KeepSafe
Final Product Review

Jaret Williams, Sean Letavish, Nathan Pen

Introduction

- KeepSafe is a smart self-locking package box that will prevent package theft
- KeepSafe will provide consistent package security and provide the user with real time package delivery updates

KeepSafe Team

Jaret Williams
Electrical Engineering

Sean Letavish
Electrical Engineering

Nathan Pen
Electrical Engineering

Needs and Motivation

- Smart home technology integration and package theft are both continue to rise every year
  - Smart home technology is a growing trend and is currently a $80 billion market (Mordor Intelligence)
  - $6 billion worth of package stolen in 2020 (Swift Lane)
- Covid-19 pandemic has also contributed to an increase in online shopping
- KeepSafe is unlike any product currently on the market, combining smart features and easy to use operation for the user and delivery driver
- Will impact the consumer, seller, and environment

Package theft gives rise to delivery alternatives | azdailysun.com

The Story

2013

NEW VENTURE COMPETITION

2019

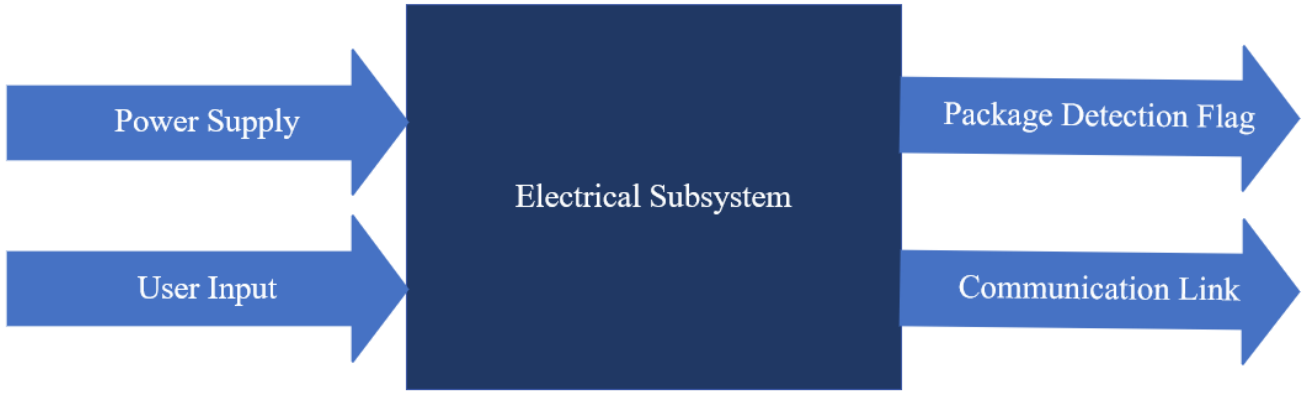
2022

Overall Requirements and Specifications

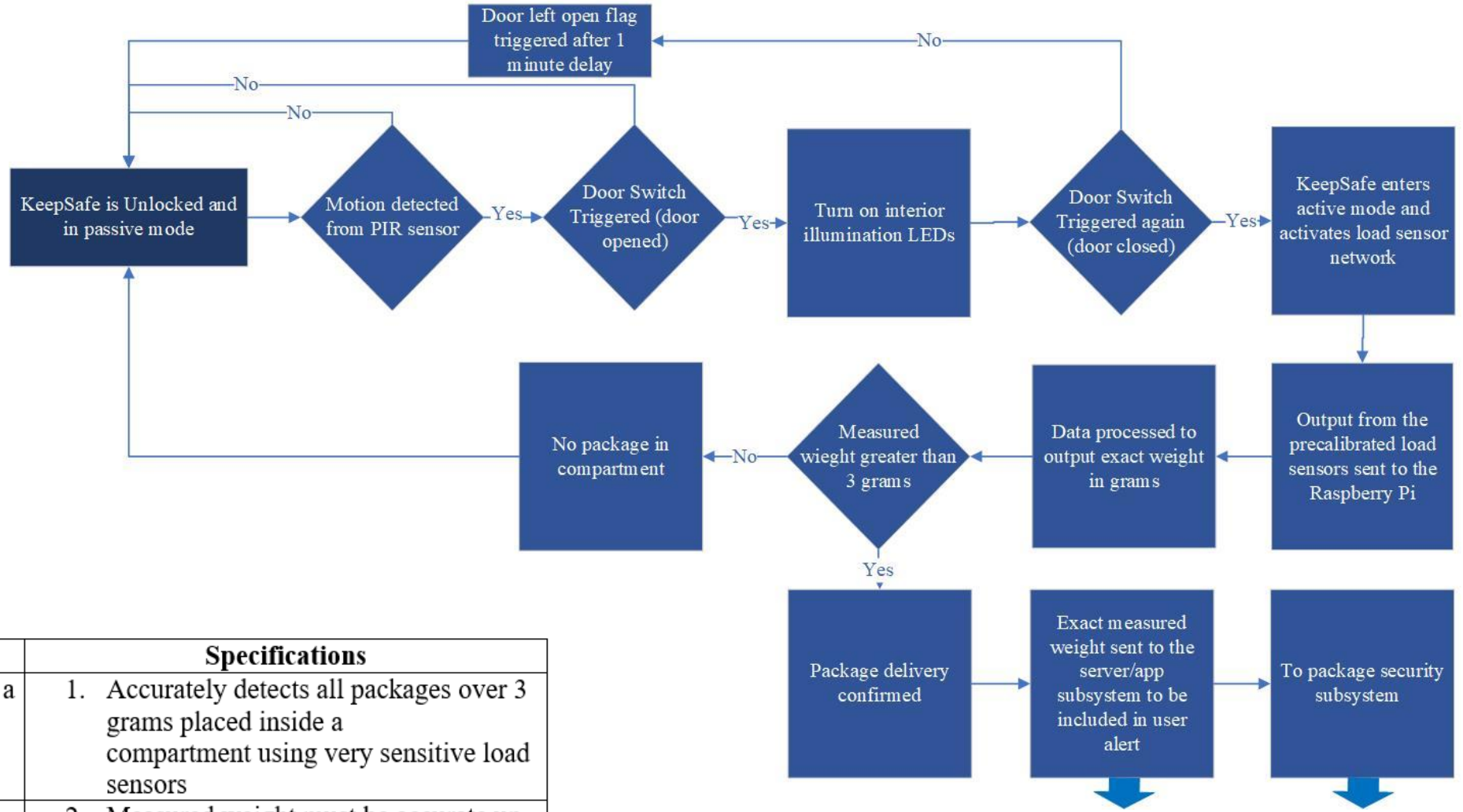
Requirements	Specifications
(1) Package must be detected	(1) Any sized package must be consistently recognized by the system
(2) Package must be secured	(2a) Automatically locks the packages inside the box when delivered. (2b) Lock must engage every time with no error.
(3) User must be notified	(3) User must be alerted every time after a package is detected and secured
(4) Overall system must be protected from theft	(4a) KeepSafe will bolted to the ground when possible (4b) Provide the user will GPS coordinates of the box when it is being moved
(5) User must be able to control KeepSafe through an application	(5) User will be able to perform the following functions through the app: - Unlock KeepSafe - Check Package Status - Check GPS Location - Check Camera

Electrical Subsystem

Requirements	Specifications
1. The device must illuminate inside of box when opened	1. Line the inside of the box with bright low power LEDs to make nighttime use easier
2. The device must be equipped with an electronic door switch	2. Device knows the state of the door to then check for packages
3. The device must be equipped with an electronic keypad for manual unlock and package retrieval	3. Packages can be retrieved by anyone in the household if they do not have access to the app

