Keyless Smart Lock (Secured Illini)
ECE 445 Design Document - Spring 2025

Project #61
Andrew Ruiz, Sebastian Sovailescu, Bowen Cui

Professor: Arne Fliflet
TA: Sanjana Pingali

Contents:

1. Introduction

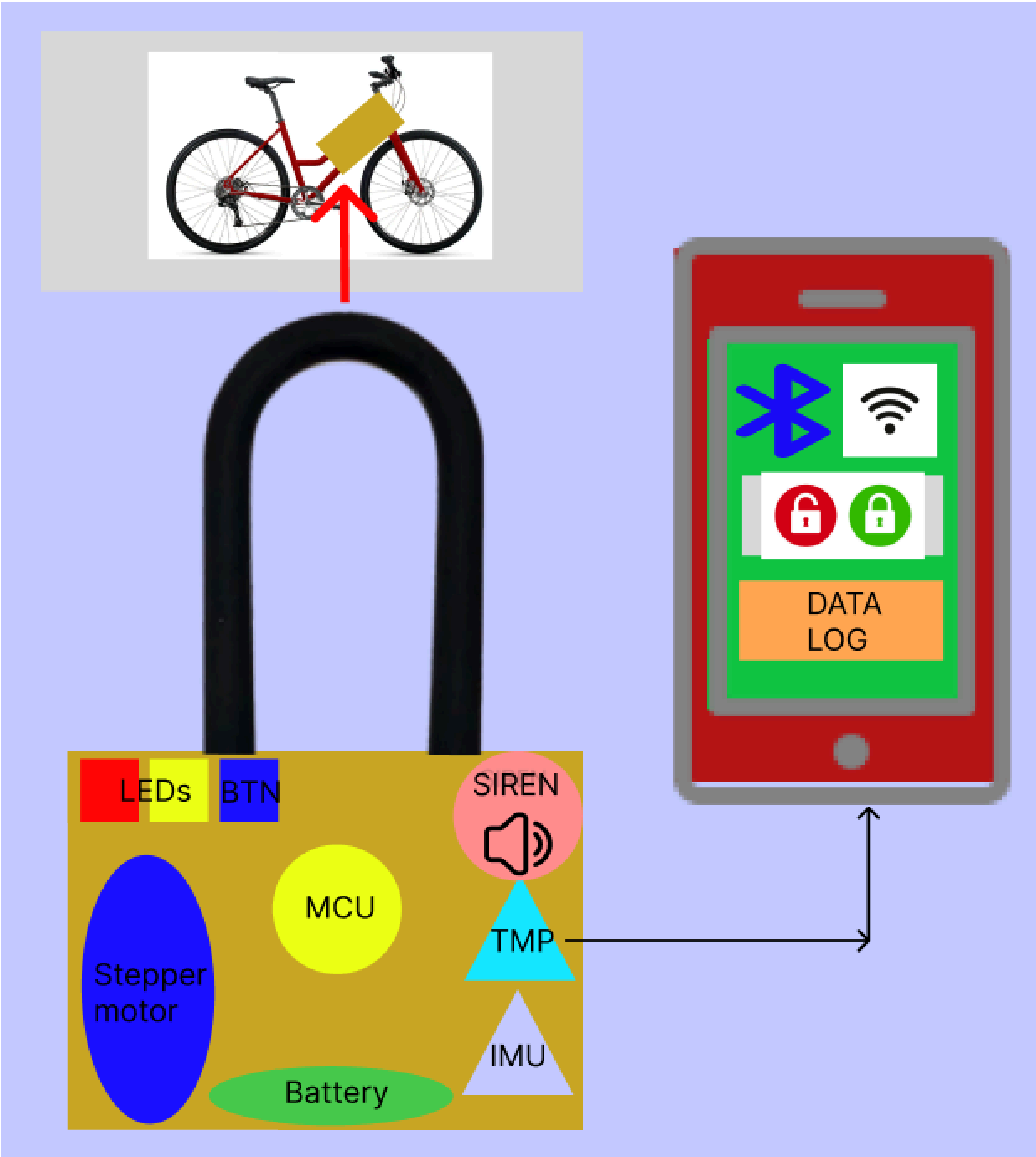
1.1. Problem

Bike theft remains a major issue in urban and suburban areas, with millions of bicycles stolen annually due to the shortcomings of conventional locks. Despite the use of Ulocks and chain locks, thieves easily bypass them using bolt cutters, angle grinders, and lock-picking tools. According to 529 Garage, over two million bikes are stolen each year in North America, discouraging cycling and undermining sustainable transportation efforts. Research by Sidebottom et al. (2009) highlights that even high-security locks can be compromised within minutes, exposing the need for more effective theft prevention measures. Additionally, improper locking techniques further contribute to the problem, leaving bicycles vulnerable. Addressing these security gaps is essential to protecting cyclists and promoting bicycle use as a reliable mode of transportation.

