[Locked In] Phone Lock Box: EECS 373 Embedded Systems Project

Team 9 - Isabella Longo, Rachel Nowak, Colin Riker, Aryan Thomare {blongo, nowakra, colinrik, athomare}@umich.edu

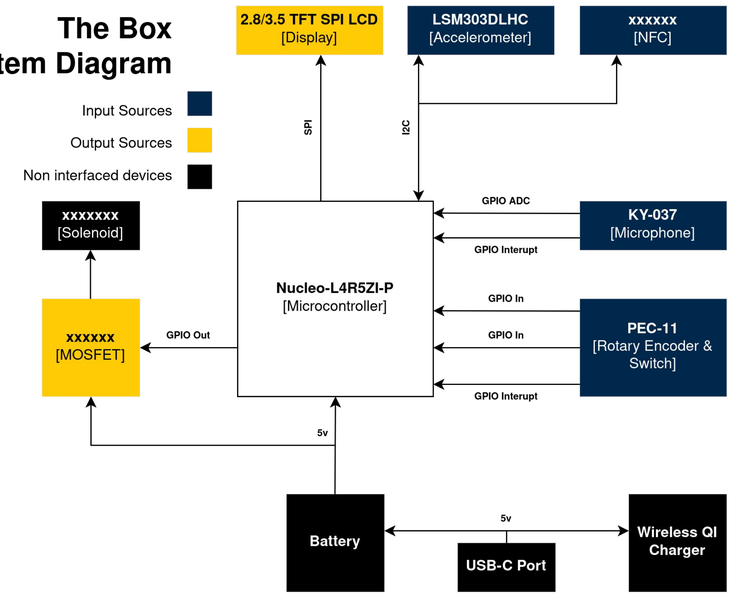
MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Background and Motivation

Our project is a phone lock box!

- Functionally **intuitive** and **simple** to use
- Works as a study tool to lock a phone away to **prevent distractions** for a set time period
- Embedded ringtone override for emergencies
- Designed to reduce distraction while keeping **real-life usability** in mind

The Box System Diagram



The High Level View

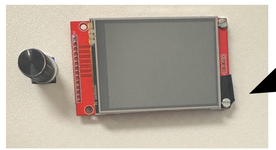
- **Single user input** device: a **rotary encoder** that functions as both a dial and a push-button.
- The **LCD screen** provides all user-facing information and feedback for functionality.
- **Wireless charging passthrough** when a charging cable is connected.
- Built-in **emergency override** system, including:
  - **Audio detection** of a preset ringtone
  - **Manual lock forced override**
- NFC for **phone detection** confirms placement before locking.

Going Forward

- **App Integrations** (Google Calendar, Canvas, etc)
- Develop custom **audio unlocks** (Raspberry Pi integration).
- Create a companion **phone app** for setting lock duration, notifications and override settings.

Mapping Function to Hardware: User Functionality

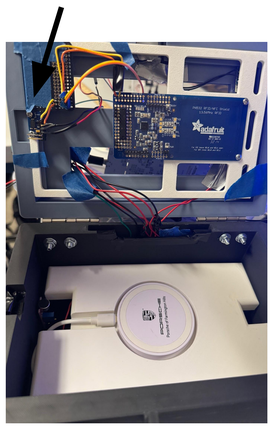
Our team utilized s**ystem and state diagram**s to map our projects functionality to output device actions. These diagrams detail our **hardware connections** and **state transitions**, and the flowchart below outlines how a user would interact with the box.



The rotary encoder or accelerometer **'wakes up'** the screen.

The screen then prompts the user to:

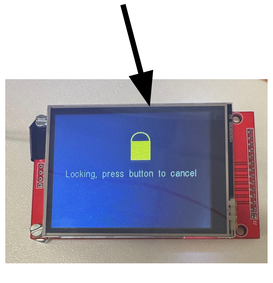
- **Place their phone** in the box (via NFC connection)
- **Close the box** (confirmed by magnetometer)
- **Set the timer** (using the push button)
- **Lock the box** (which triggers solenoid)





The locking stage has a brief waiting period, allowing the user to adjust the timer if it was set incorrectly.