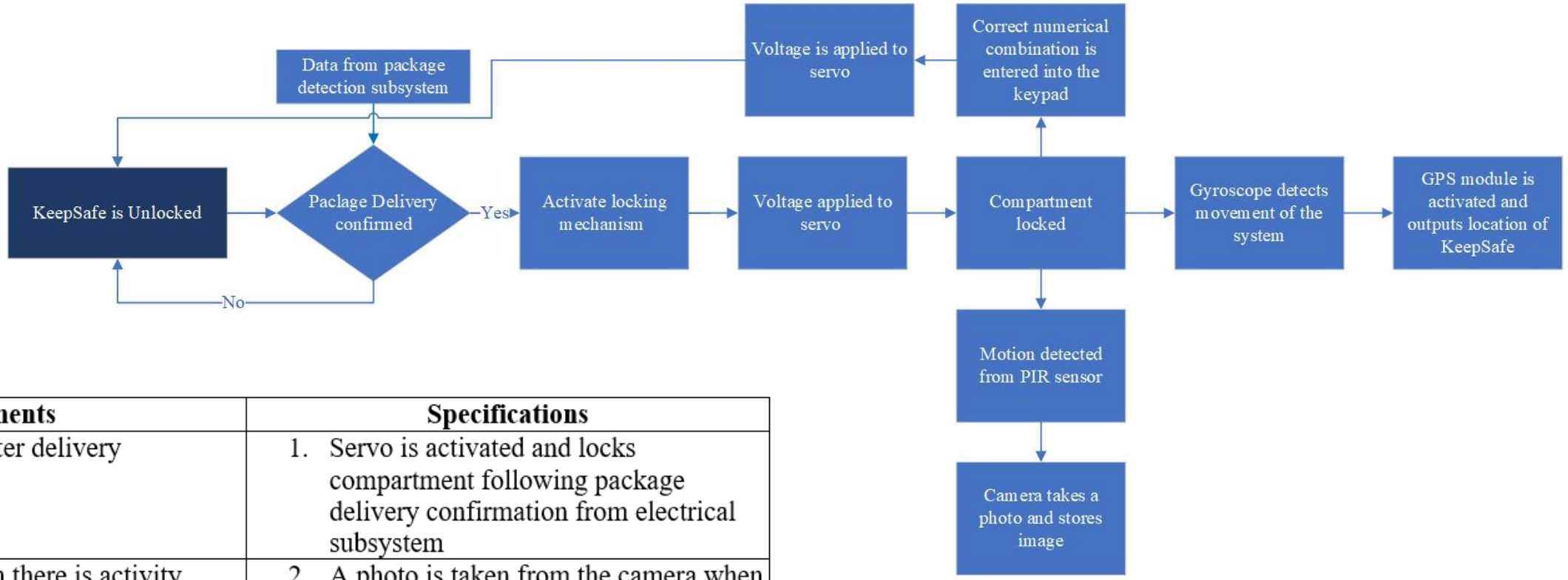


Electrical Subsystem: Package Detection



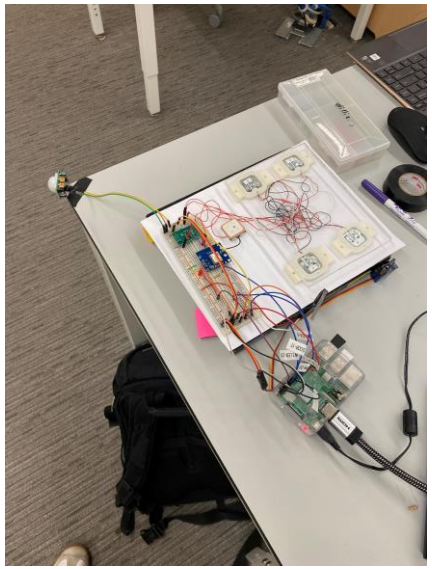
Requirements	Specifications
1. The device must be able to detect when a package has arrived	1. Accurately detects all packages over 3 grams placed inside a compartment using very sensitive load sensors
2. Accurately inform the user of the packages weight	2. Measured weight must be accurate up to 5% error

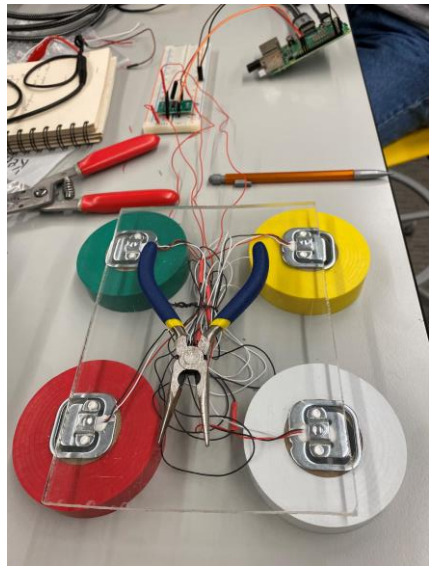
Electrical Subsystem: Package Security

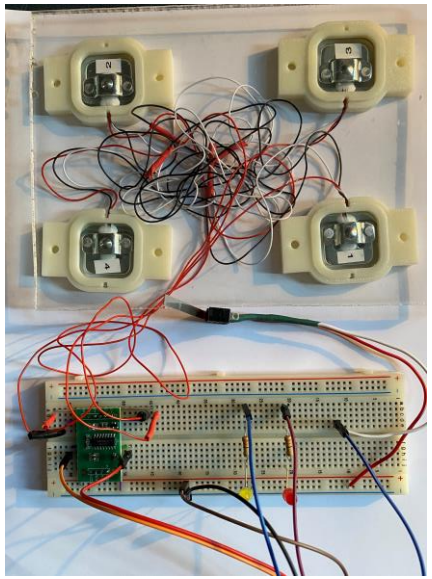


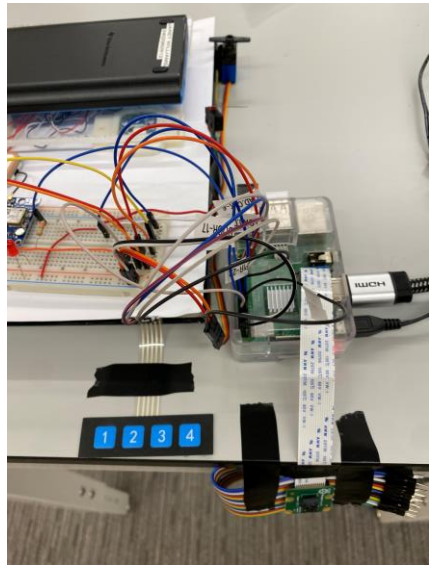
Requirements	Specifications
1. Package secured after delivery	1. Servo is activated and locks compartment following package delivery confirmation from electrical subsystem
2. Takes a photo when there is activity around the box	2. A photo is taken from the camera when motion is detected from the PIR sensor and stores the image
3. Track the location of the system if it is stolen	3. GPS module is activated by the gyroscope when the entire system is being physically moved (i.e., potential theft)