1.2. Solution

We propose a smart bike lock equipped with tracking, a keyless locking mechanism via Bluetooth, and an integrated siren that offers a comprehensive solution to the problem of bike theft. GPS/WiFi tracking ensures that stolen bikes can be quickly located and recovered, significantly increasing the chances of retrieval compared to traditional locks. The keyless locking mechanism eliminates vulnerabilities associated with physical keys or combinations, reducing the risk of lock picking or brute-force attacks. By using Bluetooth connectivity, cyclists can securely lock and unlock their bikes through a smartphone app, adding convenience while maintaining security. Additionally, a built-in siren serves as an active deterrent by emitting a loud alarm when unauthorized tampering is detected, drawing attention and discouraging thieves. This multi-layered security approach not only makes theft more difficult but also increases the likelihood of intervention before a bike is stolen. By integrating these advanced features we will be helping to reduce bike theft rates and promote cycling as a secure and viable mode of transportation.

1.3. Visual Aid

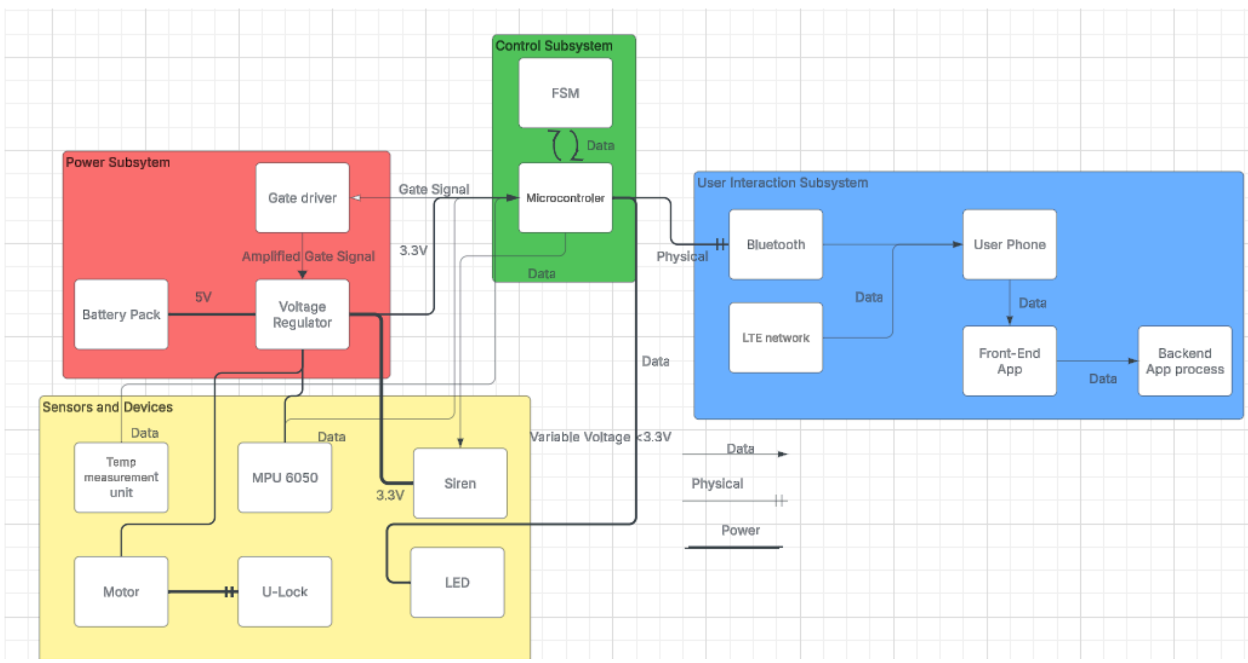


1.4. High Level Requirements

- Electronic locking system that can be manually overridden with master key (or override code) that can withstand over 1000 lbs of force.
- A 90dB siren will sound for 10 seconds when our anti-theft algorithm detects suspicious activity within a locked state. Theft attempts will be determined when excessive movement is detected which sensitivity will be experimented with.
- Can send and receive real-time alerts, commands, or temperature readings over WiFi and/or Bluetooth using a custom built web application. All communications will be safe.
- 3.3V Indicator LEDs to indicate the lock's current state as well as power level.

