Homing Security

A Mary Swords Design

Submitted in partial fulfillment of the requirements of Embedded Senior Project (CST 473)

Revision History	Updated by	Reason	Date
Final Proposal Report	Mary Swords	Final Proposal Report	June 8, 2020
Updated with Price Comparison Tables	Mary Swords	To make additions for Fall Term	November 1, 2020 November 11, 2020
Requirements Update	Mary Swords	Updated requirements to fit with current design	November 28, 2020 December 2, 2020
Final Submission	Mary Swords	Final Submission	December 5, 2020
Critical Design Discussion Update	Mary Swords	Updated for Week 5	January 30, 2021 January 31, 2021
Removed unnecessary requirements	Mary Swords	Week 6 email Update	February 9, 2021 February 12, 2021
Final Winter Term Update	Mary Swords	Final Submission	March 13-16, 2021
Final Spring Term Update	Mary Swords	Final Submission	June 1, 2021

This proposal is approved as part of the requirements for class CST473, Embedded Project Proposal.

Mary Swords, Student Date

Kevin Pintong, Professor Date

Troy Scevers, Advisor Date

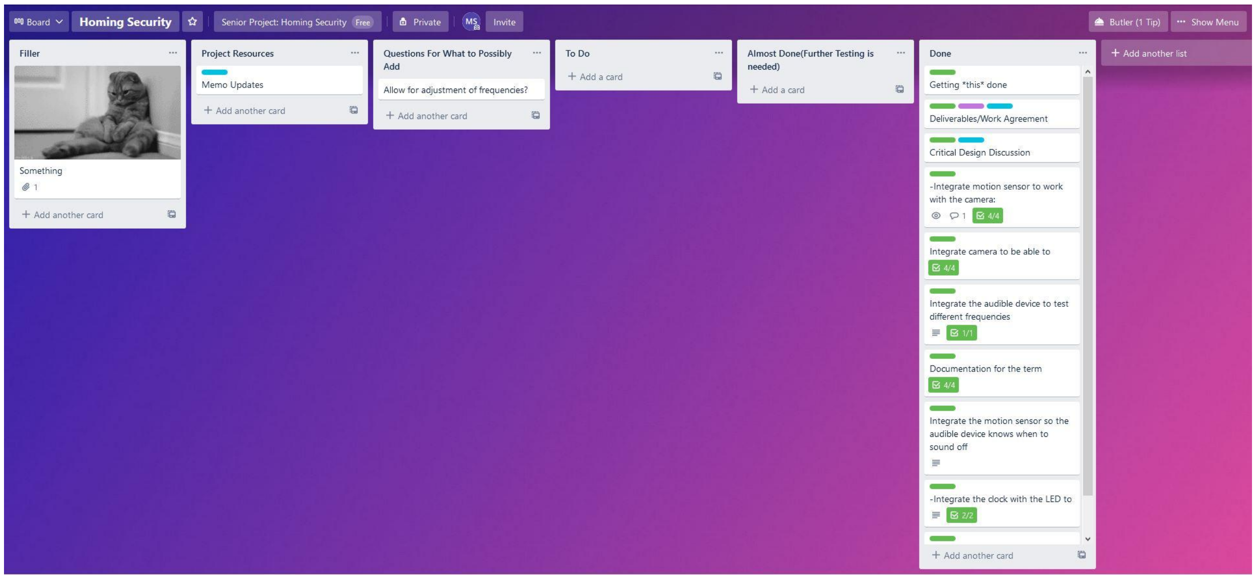
Abstract

The project is called Homing Security, which is meant to be a cost-effective home security system. Most security systems are very costly, so this project intends to make a security system that is cost-effective and simple to use without the need for a phone application. The document will explain the timeline of the building process of the device, and have what fields are required to make the device. It will also explain how certain aspects of the system are intended to run, including a system block diagram, hardware and software descriptions, and how the subsystems of the device will be implemented. The features the device will have is an audible device which will act as an alarm, a camera that takes pictures and records videos, and a removable storage device that the user can connect to a computer to access files taken by the security camera. There will also be a notification light on the device so the user can see when last there was a security breach. Overall, the security system is meant to be able to do what other systems can, but for a lower price, and be easier to use without a phone application.