Electrical Subsystem



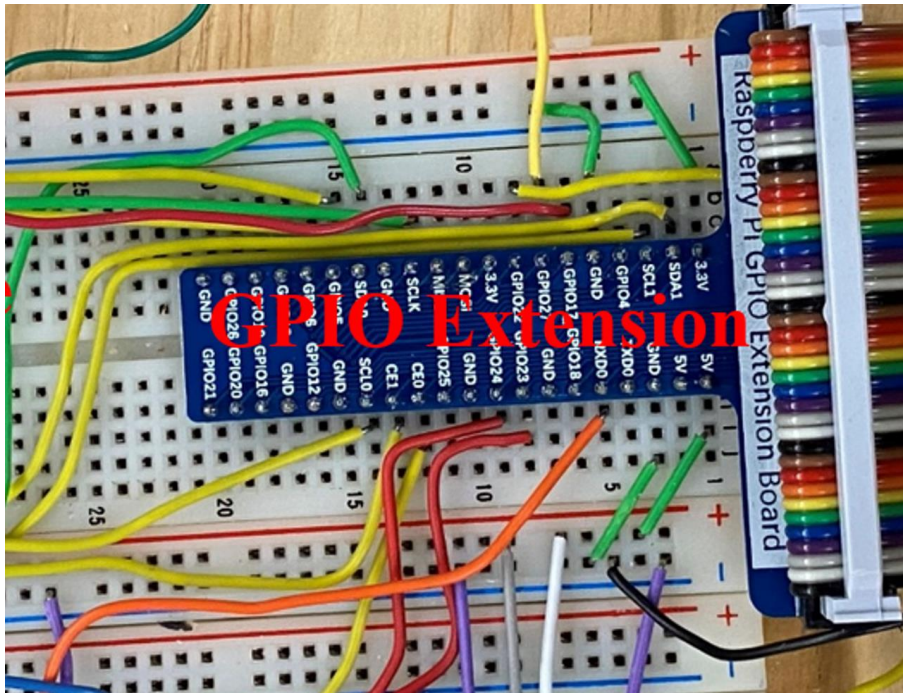






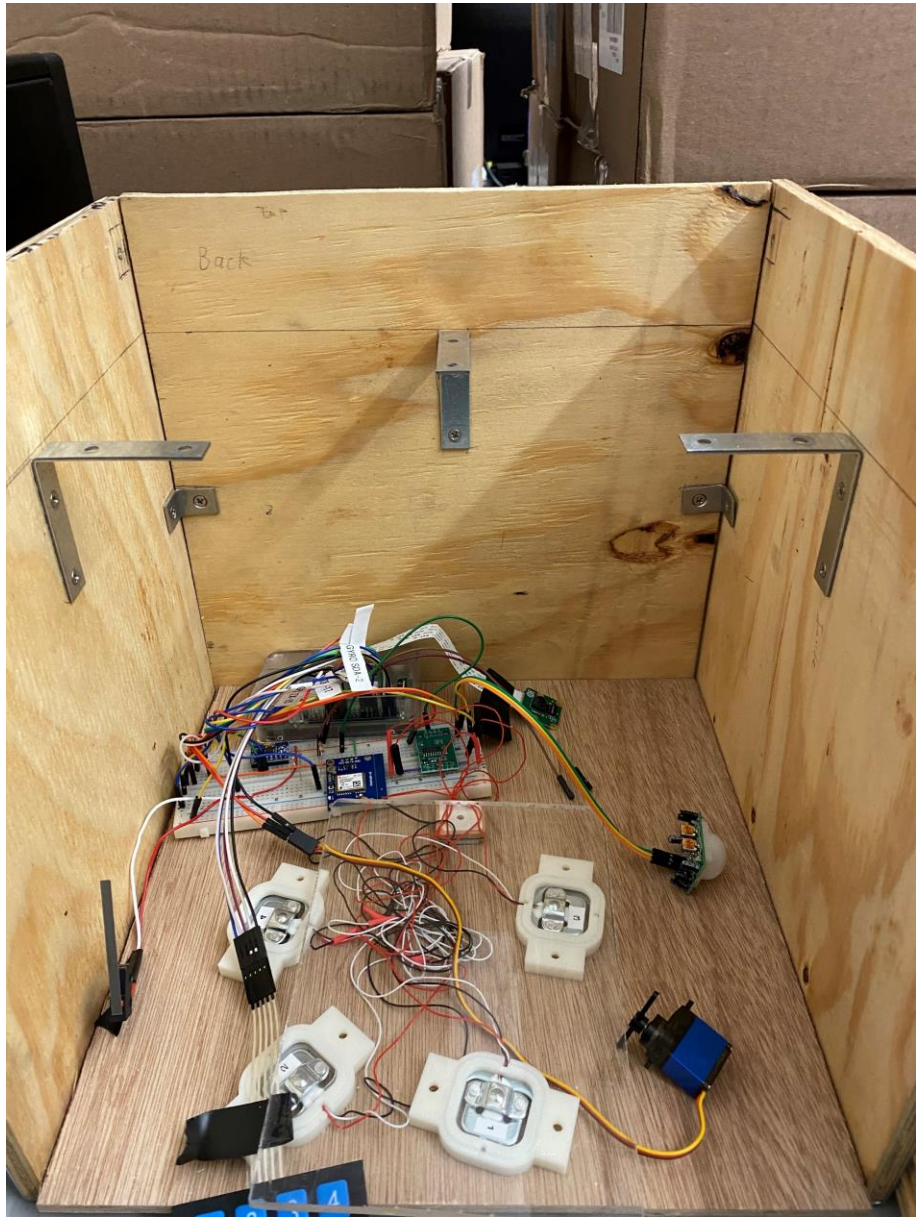
Electrical Subsystem

Part	Number	Pin #	Pin	Total
Door switch	1	1	GPIO 17	1
Interior LED	2	1	GPIO 23, 24	2
Keypad	1	5	GPIO 11, 8, 7, 1, 0	5
Servo	1	1	GPIO 27	1
Load sensor network	1	2	GPIO 5, 6	2
PIR Sensor	1	1	GPIO 22	1
GPS Module	1	1	GPIO 15 Rx (connect gps tx only)	1
Gyroscope	1	2	GPIO 2, 3 (SDA, SCL)	2
				15

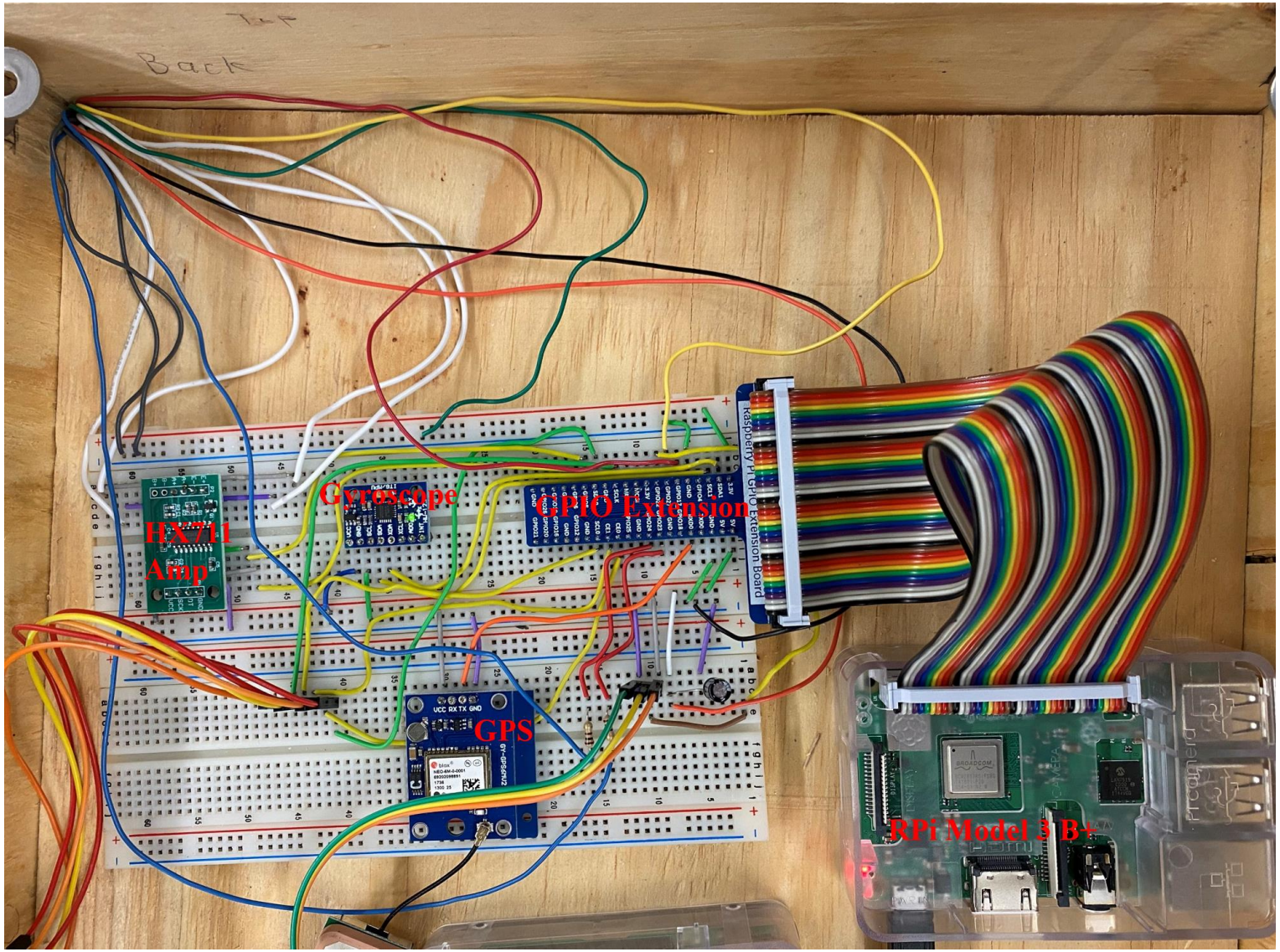


1	3.3V	5V	2
3	GPIO 2	5V	4
5	GPIO 3	GND	6
7	GPIO 4	GPIO 14	8 Tx
9	GND	GPIO 15	10 Rx
11	GPIO 17	GPIO 18	12
13	GPIO 27	GND	14
15	GPIO 22	GPIO 23	16
17	3.3V	GPIO 24	18
19	GPIO 10	GND	20
21	GPIO 9	GPIO 25	22
23	GPIO 11	GPIO 8	24
25	GND	GPIO 7	26
27	GPIO 0	GPIO 1	28
29	GPIO 5	GND	30
31	GPIO 6	GPIO 12	32
33	GPIO 13	GND	34
35	GPIO 19	GPIO 16	36
37	GPIO 26	GPIO 20	38
39	GND	GPIO 21	40