2. Design

2.1 Block Diagram



2.2 Physical Design

The smart bike lock will have a U-lock shape, combining durability and security while accommodating the necessary electronics. The main housing at the base will enclose the PCB, battery, and motorized locking mechanism, ensuring a compact yet functional design. A stepper motor will control the locking bolt, allowing for electronic and backup manual operation. The U-shaped shackle, made of steel, will resist cutting and prying. To withstand outdoor conditions, the lock will feature weather-resistant materials and an IP-rated enclosure. Designed for easy mounting and portability, the lock will balance security, usability, and smart connectivity.

2.3. Subsystems

2.3.1. Power

Description

This subsystem is very small as it just consists of two components, but it is one of the most critical components of the entire design. This subsystem consists of a 5V 8AH rechargeable battery pack and a LM1117 3.3V voltage regulator that will step down the 5V to 3.3V and provide power to most components in the system.

Interfaces

-5V rail directly connected from Battery pack

-3.3V rail from voltage regulator

-Recharge port on battery pack

Requirements	Verification
Battery life can last up to 7 days	We can test voltage differences using a multimeter.

2.3.2. Controls

i. Description

The Control subsystem, centered around the ESP32, is the “brain” of the smart bike lock. It reads sensor data(accelerometer/gyroscope, temperature), processes user inputs (button presses or wireless commands), and sends outputs to the actuator (stepper motor), alarm (piezo buzzer), and LED lights. By monitoring for theft attempts, executing lock/unlock operations, and managing alerts and states, the MCU helps accomplish multiple high-level requirements. locking/unlocking within 100ms; withstanding at least 1000 lbs of force; distinguishing between noise and tampering; sounding alarm for 5-10 seconds; indicating state correctly using LEDs.

ii. Interfaces

Motor driver <-> ESP32 : GPIO

Temperature <-> ESP32 : GPIO

Button <->ESP32 : GPIO

LEDs <->ESP32 : PWM GPIO

iii. R&V

Requirements	Verification
Lock/Unlock	Metric : <100ms from press/command to response Verification : timestamp at time of press and unlocked status reached

2.3.3. Anti-Theft

i. Description

The Anti-Theft system incorporates both detection and alarm. Controlled by the ESP32, they first use the Gyroscope and Accelerometer to detect the current position of the lock. If the lock is disturbed too significantly, the alarm would be triggered. The gyroscope should take a snapshot of the current lock position every time the lock goes in to the locked and armed state, regardless of whether it was from the unlocked or the alarmed state. The accelerometer is much less complicated with it simply raising the alarm when it detects a high enough acceleration.

ii. Interfaces

Gyroscope <-> ESP32 : GPIO I2C
Accelerometer <-> ESP32 : GPIO I2C
Siren <-> ESP32 : GPIO PWM

Requirements	Verification
Alarm	Metric : raises the alarm if the bike is disturbed if the bike is turned for more then 45 degrees or moved for more than 1 meter. Verification : Alarm raised if conditions are met.

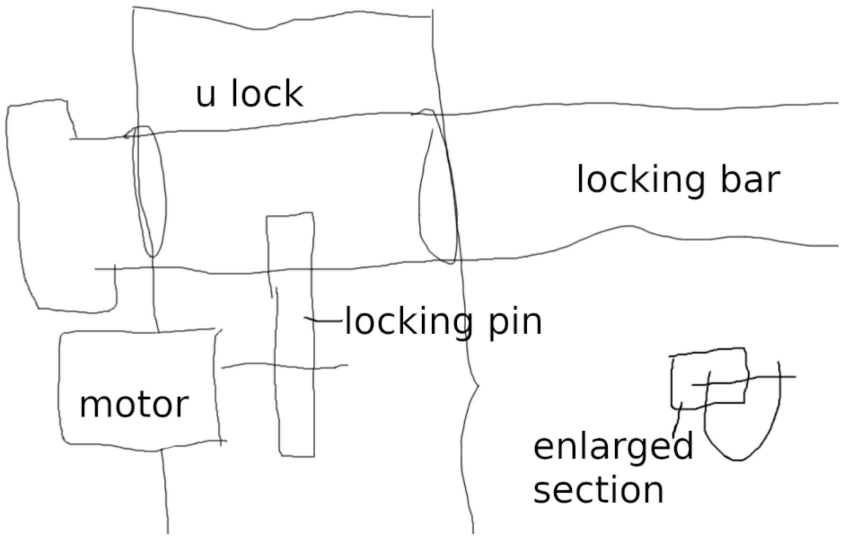
2.3.4. Locking

i. Description

The locking system is designed to prevent the lock from being opened while it is not in the unlocked state. We accomplish that by using a U lock based design that has 3 main components, the U bar, the locking bar and the locking pin. When unlocked, the locking bar should be able to freely slide within the U bar to either prevent or allow the lock to be removed. The Locking bar should be held captive within the U bar so they stay as one piece. The locking pin is within the U bar itself and when the locking bar is in the closed position, there would be a hole that lines up with another hole in the U bar itself. The motor then pushes the locking pin through the holes to prevent the locking bar from being moved when it receives the locking command from the controller.