Table of Contents

Project Management

Project Time Management Chart



Board · Homing Security · Senior Project: Homing Security Free · Private · MS · Invite · Butler (1 Tip) · Show Menu

Filler
- Something
  1
- + Add another card

Project Resources
- Memo Updates
- + Add another card

Questions For What to Possibly Add
- Allow for adjustment of frequencies?
- + Add another card

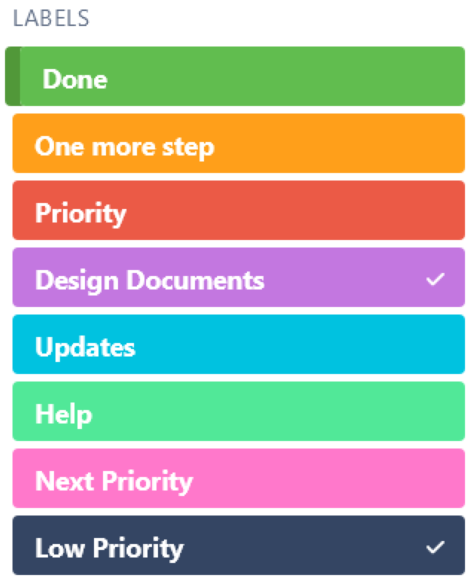
To Do
- + Add a card

Almost Done(Further Testing is needed)
- + Add a card

Done
- Getting *this* done
- Deliverables/Work Agreement
- Critical Design Discussion
- -Integrate motion sensor to work with the camera:
  1 4/4
- Integrate camera to be able to
  4/4
- Integrate the audible device to test different frequencies
  1/1
- Documentation for the term
  4/4
- Integrate the motion sensor so the audible device knows when to sound off
- -Integrate the clock with the LED to
  2/2
- + Add another card

+ Add another list

Figure 1



LABELS

- Done
- One more step
- Priority
- Design Documents ✓
- Updates
- Help
- Next Priority
- Low Priority ✓

Figure 2

Spring 2021	Week 1	Week 2	Week 3	Week 4
	Begin Work on housing design	Continue to work on housing design	Continue working on improving design	Continue working on improving design
	Continue working on improving design	Continue working on improving design	Update documents	Update documents
	Update documents	Update documents		
	Week 5	Week 6	Week 7	Week 8
	Housing Completed	Update documents	Update documents	Update documents
	Update documents			
	Week 9	Week 10	Finals	
	Finish project	Finish project	Done!	
	Finish Documents	Finish Documents		

Figure 3

Comparison Tables

Development Board

	Raspberry Pi 4 8GB	Raspberry Pi 4 B	Raspberry Pi 3 Model A+	Raspberry Pi Zero WH
Price	$75 USD	$35 USD	$25 USD	$15 USD
SoC Type	Broadcom BCM271	Broadcom BCM2711	Broadcom BCM2837B0	Broadcom BCM2835
Core Type	Cortex-A72 (ARM v8) 64-bit	Cortex-A72 (ARM v8) 64-bit	Cortex-A53 64-bit	ARM1176JZF-S
# of cores	4	4	4	1
GPU	VideoCore VI	VideoCore VI	VideoCore VI	VideoCore VI
CPU Clock	1.5 GHz	1.5 GHz	1.4 GHz	1 GHz
RAM	8 GB LPDDR4	1 GB , 2 GB, 4 GB LPDDR4	512 MB DDR2	512 MB

Figure 4

Motion Sensors

Part	Pros	Cons
HC-SR501 PIR	● Compatible with the Pi4 ● Is the most cost efficient ● Small time delay of 0.5s	● Sensing range is not long enough for the product specifications ○ Less than 10 ft ● Higher input voltage
287-18001 PIR Sensor	● Compatible with the Pi4 ● Small delay time of 0.3s	● Sensing range is not long enough for the product specifications ○ Less than 10 ft ● Very similar to the previous mentioned ● Higher input voltage
PIR Motion Sensor - Large Lens version	● Compatible with the Pi4 ● Lowest input voltage	● Sensing range is not long enough for the product specifications ○ Less than 10 ft ● Highest cost of the three ● Very similar to the others