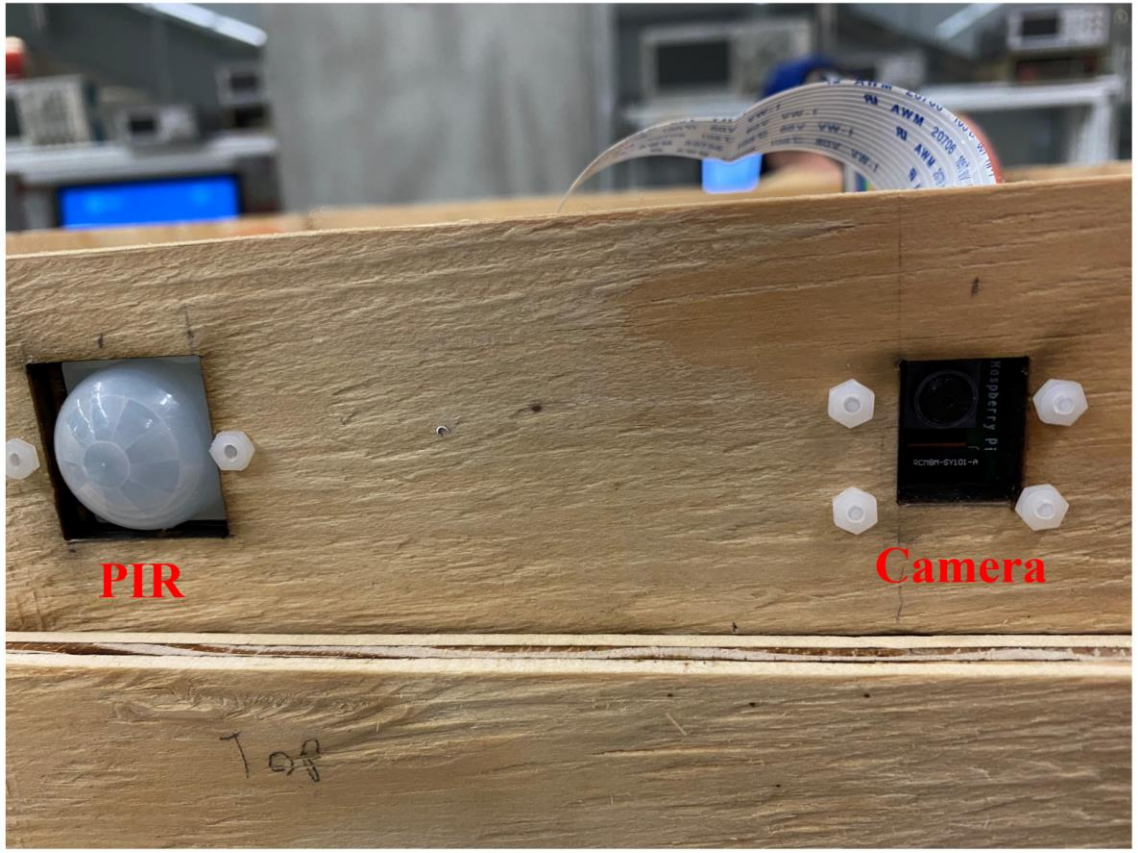
Electrical Subsystem

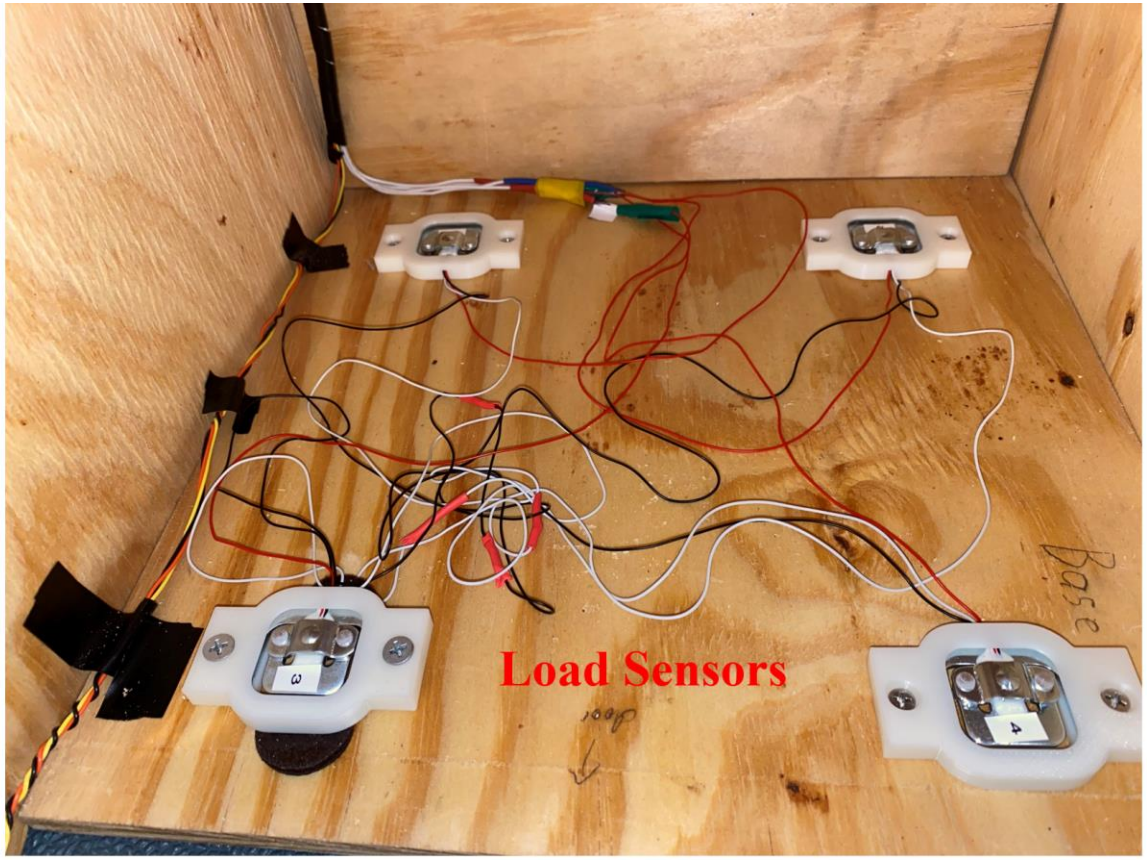


Electrical Subsystem

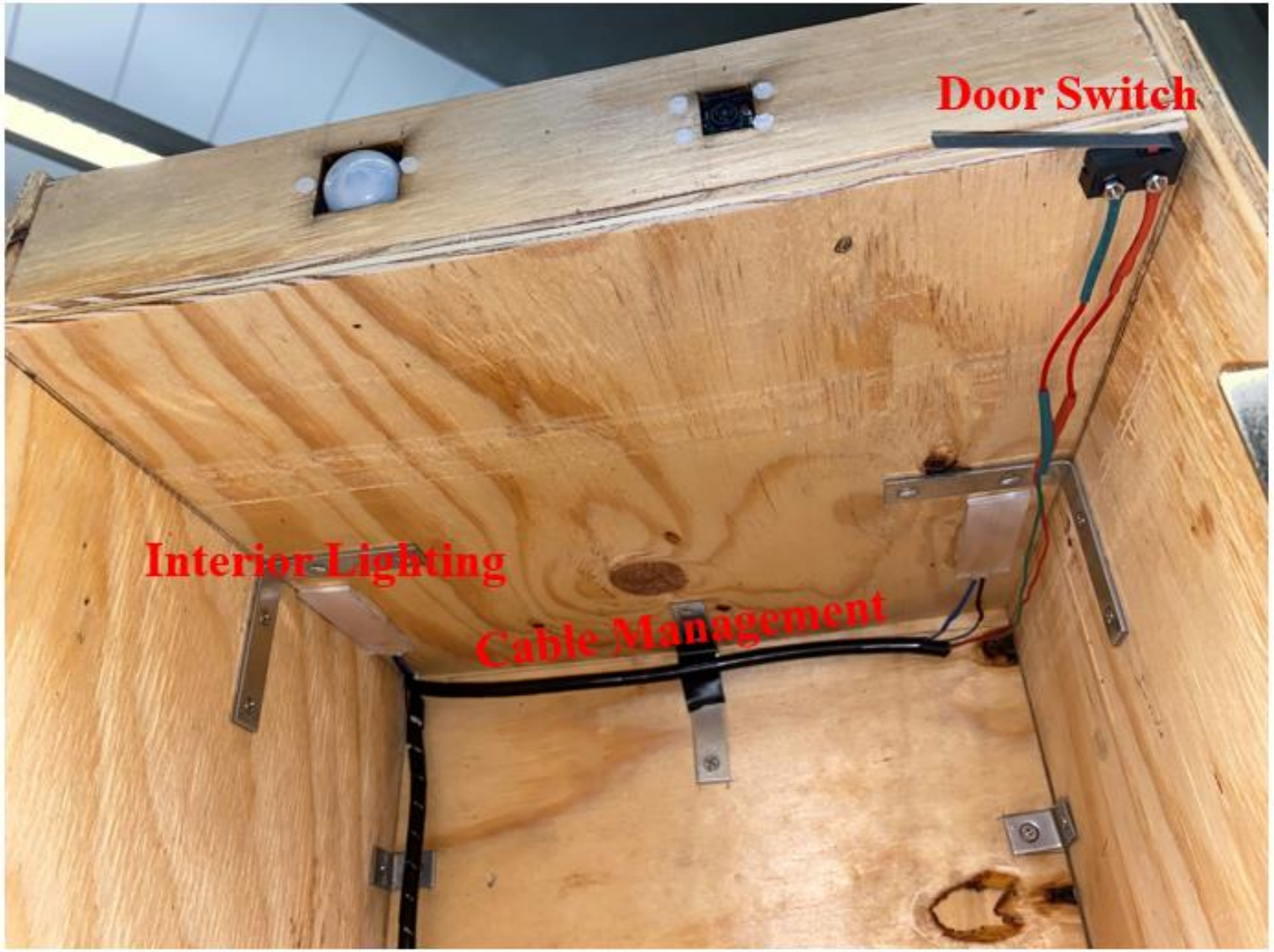


Electrical Subsystem





Electrical Subsystem

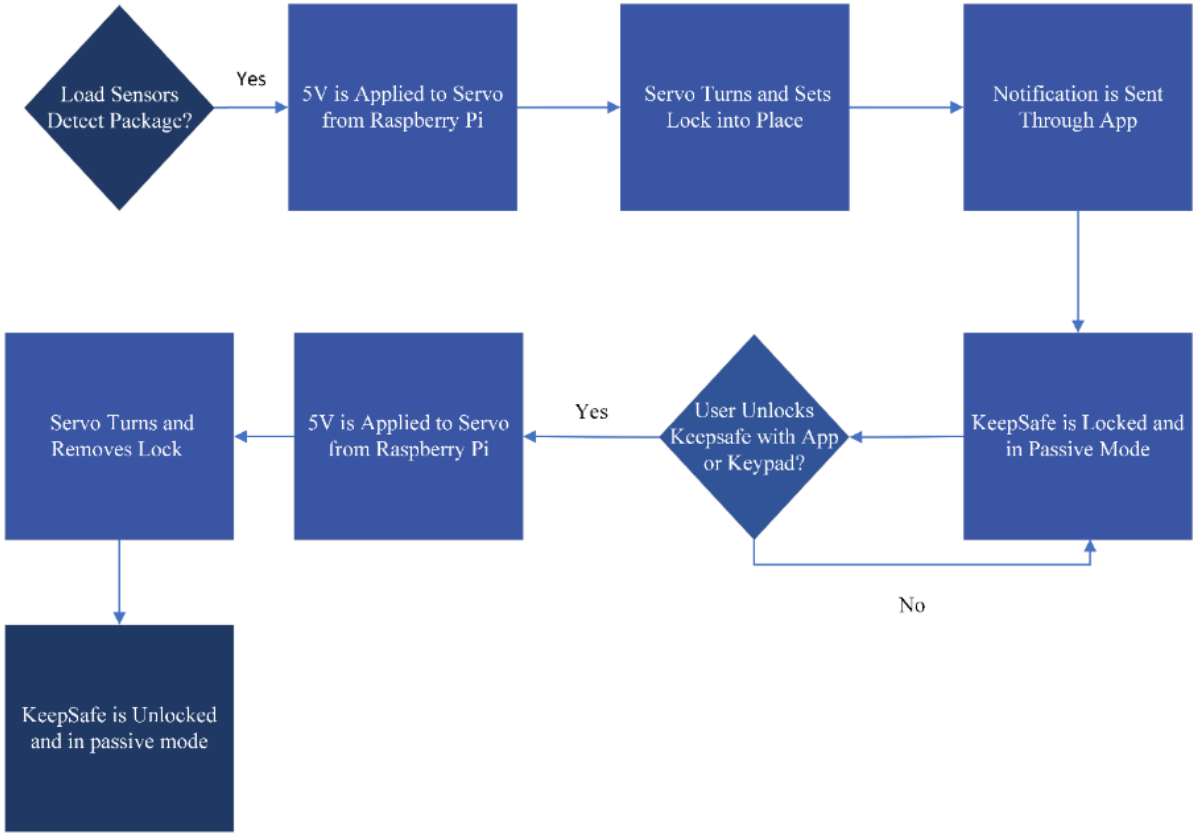


Electrical Subsystem: Code

```
234  hx.reset()
235
236  hx.tare()
237
238  print("Tare done! Add weight now...")
239  MPU_initialization() #initializes gyro
240  pir.wait_for_motion()
241  pic()
242  Gx,Gy,Gz = Gyro()
243
244  #After KeepSafe detects motion, it checks for movement of the box it
245  #if any is detected, KeepSafe goes into safety mode
246  if((abs(Gx)>=GxLimit or abs(Gy)>=GyLimit or abs(Gz)>=GzLimit)):
247          DoorLock()
248          GPS()
249          print("Program end");
250  else:
251      button.wait_for_release()
252      GPIO.output(GPIO_LIGHT,True)
253
254      time.sleep(5)
255      for x in range(numCount):
256
257          # Prints the weight.
258          val = hx.get_weight(5)
259          #print(val) #debug line
260          TotalWeight += val
261
262          hx.power_down()
263          hx.power_up()
264          time.sleep(0.1)
265
266      button.wait_for_press()
```

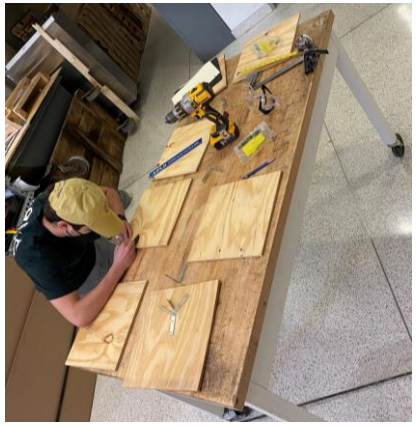
```
266      button.wait_for_press()
267
268      GPIO.output(GPIO_LIGHT, False)
269
270      AvgWeight = TotalWeight/numCount
271
272      if(AvgWeight >= packageRef):
273          DoorLock()
274          print("the weight of the package is ", AvgWeight, 'g')
275
276      while(Door_locked):
277          kt = k1+k2+k3+k4
278          #print(kt) #debug line
279          readLine(C1, ["3", "4", "1", "2"])
280          time.sleep(0.1)
281
```