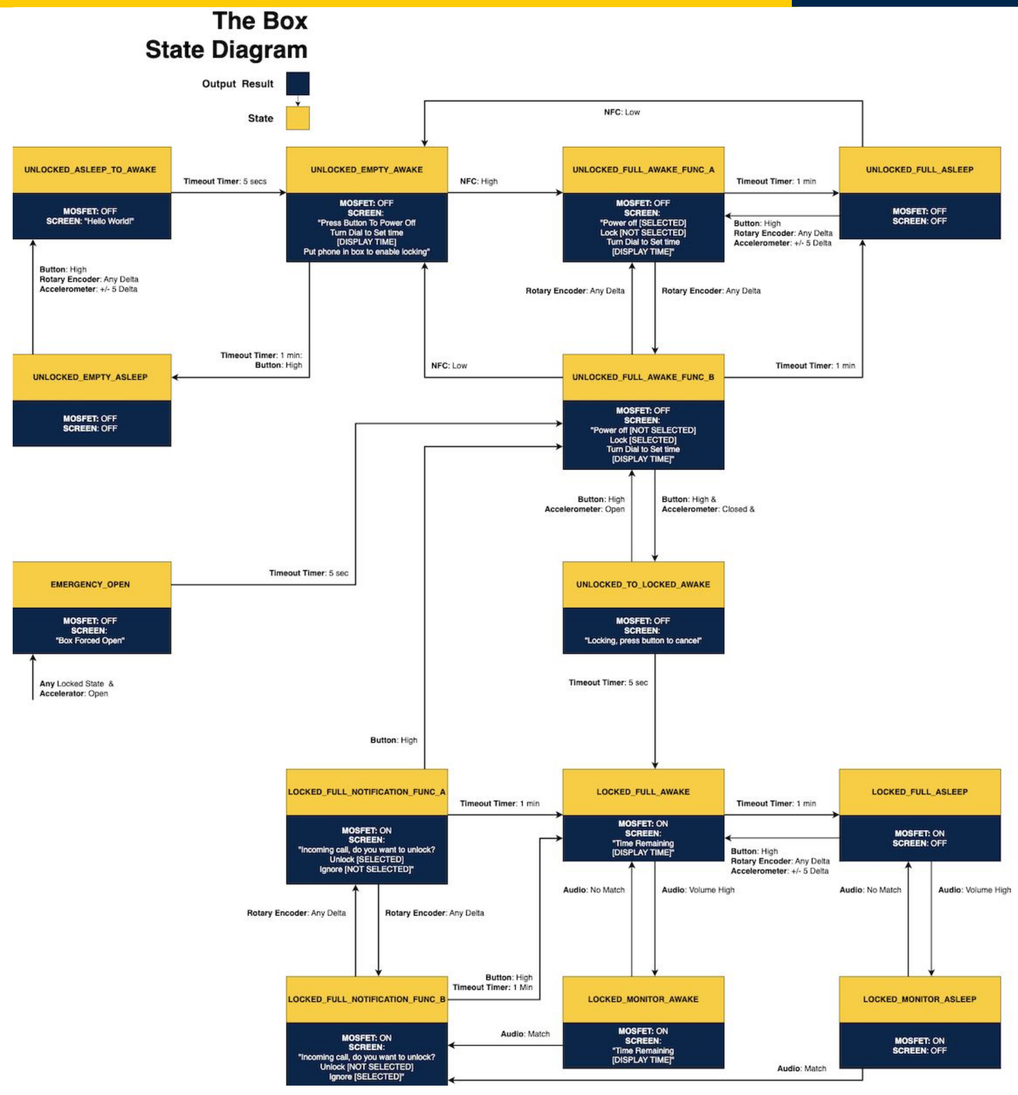
Once the time is set, a **predefined ringtone** going off will override the countdown and prompt the user to unlock.







The Box State Diagram



The Hardware

- **STM32 Nucleo** board as the core of the embedded system
- **Hardware interrupts** configured for both the microphone & switch
- **I2C lines** used to communicate with the **NFC** and **accelerometer**
- **Solenoid lock** driven by **MOSFET** and **voltage regulator**
- Timers used for both **LCD screen** and **rotary encoder** functionality
- System is **battery powered** with **wireless charging** passthrough for added convenience

The Software

Building an RTOS

- Prevents **time-consuming operations** running too frequently
- Only **polls devices** when in relevant states to reduce overhead
- All operations are handled as **events**, which trigger specific responses

State Machines

- **FSM** model used to control transition logic
- **Events & Interrupts** set flags that inform the FSM
- States were defined for **every unique output combination**, ensuring precise control logic