Figure 5

Cameras with Infrared Capabilities

Part	Pros	Cons
MakerFocus Fisheye Camera 5mp IR-CUT Night-vision Camera	● Is able to swap between night and day vision ● Infrared filter	● Larger compared to the others ● Second Most Expensive
Pi NoIR Camera V2	● No infrared filter, so it can take normal pictures in the day and at night ● Small to potentially keep the size of the final product down	● Because of the no infrared/night filter filter, separate infrared capabilities would have to be purchased separately. ● Most expensive
Arducam 5MP Camera for Raspberry Pi	● Most cost efficient ● Very small for ensuring the device is small	● Does not have night vision/infrared capabilities

Figure 6

Real-Time Clock Modules

Part	Pros	Cons
Mini RTC for Raspberry Pi	2nd Cheapest, Compatible with most Pi's	Not the cheapest
DS3231 High Precision RTC Clock Module for Raspberry Pi B+	Cheapest in bulk	Not compatible with most Pi's

Figure 7

Updated Cost Estimation Table

Parts	Quantity	Price
Raspberry Pi 4 B	1	$47
HC-SR501 PIR	5	$9.49
Makerfocus Camera	2	$43.98
RGB LED	100	$5.00
SD Card	1	$9.49
USB 2.0	1	$7.99
Piezo Buzzer	3	$4.50
Housing	1	$7.11
RTC Component	5	$11
Total		$145.56

Figure 8

Labor Costs Table

Labor	Run (in minutes)	Cost/Hour	Total
Wiring	30	$10	$5
Testing Components	45	$20	$15
Programming	150	$20	$50
Debugging	150	$20	$50
Design	300	$20	$100
Total			$220

Figure 9

Non-Recurring Engineering Cost Estimate

Starting Wage		$75 per hour
Total Hours		300 hours
Total Amount		$22,500
Overhead	30%	$6,750.00
Overall Total		$29,255.00

Figure 10

Change Management Procedures

Any changes that are requested by the engineer will be sent in a memo to Kevin Pintong through email. The engineer will then wait for the Professor's approval before going through with the design changes. The update memos will be in a PDF format, and will show changes to the Project Time Management Chart by indicating a new section highlighted in yellow to show what the changes are, and what part of the timeline it will affect the chart. Currently, Excel sheets are being presented in this document, but the schedule was moved over to Trello by Winter Term of 2021.

Status report

The status of the project will be updated once a week through a memo or at the Professor's request sent through email to Kevin Pintong. These memos will be in a PDF format, and will show the progress made by using the Project Time Management Chart. The chart will have a section that will indicate what has been completed in yellow.

Skills and Knowledge Areas

Electronics - The knowledge of electronics will apply to circuitry with devices, and how to properly connect them. This will require caution as it may damage the parts if not done correctly, and cost more time in the long run. Mathematical circuit equations will be needed, and thus knowing how resistors and a power supply will affect the device. How power runs through a system is also required, so doing mathematical power equations will be needed.

Cameras - The knowledge needed will be working with camera resolution and how it will be able to connect to the development board. There is limited knowledge here, so it will need to research in order to understand how the camera will be compatible with the development board. The camera resolution is necessary so the user can see the image and video clearly, and also how to manipulate how the image will appear to the user.

Sensors, microcontrollers, and Raspberry Pi with Python - Python has been chosen as the coding language to be used for the system. Any other programming language can be used, but will need to be compatible with the microcontroller and also will need knowledge of beforehand. This is due to limited knowledge of Python, so using another unlearned language will take up more time than what might be given. The Linux type console for the Raspberry Pi 4 will also be used in order to help properly set up and control the different parts to be used in the project. The sensors and microcontroller must be compatible with the programming IDE, otherwise there will be difficulty in setting up the project. As of currently developing, the real time clock module has been the most difficult to set up, so there will be expected problems with that in the future of developing the project. The same will be for the camera in order to have it become integrated with the project without having the need of using a command line in order to operate it.