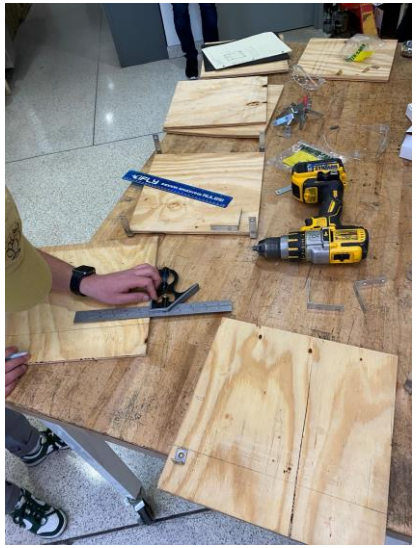
Mechanical Subsystem

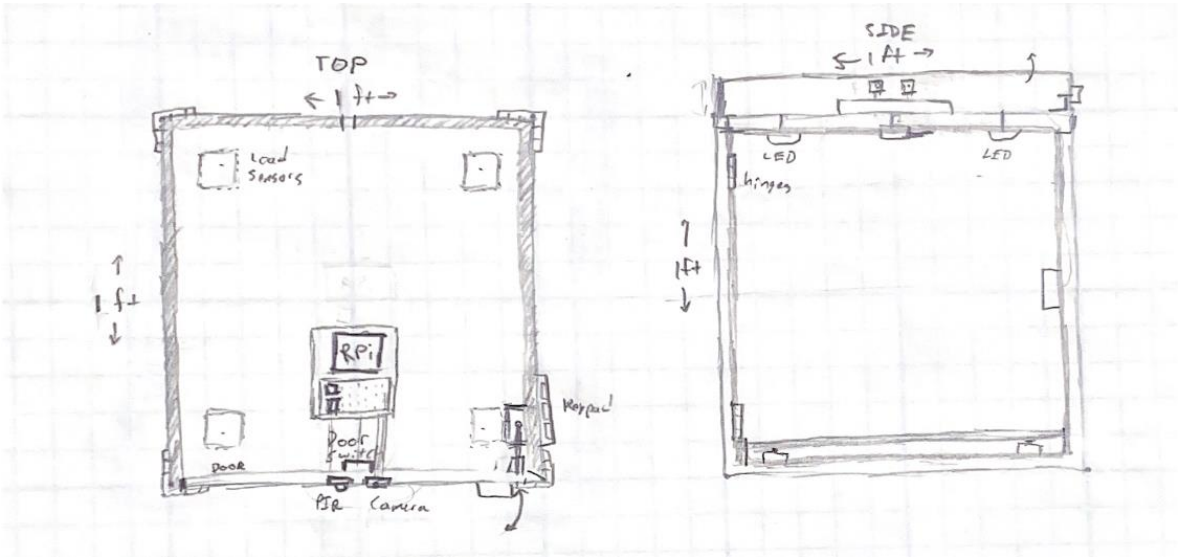


Requirements	Specifications
1. The device must secure the package	1a. Automatically locks the packages inside the box when delivered 1b. Locks must engage in under 0.5s 1c. User can unlock the box through phone app control or a manual keypad input on the box itself 1d. Lock must engage every time with no error
2. The device must utilize materials and design to prevent thieves potentially taking the device itself	1a. Materials used must be strong and heavy to prevent thieves from trying to pick up and steal the entire device. The device can potentially be bolted down as well to further prevent the device being stolen 1b. GPS system must be used and must be accurate within a 2m radius 1c. Motion sensors must be combined with camera to take pictures of any disturbances. When an object is within a 7m range for over 10s, camera must take picture and save the photo
3. The device must be easy to use by delivery driver to ensure consistent package security	3. No external input needed from the delivery driver, must be able to simply open the box and place the package inside to ensure the device is consistently used by the delivery driver

