. This button will not be easily accessible.
nocLock	Knock Sense	Analog	This analog voltage will indicate when a knock has happened and will include some protection circuitry to reduce any voltages higher than the tolerance of the microcontroller.
nocLock	Programmer	Data	The programmer input is a connection to another programmer, to allow the microcontroller to be programmed. This input will be composed of several lines.
<b>Outputs:</b>			
Block Name	Output Name	Output Type	Description
nocLock	LED 1 (high current)	Digital	This will control a LED of a single color to indicate where the user is in the program and it is used for visual queues for confirmation or denial.
nocLock	LED 2 (High Current)	Digital	This will control a LED of a single color to indicate where the user is in the program and it is used for visual queues for confirmation or denial.
nocLock	Unlock Signal (High Current)	Digital	The unlock signal will most likely be driving an inductive load that requires a large amounts of current to operate.

Table 1. Level (0) input and output descriptions

Level 1	
Block Name	Description
Button De-bounce n	Button de-bounce will be responsible for removing and "mechanical jitter" or bounces to occur. This assures that the microcontroller see only one state change over the duration of the button press. There is only a single wire output, referenced to ground. There will be a total of 3 button de-bounce systems, one per button.
Internal Power Supply	The internal power supply is used to step down the supply voltage and remove any line transients. It will need to be capable of providing power to the microcontroller and up to 2 LEDs.
Sensor input	Depending on the sensor, the sensor input will be used to protect the microcontroller from any harmful voltage levels. It will be responsible for either level shifting a digital signal, or amplifying/clipping and analog signal. It will have a single wire output to the microcontroller.
uC and programmer	The microcontroller and programmer system will be responsible for interpreting all the button presses, knocks and also for changing the state of the LEDs and the unlock sender.
LED Driver n	The LED driving system will be responsible for taking the small amount of current from the microcontroller and turning it into a larger current from the power source, that is capable of illuminating an LED. There is a single wire output that connects to an LED.
Unlock Sender	The unlock sending system is similar to the LED Driving stage. However it needs to be capable of driving much more current to a device that locks and unlocks the box.
Source Selection	The source selection system is capable of automatically switching between either wall wart power supply or the 9V batter. The output is a single wire to the Unlock sensor and the internal power supply.

Table 2. Level (1) input and output signal descriptions

### 2.1.2 Level (0) Breakdown:

In the Diagram below we see a level (0) view of all the sub-components with their corresponding signals broken down.