Ii.Interfaces

Motor <-> motor controller <-> ESP32



iii.

Requirements	Verification
Withstand 1000 lbs	Hit the lock with a hammer. Drop it from a building.
Be waterproof	Create a waterproof enclosure

2.3.5. User Interface

i. Description: The UI Subsystem is there to facilitate communication between the user and the smart bike lock. The web interface would allow the user to lock or unlock the smart bike lock using either WiFi or Bluetooth. It would also collect and display data logs of alerts that include timestamps and status.

ii. Interfaces: Arduino IDE has libraries for Bluetooth Low Energy and WiFi.
Lock <-> ESP32 : ThingSpeak or local server (WiFi)
Lock <-> ESP32 : Bluetooth
Tmp <-> ESP32 : GPIO

iii.

Requirements	Verification
Website	Once a button is pressed, the lock follows the command within 100ms.
Data Logs	The website updates its logs every time a connection is established.

Requirements	Verification
Website	Once a button is pressed, the lock follows the command within 100ms.
WiFi	Data is sent to a cloud database. We can unlock the bike over WiFi using a HTTP request.
BLE	We can establish a connection, and show the status on the backend. We can unlock/lock the bike using the BLE connection

2.4 Tolerance Analysis

2.7.1 Battery life

assuming we use a 12V 5-10 Ah battery, we calculate the battery life using the formula : Battery life = Battery Capacity / Total Current Draw. Based on the ESP32- S3-WROOM datasheet, we made the following assumptions about each subsystem's average current consumption Component Average Current MCU 25mA BLE/WiFi 20mA Accelerometer/Gyroscope 3mA Siren 0.1mA Stepper Motor 0.1mA LED 0.001mA Total Current Draw 48.2001mA TABLE II COMPONENT AVERAGE POWER DRAW Total battery time = [5000 [mAh]/ 48.2001[mAh] = 103.734 [h] /24 = 4.32 days , 10000/ 48.2001 = 207.468 = 8.64]. In order to reach a battery life of one week we would likely need a 10000mAh cell or multiple smaller cells with a total capacity over 10000mAh

2.7.2 Lock Force Tolerance

With the locking mechanism, we have a quarter-inch diameter, 3 inch long steel bar that is held by two supports on the top when force is applied upwards. The U-bar itself is focused on a 1/4 inch section at the center of the bar so the tolerance is how much force can be applied across this section. Since the force is applied at the center of the quarter-inch diameter (0.25 inches) steel pipe, with support at both ends, we can model this as a uniformly distributed load (UDL) over a small length. The primary failure mode to consider is bending stress. For a simply supported beam with a point load at the center, the maximum bending moment Mmax is given by: Mmax = F(L − a)/4 with “F” being applied force and ”L” being the effective length of the beam (span between supports) which ends up being about an inch and “a” being the 0.25in length of the U-bar. The maximum bending stress in a circular cross-section is given by:MmaxC/I with “C” being the outer radius 0.25/ 2 = 0.125in and “I” being equal toI = (pid4)/64 The lock will fail when the maximum stress exceeds the yield strength of the steel which is about 53,700 psi. Given all of this we calculated that the maximum force the lock can withstand before bending is approximately 1,757.3 lbs. If the applied force exceeds this value, the lock will begin to deform permanently, reducing its effectiveness. We found that this value was more than acceptable because it would be very hard to manually apply this much force without industrial grade tools

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

Given as this project is for a class the true cost of labor will be $0, but we wanted to calculate how much a project like this would cost at market rate. Assuming a team of three individuals, a hardware engineer, a software engineer, and a product designer working on the project, we can estimate labor costs based on typical hourly wages.

Hardware Engineer: Responsible for circuit design, PCB layout, and integration of electronic components such as Bluetooth modules and locking mechanisms. Estimated hourly rate: $50–$70.

Software Engineer: Develops firmware for microcontrollers, mobile app connectivity, and security features like encryption. Estimated hourly rate: $60–$80.