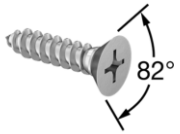
Mechanical Subsystem

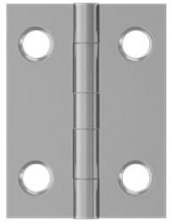


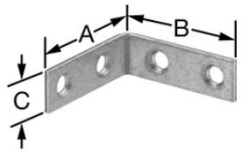




Mechanical Subsystem



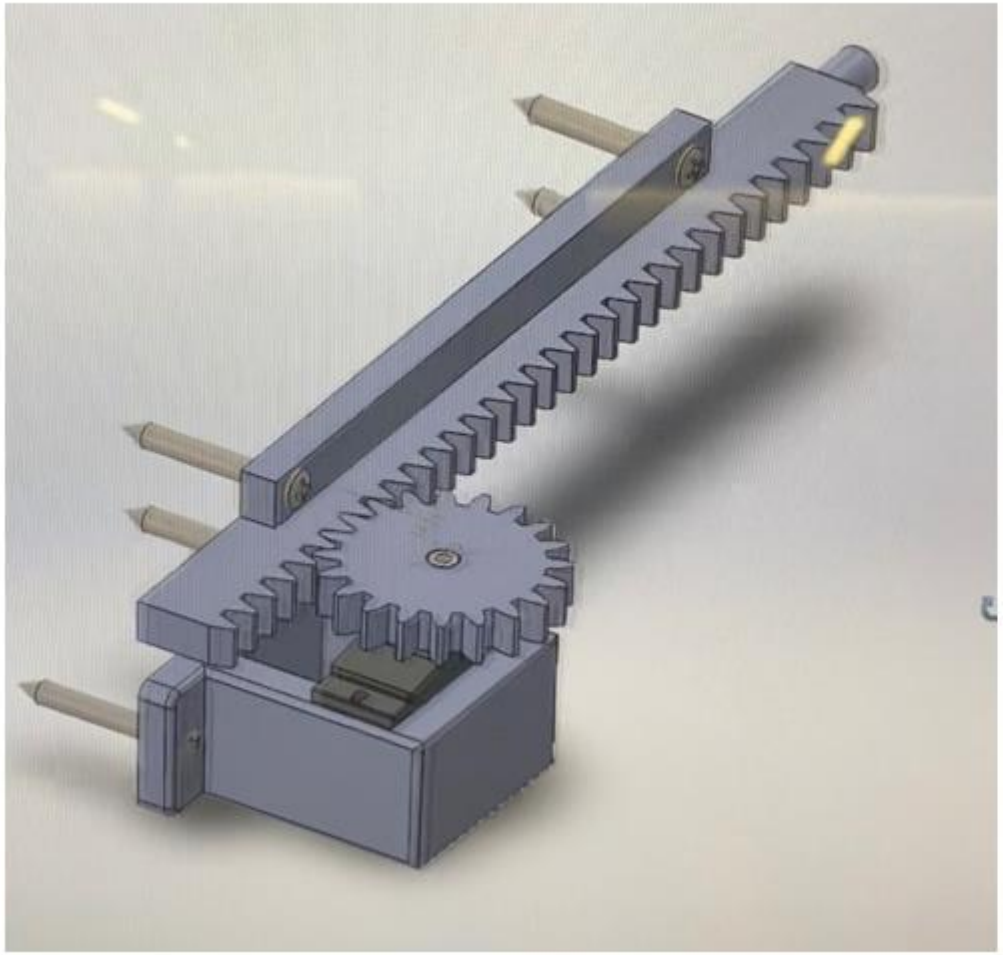


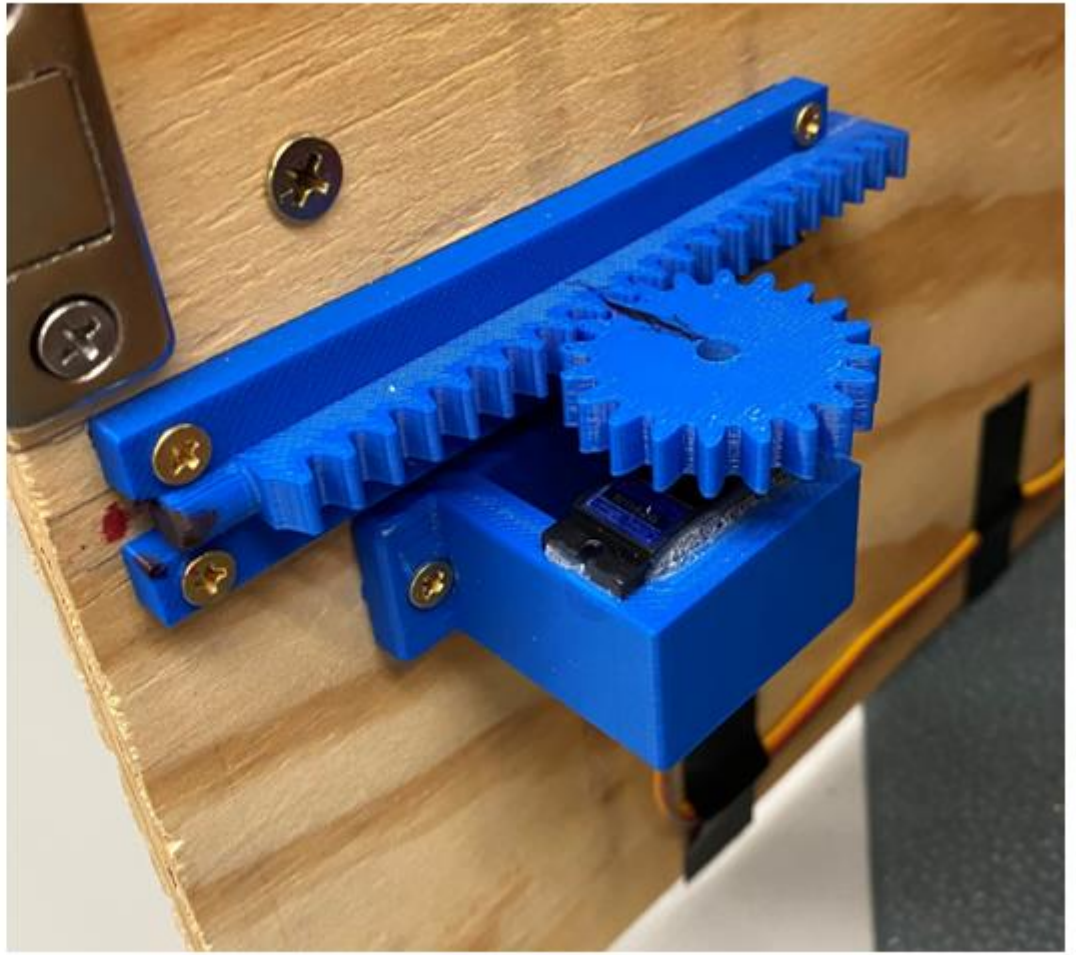


McMASTER-CARR

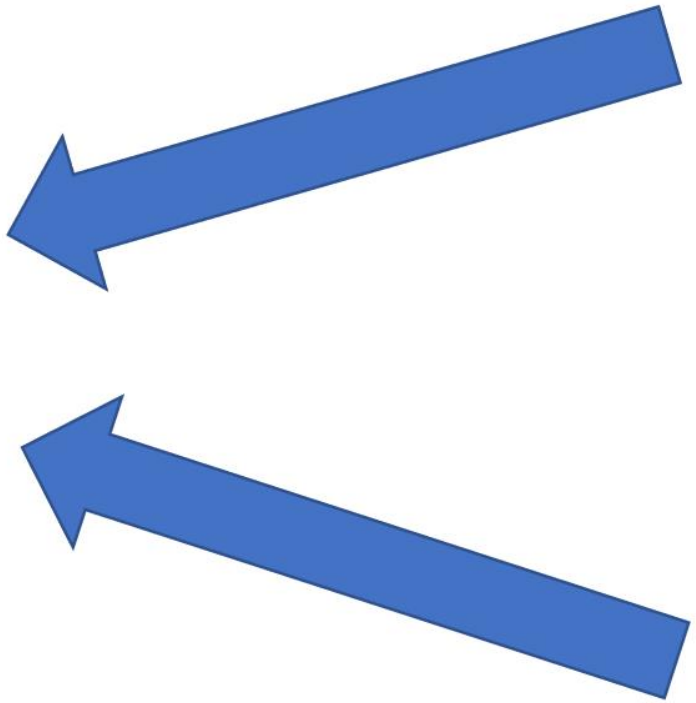
amazon

Mechanical Subsystem

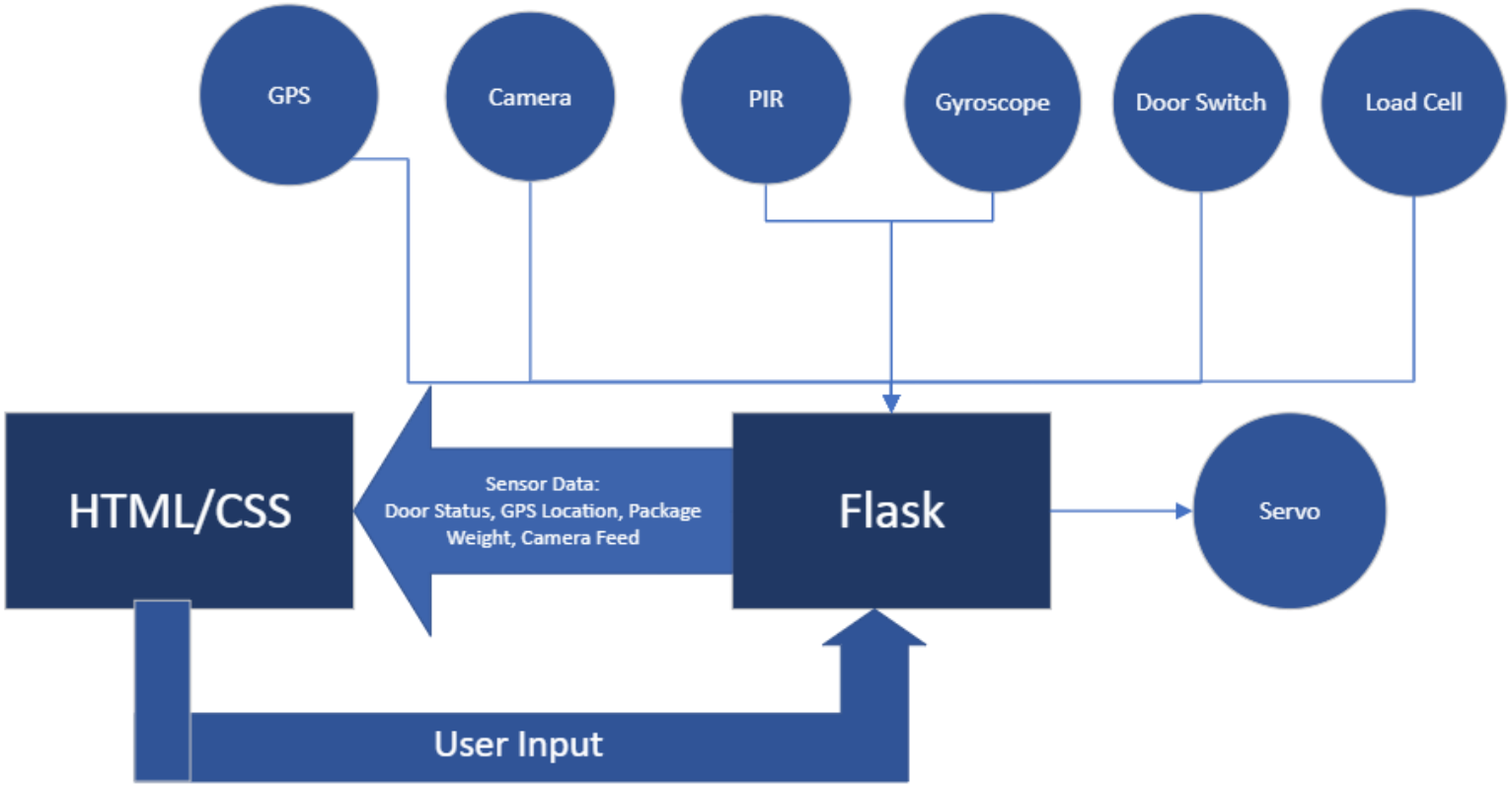




Application Subsystem