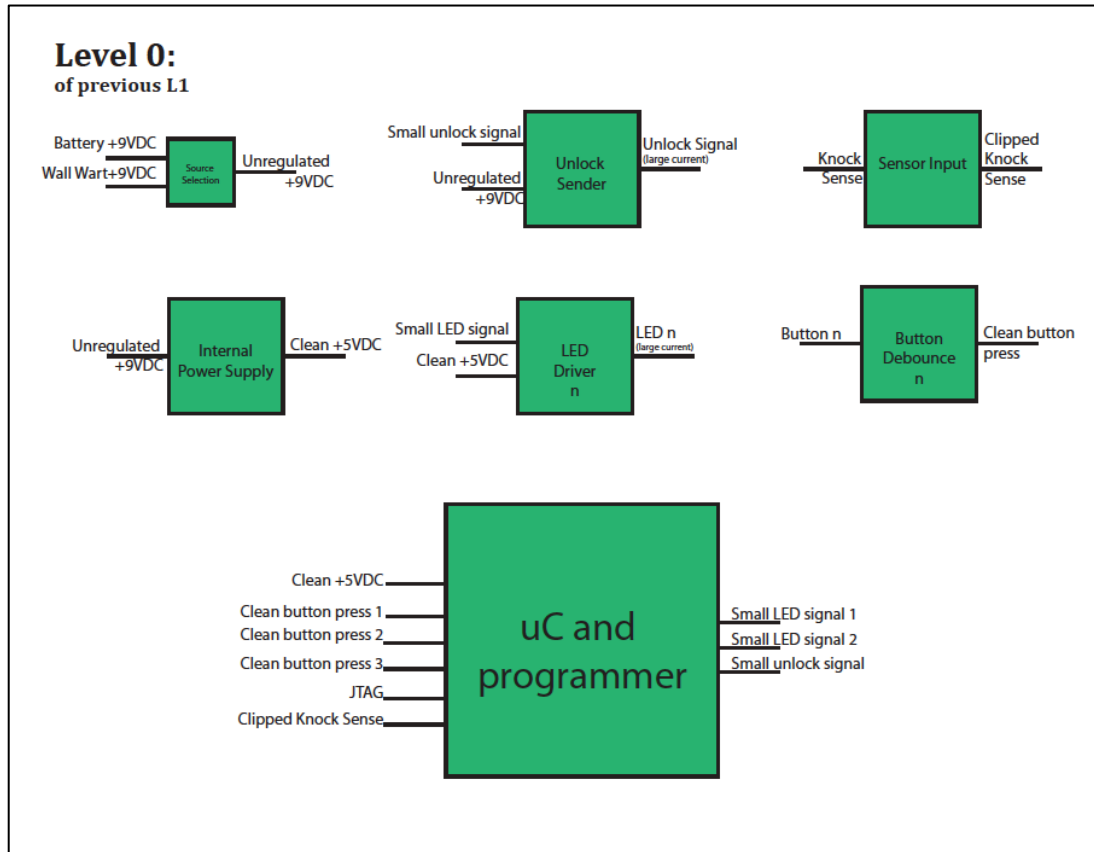


Figure 2. Level (0) Block Diagram of nocLock System and sub-components

Level (0)			
Inputs			
Block Name	Input Name	Input Type	Description
Source Selection			
	Battery +9VDC	Power	Input can be any standard 9V battery-. It is a single wire, referenced to ground.
	Wall Wart +9VDC	Power	Input can be any 9VDC AC adapter, capable of supplying at least 1500mA. It is a single wire referenced to ground.
Internal Power Supply			
	Unregulated +9VDC	Power	This input is the single wire output of the Source Selection system.
LED Driver n			
	Small LED Signal	Digital	This input is an output of the microcontroller, and is expected to be a small current.
	Clean +5VDC	Digital	This input will be connected to the output when the other input is asserted.
Unlock Sender			
	Small Unlock Signal	Digital	Similar to small LED Signal above.
	Unregulated 9VDC	Digital	This input will take the larger main power supply and apply it to a device that unlocks the box. It is one wire, referenced to ground.
Sensor Input			
	Knock Sense	Analog	This input is the raw sensor voltage. It is assumed that this signal will have large voltage spikes that can harm the microcontroller. It is one wire, referenced to ground.
Button De-bounce			
	Button n	Digital	The input is a rising or falling edge that can be followed by many more of the same. This can be confusing to the microcontroller and may cause unpredictable behavior. It is one wire, referenced to ground.
uC and programmer			
	Clean +5VDC	Power	This is the power supply to the microcontroller. It is the output of the internal power supply.
	Clean button press 1	Digital	The output of a button de-bounce stage. It is either a rising or falling edge that indicates a button press.
	Clean button press 2	Digital	See above
	Clean button press 3	Digital	See above
	JTAG	Data	6 lines total, 4 of those are MOSI, MISO, SCK and Reset. The remaining 2 are VCC and GND. This is used to send program data to the Microcontroller.
	Clipped Knock Sense	Analog	A safe voltage level that indicate the presence or absence of a knock.

Level (0)			
<b>Outputs</b>			
<b>Block Name</b>	<b>Output Name</b>	<b>Output Type</b>	<b>Description</b>
<b>Source Selection</b>			
	Unregulated +9VDC	Power	Output is a single unregulated 9VDC source. It is one wire that is referenced to ground.
<b>Internal Power Supply</b>			
	Clean +5VDC	Power	The output is a clean, regulated voltage that is safe for the microcontroller to use. Its is a single wire, referenced to ground.
<b>LED Driver n</b>			
	LED n (large current)	Digital	This output will be a larger signal that is capable of illuminating an LED. It is a single wire, referenced to ground.
<b>Unlock Sender</b>			
	Unlock Signal (large current)	Digital	The output will drive a unlocking device. It will most likely be some inductive load, needing a relatively large amount of current.
<b>Sensor Input</b>			
	Clipped Knock Sense	Analog	The output is a voltage level that has a maximum value that is less than that which can harm the microcontroller I/O pins. It is one wire, referenced to ground.
<b>Button De-bounce</b>			
	Clean button press	Digital	A single clean rising or falling edge so that the microcontroller can act in a predictable way.
<b>uC and programmer</b>			
	Small LED signal 1	Digital	A small signal asserting the voltage level of and LED.
	Small LED signal 2	Digital	See Above
	Small unlock signal	Digital	A small signal indicating that the locking mechanism should be engaged or disengaged.