Product Designer: Designs the physical enclosure, ensuring durability, weather resistance, and usability. Estimated hourly rate: $45–$65.

If each team member works 40 hours per week for 12 weeks, the total labor cost can be estimated as follows:

Hardware Engineer: $50 × 40 × 12 = $24,000 (minimum estimate)
Software Engineer: $60 × 40 × 12 = $28,800
Product Designer: $45 × 40 × 12 = $21,600
Total Estimated Labor Cost:
At minimum rates, the total labor cost for 12 weeks would be $74,400, while at higher rates, it could exceed $100,000. Additional costs may arise from extended development time, testing, and unforeseen challenges. Clearly it is beneficial to be designing this project as a passion project for this class as labor can be the biggest cost to the design.

3.1.2 Parts

Description	Manufacturer	Part #	Quantity	Cost
Unipolar stepper motor + Motor driver	KOOKYE	28BYJ-48 + ULN2003	2	$8.99
Microcontroller	Espressif	ESP32-S3 module	3	$13.80
Piezo buzzer sensor	Adafruit	SBZ-204	1	$1.62
6-axis Accelerometer Gyroscope Sensor	HiLetgo	MPU-6050	3	$10.99
5V 8Ah Battery Pack	Adafruit	1566	1	$29.99
Plastic Electronic Project Box	WeiMeet	IP65	1	$7.99
LEDs	Digikey	QBL7IB60D	2	$0.38 x 2 = $0.76
Push-button	Adafruit	1683	1	$3.33
Temperature Sensor	Mouser	LM335AH	1	$0.68
Micro USB Connector	Digikey	WM11262TR-ND	1	$1.12
Voltage Converter	Digikey	LM1117MP-3.3	1	$1.38

Total				$79.27

3.1.3 Total costs
Labor ($74,400) + Parts ($78.15) + Machine shop hours ($100 x 15) = 75,978 ~ **$76,000**
3.2 Schedule

	3/10 - 3/24	3/24 - 4/7	4/7 - 4/21	4/21 - 5/5
Sebastian	Demo : MCU + IMU + Servo.	Website with backend (WiFi) Anti-Theft	Bluetooth	Final Report Final Demo
Andrew	PCB design improvement and testing	PCB design cont. And begin to implement with full circuit	Implement full design and prepare demo	Final Report
Bowen	Anti-Theft algorithm Locking mechanism	3D printed case Locking mechanism improved	Prepare demo	Final report

4. Ethics and safety

Developing a smart bike lock with tracking, Bluetooth locking, and an alarm system presents several ethical and safety considerations, particularly in line with the IEEE Code of Ethics and the ACM Code of Ethics. Privacy is a major concern, as location tracking must be handled responsibly to prevent misuse. To ensure user data security, we will require explicit consent for location tracking. Additionally, reliability is crucial, as a malfunction could leave a user stranded. Following ethical guidelines to ”avoid harm,” we will integrate redundant unlocking mechanisms, such as backup PIN entry or an emergency override. In addition to ethical considerations, our design must comply with safety and regulatory standards. Since our smart lock utilizes Bluetooth and GPS, it must meet FCC Part 15 regulations for radio frequency emissions and comply with UL 437 security standards to ensure resistance to physical attacks like cutting or drilling. Additionally, we must consider state laws regarding electronic tracking devices, ensuring that location data remains private and is accessible only to the owner. Potential safety concerns, such as false alarm activations will be addressed by implementing adaptive sensitivity settings. By adhering to these ethical and safety standards, we can develop a secure, reliable, and compliant smart lock that effectively reduces bike theft while minimizing risks to users.

5. References

ACM. *ACM Code of Ethics and Professional Conduct*. Association for Computing Machinery, 2018, https://www.acm.org/code-of-ethics.