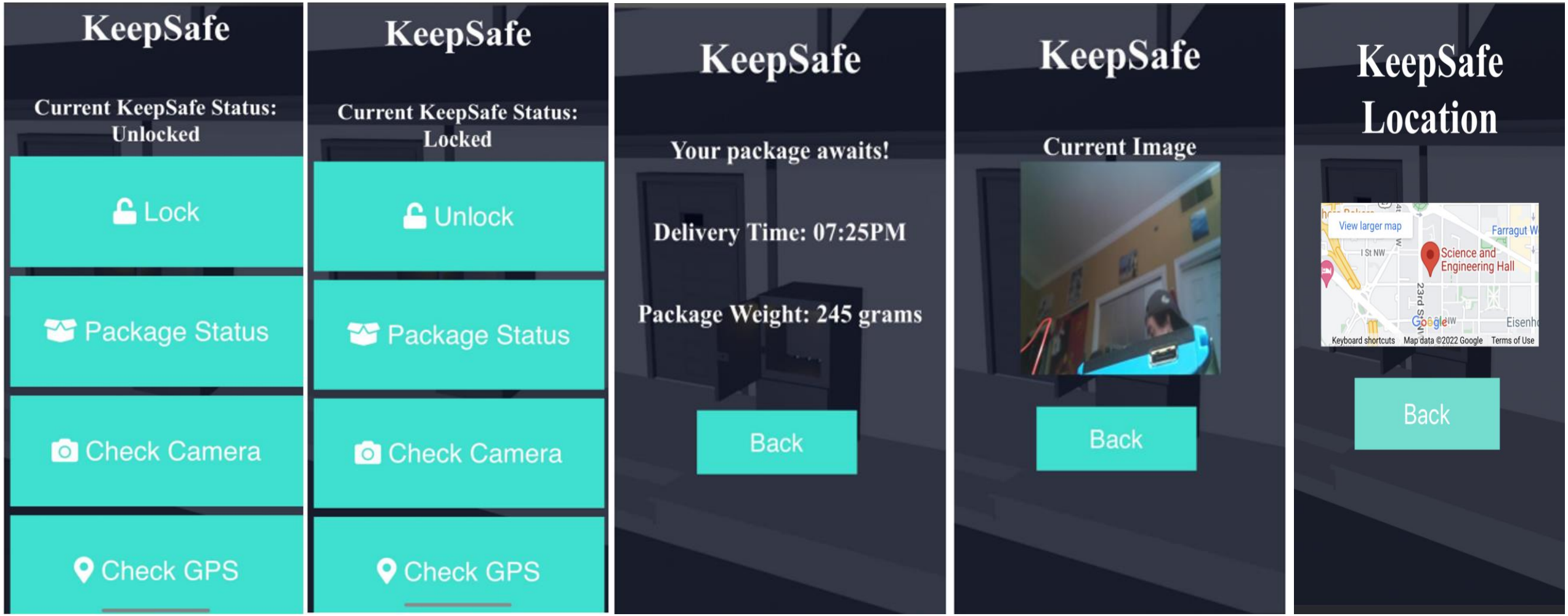


Application Subsystem



Requirements	Specifications
User must control locking mechanism	Easy to understand, intuitive interface for all controls
User must be able to see live camera feed	Seamless navigation between pages with minimum downtime
User must be able to see GPS Location	
User must be able to see package information	Information includes package weight and delivery time