Table 3. Level (0) description of input output functionality



### 3 Unified Modeling Language Views

The Purpose of the UML Views seen in this section are to describe the functionality expected in the nocLock system, Give a visual representation and description of the behavior seen in each state as well give an explanation of each use-case scenario a potential user might encounter.

#### 3.1 State Machine View:

The State machine below helps to visual the behavior expected in the system. While supporting the functionality and decomposition of the sub-components seen the block diagrams above.

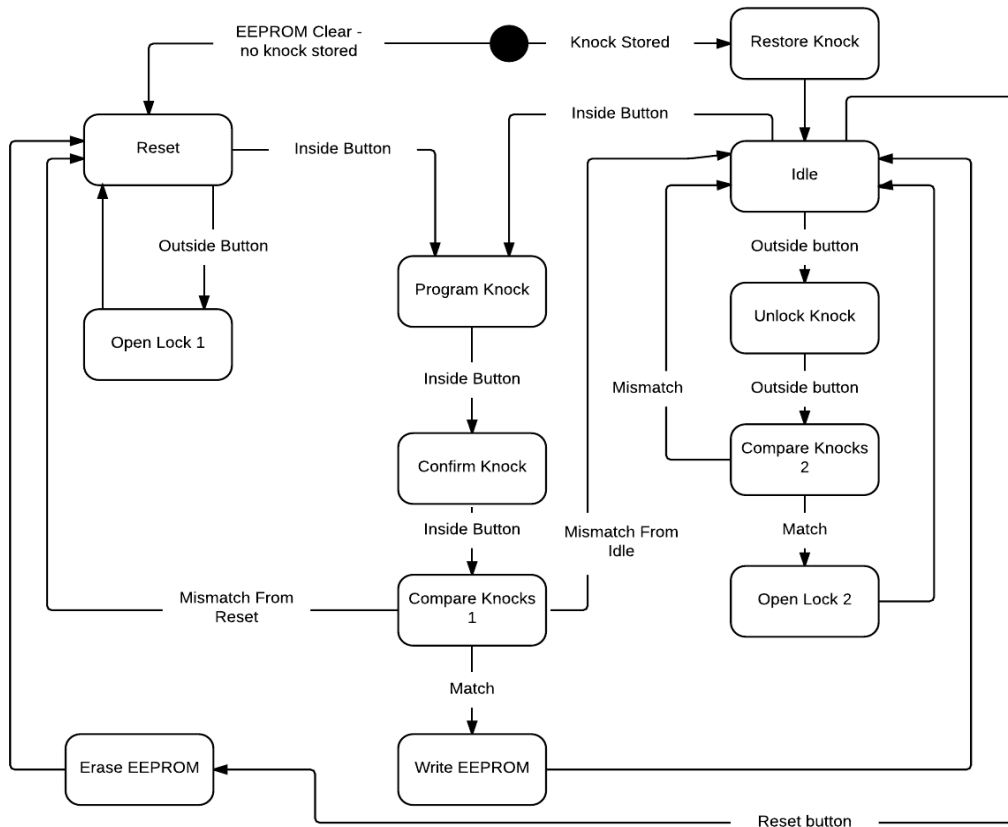


Figure 3. State Machine View of nocLock System

### 3.1.1 State Table/Description:

The Table below states the corresponding inputs and outputs of the state machine above, and gives a brief description of those ports functionality.