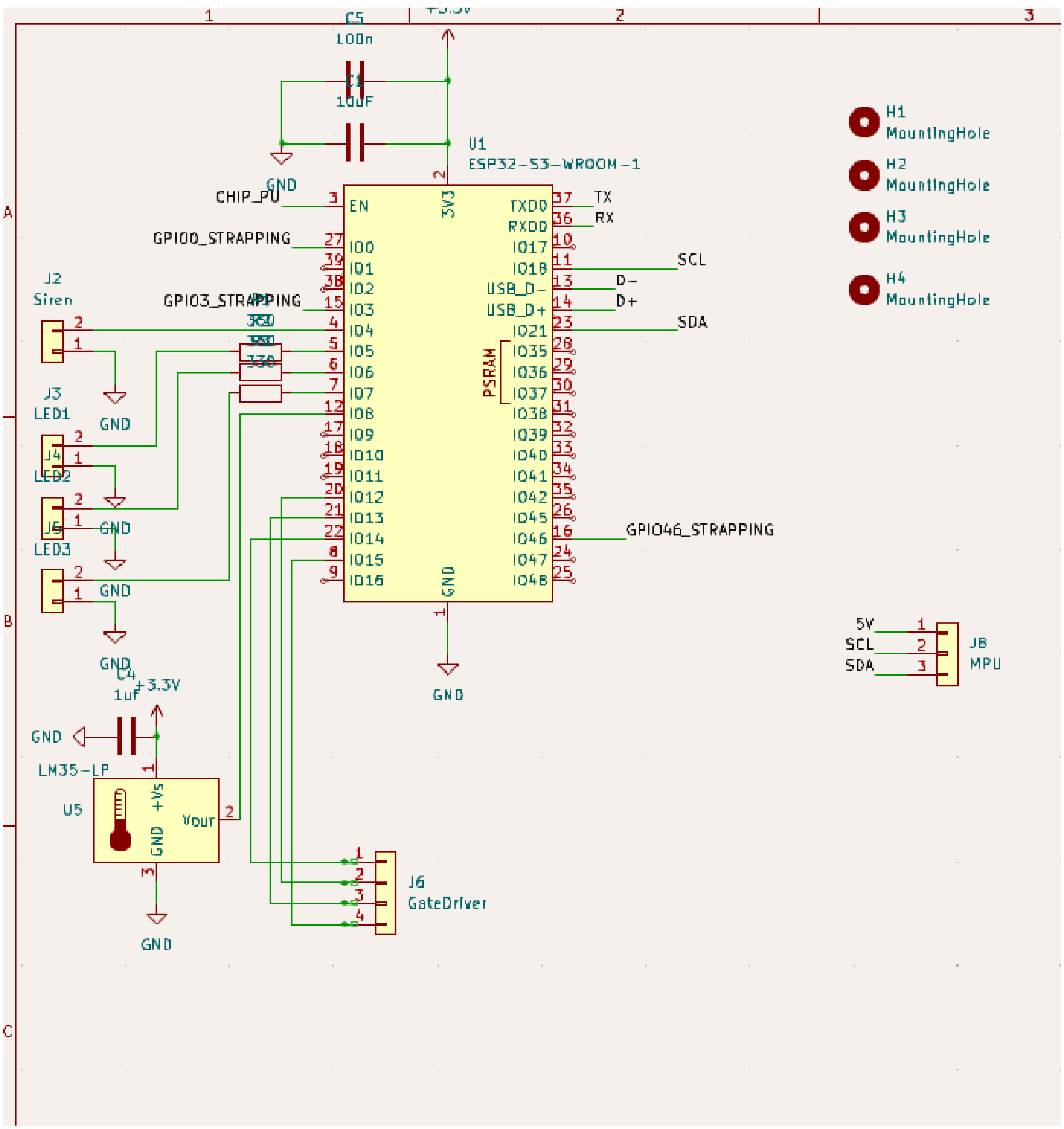
FCC. *Title 47 CFR Part 15 – Radio Frequency Devices*. Federal Communications Commission, https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15.