Application Subsystem



Application Subsystem

FLASK:

- Importing Libraries
- Declaring Pins and I/O
- Functions for Sensors
- Functions to Display Webpages

- 19 Libraries
- 15 Pins
- 15 sensor functions
- 8 webpage functions

```
120 > def CheckDoorStatus(): #Function to determine door status···
135
136 > def WaitForPackage(): #This function turns on the lights, then takes 5 readings and averages them to determine package weight···
161
162 > def MeasureWeight(): #This function can be called at anytime, and is not tied to the door functionality···
177
178 > def DoorLock(): #Turns of lights, and turns servo to lock···
189
190 > def DoorUnlock(): #Unlocks the door and resets keypad controls···
205
206 > def fixInitial(): #Sets the initialize boolean to 1···
210
211 > def InitialUnlock(): #KeepSafe begins operation unlocked···
218
219 > def GetLockTime(): #This function returns the time when the door was last locked···
223
224 > def readLine(line, chars): #Keypad function. Code is 2143···
287
288 > def video_stream(): #Function to get live camera feed···
297
298 > def MPU_initialization(): #Initializes gyroscope settings···
304
305 > def Read_data(reg_add): #Activates serial bus for gyroscope···
315
316 > def GyroCheck(): #Begins reading gyroscope data. If the reading is higher than assigned threshold, the door will lock···
339
340 > def NMEAConverter(data,direction): #NMEA converter for GPS. Outputs coordinates in latitude and longitutde···
360
361 > def GPS(): #Takes data from NMEA converter and outputs values to be used for google maps···
```

Application Subsystem

5 Unique HTML Pages

```
<!DOCTYPE html>
<html>
<head>
        <meta name ="viewport" content="width=device-width, initial-scale=1.0"> <!-- Setting up html, making sure it adjusts to screen size -->
        <meta http-equiv="refresh" content="5"> <!-- Refreshes page every 5 seconds to check sensors -->
        <title> KeepSafe Home </title> <!-- Title of page -->
        <link href = "{{url_for('static', filename = 'style.css')}}" rel="stylesheet" type="text/css"> <!-- Adding CSS style, fonts, and icons -->
        <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
        <script src="https://kit.fontawesome.com/9d4ad59c28.js" crossorigin="anonymous"></script>

        <style>

        .header{
            min-height: 100vh;
            width: 100%;
            background-position: center;
            background-size: cover;
            position: relative;
            background-image: linear-gradient(rgba(4,9,30,0.7), rgba(4,9,30,0.7)), url({{url_for('static', filename='Render2.png')}});
            }
        </style> <!-- Added style to put KeepSafe Image as background -->
</head>
<body>
    <section class="header">
        <h1> KeepSafe </h1>
        <h2> Current KeepSafe Status: Unlocked</h2> <!-- Titles  -->

    <form action="/Lock">
    <button class="button  btn--block"><i class="fas fa-unlock"></i> Lock </button></form> <!-- Each button is linked to a flask function, and an icon is added to each button -->

    <form action="/PackStat">
    <button class="button  btn--block"><i class="fas fa-box-open"></i> Package Status </button></form>

    <form action="/pic">
    <button class="button  btn--block"><i class="fas fa-camera"></i> Check Camera </button></form>

    <form action="/gps">
    <button class="button  btn--block"><i class="fas fa-compass"></i> Check GPS </button></form>

</section>
</body>

</html>
```

Application Subsystem

1 CSS Script for Design

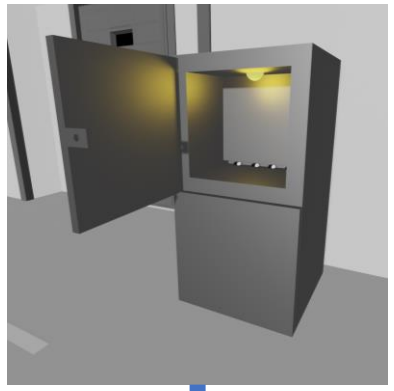
```css
*{
    margin: 0;
    padding: 0;
} /* Setting webpage to display with no border in center */

.header{
    min-height: 100vh;
    width: 100%;
    background-image: linear-gradient(rgba(4,9,30,0.7), rgba(4,9,30,0.7)),url(images/Render2.png);
    background-position: center;
    background-size: cover;
    position: relative;
}/* Centering background image*/

.CurrentImage{

    height: 450px;
    width: 400px;

    color: white;
    padding-top: 500px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 36px;
    margin: 4px 2px;


    top: 100px;
    left: 300px;
    background-image: url(images/SeanLetavish_Picture.jpg);
    background-size: cover;
    position: relative;

} /* Setting size for the live video feed */
```

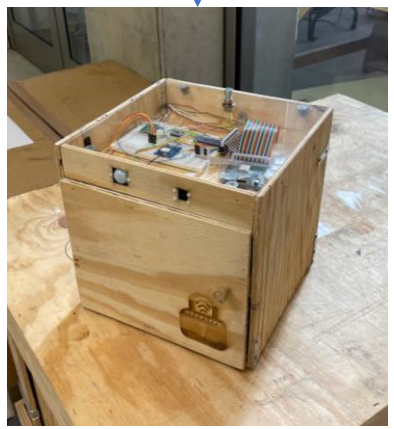
```css
.button {
    display: inline-flex;
    height: 250px;
    width: 200px;
    border: none;
    color: white;
    padding: 0;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 36px;
    margin: 4px 2px;
    cursor: pointer;
    background: #40e0d0;
    position: absolute;
    top: 300px;
    left: 560px;
} /* Creating buttons. Buttons are t

.button:hover {
    background: #34b7aa
} /* Button slightly darkens when ho

.button:active {
    background:#227a71
} /* When pressed, button darkens ev
```

```css
.button4:hover {…
}

.button4:active {…
}



h1{
    color:white;
    text-align: center;
    padding-top: 30px;
    font-size: 50px;
} /* Setting headings size, color, a

h2{
    color:white;
    text-align: center;
    padding-top: 60px;
    font-size: 30px;
}

h3{
    color:white;
    text-align: center;
    padding-top: 60px;
    font-size: 30px;
}

h4{
    color:white;
    text-align: center;
    padding-top: 60px;
    font-size: 30px;
}
h5{
    color:white;
    padding-left: 400px;
    padding-top: 200px;
    font-size: 30px;
}

p{
    color:white;
    padding-left: 1000px;
    font-size: 30px;
}
```

Evolution of Overall Design

- Exterior design changes
- Power consumption changes
- Multiple adjustments in these areas would be needed for KeepSafe to be sold as a commercial product
- The team was able to successfully demonstrate the desired functionality and satisfy the requirements/specifications





Sleepy Pi 2 - Micro USB B - Spell Foundry

Product Demonstration

Lessons Learned

- Effective component selection
- Working as a team
- Debugging code and hardware
- Working in the machine shop to construct prototype
- Design for testing/testability measures
- Multidisciplinary design and integration
- App development
- Plywood bought online is not always going to be perfectly square

Conclusion

- The team was fully aware and followed both safety and engineering standards to design, construct, and test the product
- The team was able to successfully create a working KeepSafe prototype the fulfills the requirements and specifications
- Working collaboratively, the team utilized their individual skill sets and applied what they have learned over the past four years

Thank you!

- Professor Aslani
- ECE Department
- Machine Shop

Questions?