State	Inputs	Outputs	Function
Reset	- Outside button - Inside button	- Red LED on - Green LED off - Solenoid closed	No current knock is stored. Allows for the door to be opened without inputting a knock.
Open Lock 1	- None	- Red LED off - Green LED on - Solenoid open	Opens the lock for 5 seconds and turns the green LED on to signal that the door is open. Returns to reset after 5 seconds.
Program Knock	- Inside button - 5V analog ADC "knock" value	- Red LED on* - Green LED on* - Solenoid closed	Checks the ADC value for a knock (defined as a value above a certain thresh hold). Stores the time of each knock in milliseconds. When the inside button is pressed, the sequence is over.
Confirm Knock	- Inside button - 5V analog ADC "knock" value	- Red LED on* - Green LED on* - Solenoid closed	Briefly, only the green LED is on to signal that the knock was recorded and now the user must confirm their knock. This function does exactly what Program Knock does, recording the sequence to a new area.
Compare Knocks 1	- Program sequence and confirm sequence	- If records match, green LED on - If no match, red LED on - Solenoid closed	Compares the program knock and the confirmation knock. If the knocks fall within a certain thresh hold of each other, the knocks match. If they do not match, return to either reset or idle, depending on which one entered the program state.
Write EEPROM	- Program knock sequence	- Red LED off - Green LED on - Solenoid closed	Writes the newly programmed knock sequence to EEPROM in case the micro controller loses power.
Restore Knock	- None	- Red LED on - Green LED off - Solenoid closed	Restores a knock sequence from EEPROM after a micro controller restart.
Idle	- Inside button - Outside button - Reset button	- Red LED on - Green LED off - Solenoid closed	Sits idle and waits for one of the three buttons to be pushed
Unlock Knock	- Outside button - 5V analog ADC "knock" value	- Red LED on* - Green LED on* - Solenoid closed	Much like Program Knock, listen for an ADC knock value above the thresh hold and record the time. When the outside button is pressed, the sequence is done.
Compare Knocks 2	- Programmed knock sequence and unlock sequence	- Red LED on - Green LED off - Solenoid closed	Compare the programmed sequence to the unlock sequence in the same way as compare knocks 1. If the knock times fall within a thresh hold of each other, the box is opened.
Open Lock 2	- None	- Red LED off - Green LED on - Solenoid open	Opens the lock for 5 seconds and turns the green LED on to signal that the door is open. Returns to idle after 5 seconds.
Erase EEPROM	- None	- Red LED on - Green LED off - Solenoid closed	Erase the programmed knock sequence in EEPROM and return to reset. This is a debugging feature only.

Table 3. State Machine input, output functionality descriptions

\* Table note - we are using a bi-colored LED, Having green and red on at the same time creates yellow.

### 3.2 Use-Case View

This view defines the operation of the nocLock system and all Potential use-case scenarios a user can encounter.