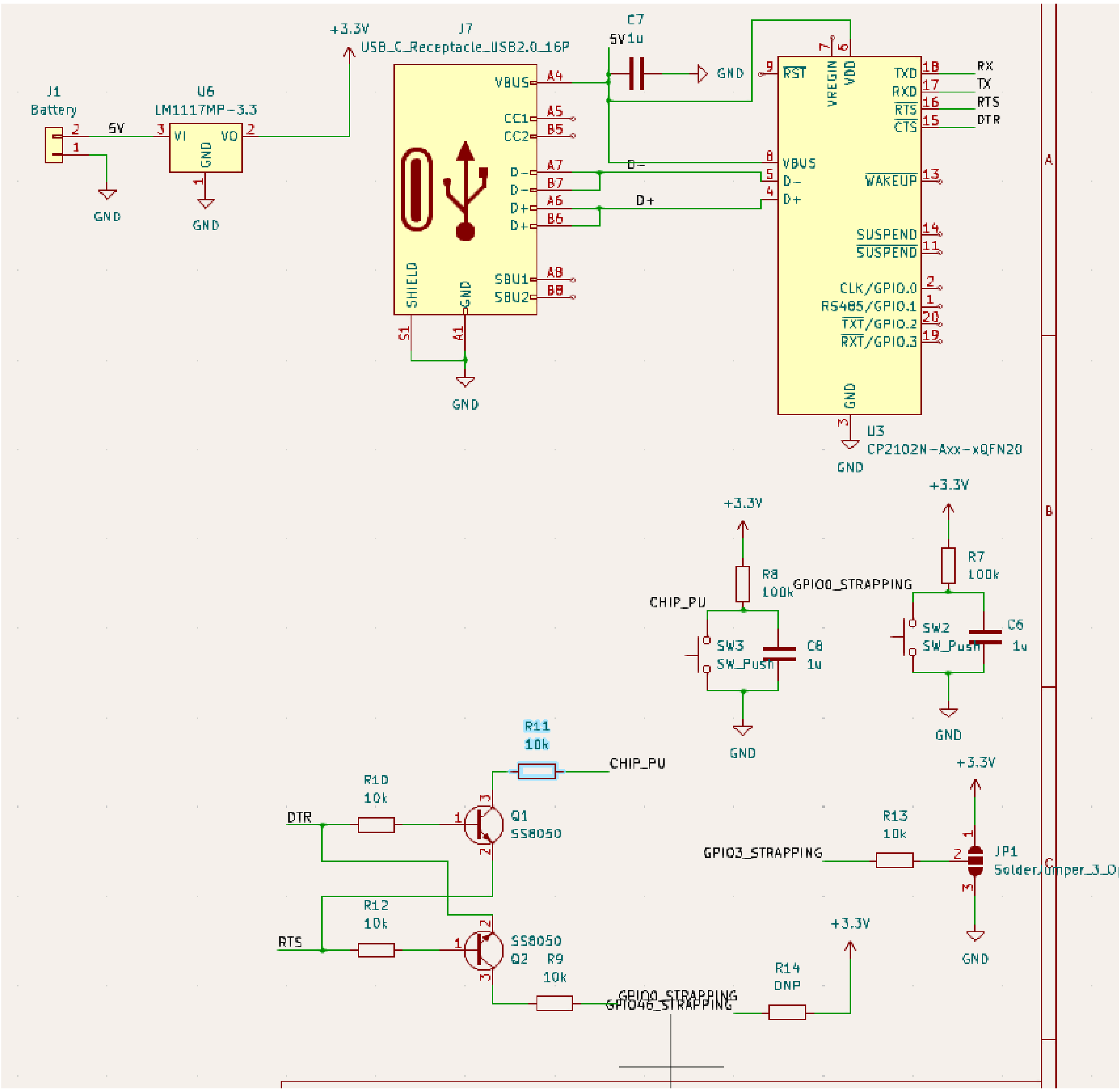
IEEE. *IEEE Code of Ethics*. Institute of Electrical and Electronics Engineers, https://www.ieee.org/about/corporate/governance/p7-8.html.

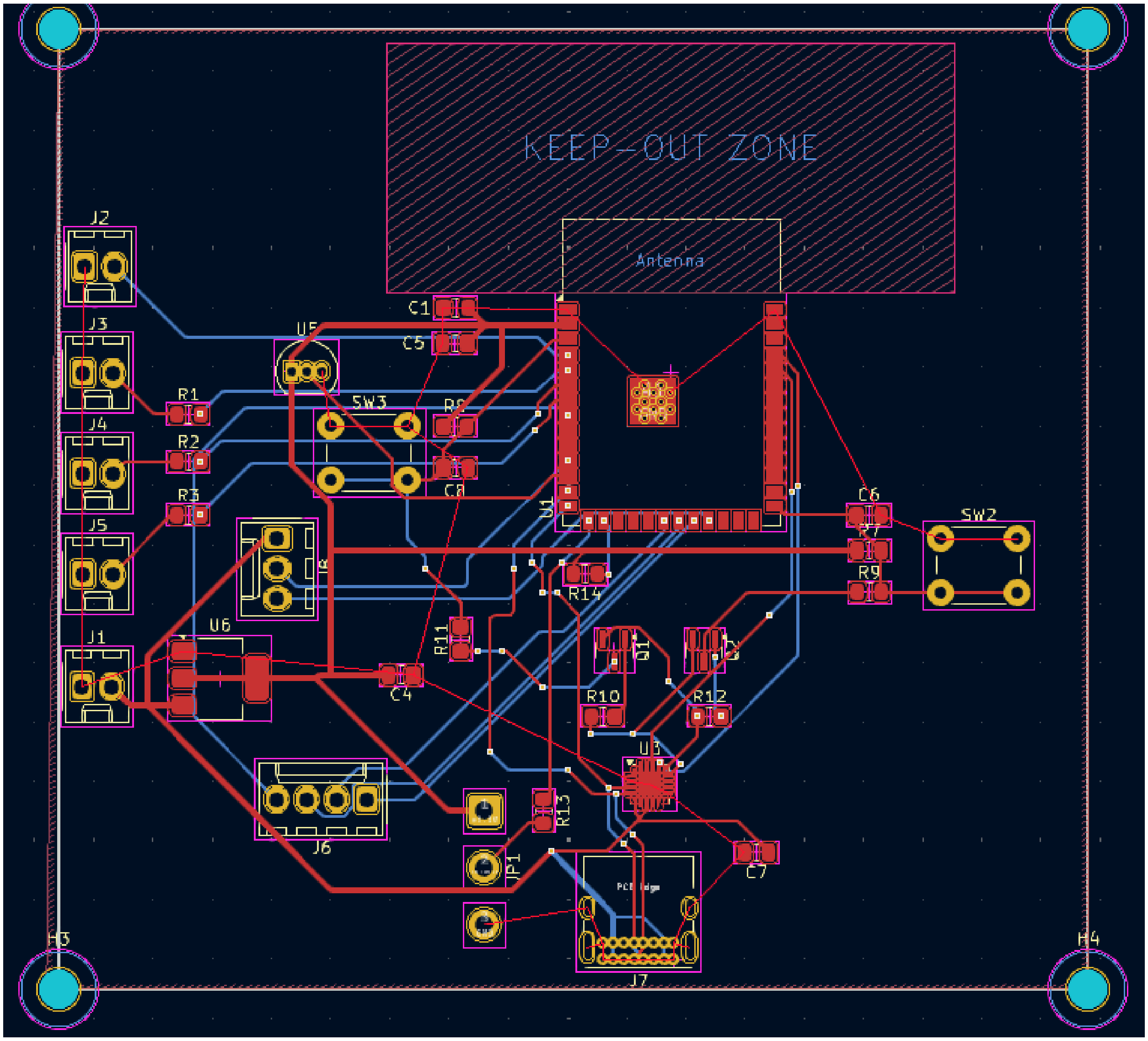
UL. *UL 437 – Standard for Key Locks*. Underwriters Laboratories, https://www.ul.com.

U.S. State Laws on Electronic Tracking Devices. *Legal Considerations for GPS Tracking*, National Conference of State Legislatures, https://www.ncsl.org.

Sebitian. *ECE445*. GitHub, last updated **6 Mar 2025**, https://github.com/Sebitian/ECE445.


C5
100n
+3.3V
C1
10uF
U1
ESP32-S3-WROOM-1
GND
CHIP_PU
GPIO0_STRAPPING
GPIO3_STRAPPING
GPIO46_STRAPPING
TX
RX
SCL
SDA
D-
D+
EN
3V3
TXD0
RXD0
PSRAM
J2
Siren
J3
LED1
J4
LED2
J5
LED3
330
C4
1uF
LM35-LP
U5
+VS
VOUT
J6
GateDriver
H1
MountingHole
H2
MountingHole
H3
MountingHole
H4
MountingHole
5V
J8
MPU


J1
Battery
5V
U6
LM1117MP-3.3
VI
VO
GND
+3.3V
J7
USB_C_Receptacle_USB2.0_16P
VBUS
CC1
CC2
D-
D+
SBU1
SBU2
SHIELD
C7
5V 1u
U3
CP2102N-Axx-xQFN20
RST
VREGIN
VDD
TXD
RXD
RTS
CTS
WAKEUP
SUSPEND
CLK/GPIO.0
RS485/GPIO.1
TXT/GPIO.2
RXT/GPIO.3
RX
TX
RTS
DTR
GND
R8
100k
CHIP_PU
SW3
SW_Push
C8
1u
GPIO0_STRAPPING
R7
100k
SW2
SW_Push
C6
1u
R11
10k
R10
10k
DTR
Q1
SS8050
R12
10k
RTS
SS8050
Q2
R9
10k
GPIO0_STRAPPING
GPIO46_STRAPPING
R14
DNP
GPIO3_STRAPPING
R13
10k
JP1
SolderJumper_3_Op



KEEP-OUT ZONE
Antenna
J2
J3
J4
J5
J1
U5
SW3
R1
R2
R3
R8
C1
C5
C8
U1
C6
R7
R9
SW2
R14
R11
U6
C4
Q1
R10
R12
U3
J6
R13
TP1
C7
PCB Edge
J7
H3
H4