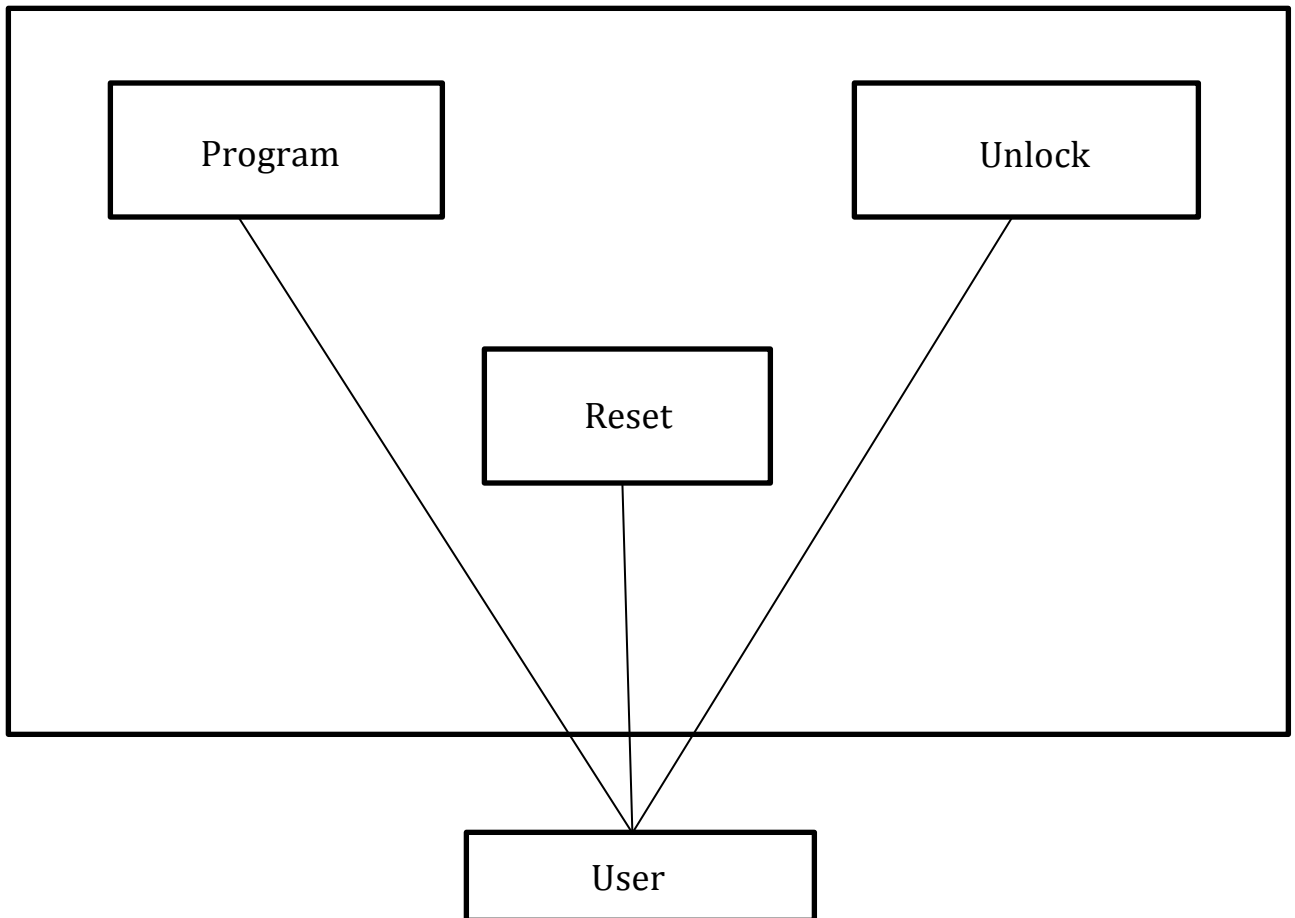


Figure 4. Use-Case View of nocLock System

### 3.2.1 Program:

Use-Case	Program
Actor	User
Summary	User programs a unique knock sequence that will later be used to unlock the device. The user opens the device and enters program mode by pushing the program button. The user enters the new knock sequence, presses the program button, and then enters the new knock sequence again to confirm the sequence. If the knock sequence is the same for both entries the yellow LED will turn green. If the knock sequences do not match the LED will turn red.
Pre-Conditions	1. The device is open allowing access to the program button.
Normal Flow of Events	<ol style="list-style-type: none"><li>1. User opens device.</li><li>2. User presses program button.</li><li>3. LED turns yellow prompting user to enter new knock sequence.</li><li>4. User enters new knock sequence.</li><li>5. User presses program button.</li><li>6. LED turns green to confirm entry of knock sequence.</li><li>7. LED turns yellow prompting user to confirm new knock sequence.</li><li>8. User enters knock sequence from step 4.</li><li>9. User presses program button.</li><li>10. LED turns green, showing that the new knock sequence was stored.</li></ol>
Variations	<ol style="list-style-type: none"><li>1a. User cannot open device</li><li>10a. LED turns red notifying the user that the knock sequence was incorrect and not programmed to memory.</li></ol>
Post Conditions	1. New knock sequence is stored in memory and EEPROM.

Table 4. Program Mode description of operation

### 3.2.2 Unlock:

Use-Case	Unlock
Actor	User
Summary	User enters the correct knock sequence and opens the device. To enter the knock sequence user must press the action button, enter the knock sequence, and press the action button again to submit the knock entry. If the entered knock matches the programmed knock, the LED will turn green and the device will unlock for five seconds. If the entered knock doesn't match, the LED will turn red and the device will remain locked.
Pre-Conditions	1. Knock sequence has been programmed.
Normal Flow of Events	1. User presses action button. 2. LED flashes yellow prompting user to enter knock sequence. 3. User enters knock sequence. 4. User presses action button to submit knock sequence. 5. LED flashes green and device unlocks for five seconds.
Variations	2a. No knock is stored, green LED flashes and device unlocks after action Button is pressed. 5a. LED turns red notifying user that the knock sequence was incorrect and Device stays locked.
Post Conditions	1. User should close device door.

Table 5. Unlock Mode description of operation

### 3.2.3 Reset:

Use-Case	Reset
Actor	User
Summary	Hitting the reset button resets the device. EEPROM is erased and the device is restored to its original state.
Pre-Conditions	1. The device is open and the user has access to the reset button.
Normal Flow of Events	1. User presses the reset button. 2. Device is reset.
Variations	N/A
Post Conditions	1. EEPROM is erased.

Table 6. Reset Mode description of operation