Conceptual Overview

Introduction

The product being created is a home security system. Security systems for in-home use are useful as they give more safety to people who have them [1]. They help with catching intruders, which also helps the police force in identifying the intruder quicker. A security system can also deter intruders, so the theft may never occur. Security systems also are shown to have a correlation to decreases in theft, so having one in home is very invaluable.

Problem Statement

There are already many security systems that have been produced, but tend to be expensive if paid for over long periods of time. Even ones that require a single payment usually cost hundreds of dollars. Another problem is that a lot of self-installed home security systems require the use of smartphones, something that not every user may be familiar with. Overall, security systems are useful as they give a sense of security to the user as they are reassured that their home is being monitored, even while away.

Intended Audience

The device is meant to be for people who do not yet own a security system due to the complexity of one, or not being able to afford one. Giving the target audience a new option would be beneficial as they would have a more cost-effective option to go with that also provides more manageability. By being more manageable, it will be less of a hassle to understand how it works and to maintain it. It will also provide a technology limited system so there is no need for a lot of technology to be required in order to operate the system.

Proposed Project

The project that is being proposed is a home security system that is meant to be cost-effective and more manageable for the user. It will have a camera that is on constantly, and the user will be allowed to access the device's storage to check security footage taken throughout the day. It will also have an LED to notify how recent the security system was tripped so the user can take action as needed.

Relation of Proposed System to Existing Systems

There are many security systems on the market currently, so the comparison table in Figure 3 will briefly compare the Homing Security to three other security systems: SimpliSafe, Frontpoint, and Wyze Cam.

System	Price	Monthly Fees	Accessible Files	Package Deal
Homing Security	~$150	None	Yes	None
SimpliSafe	~$229 and up	~$14.99 and up	None	Yes
Frontpoint	~$319.95 and up	~$45.01 and up	None	Yes
Wyze Cam	~$19.99	None	Yes	None

Figure 11

The first section of the table in Figure 3 compares the starting price and monthly fees of the existing products [3]. Here it shows that the Homing Security will not only be cheaper in the starting price, but have no monthly upkeep as it is a system meant to be maintained by the user. The next section shows the accessibility of the file storage. The other security Systems, such as SimpliSafe [4], do not allow for accessing the system files, like Homing Security. With Homing Security, the user will be able to remove a storage device where they can view the pictures and videos taken on a computer, which allows a sense of control with the system. The last section is called Package Deal, which goes with the listed price. SimpliSafe and Frontpoint both have add-ons to the security system [5], which many people would feel the need to get. With Homing Security, it will not have add-ons, like Wyze Cam [6]. To some, this may seem like a drawback, but the fewer options people have is usually better. The more the people have to pay for things, the more likely they might not use it, and having less features can sometimes better enhance a design [7]. This is another reason why Homing Security is beneficial as the customers will only have to make a one time payment, and not have to worry about buying separate add-ons that might be unnecessary [8]. However, this all applies to Wyze Cam, but one difference between the two is that the Homing Security will come with a storage device, so the user will not have to purchase one. Another aspect to Homing Security is that it does not use storage as quickly. Wyze Cam is always recording videos, whereas Homing Security only records video when it detects movement. If it does not, then it will be taking photos instead, which take up less memory. This is a benefit because the user will be able to access more older files than if they had Wyze Cam. Another benefit is that Homing Security also does not use a phone application. To some, this might seem to be a detriment, but this makes the system more simple to understand. There is no hassle for connecting the device to the system, or a learning curve for learning the phone application. Instead, the user is able to have the system installed and just have to worry about the files that are stored onto the device. Another aspect that Homing Security has is the notification LED that will be placed on the outside of the system's housing. This will notify the user how long it has been since the motion sensor was last tripped. This means the user can just look at the device and do nothing else. Even though Wyze Cam has good options, Homing Security has options that the former doesn't have, which makes it stand out.

Deliverables

The final deliverables will be the device itself with its housing. There will also be documents provided to show operations of the system, and how it was designed. The project's documents will contain information on the system so that it will be understandable to see how the system was designed. Source code will also be provided in some parts of the documentation to give a better explanation as to how each sub-system functioned within the device.

System Description

System Block Diagram

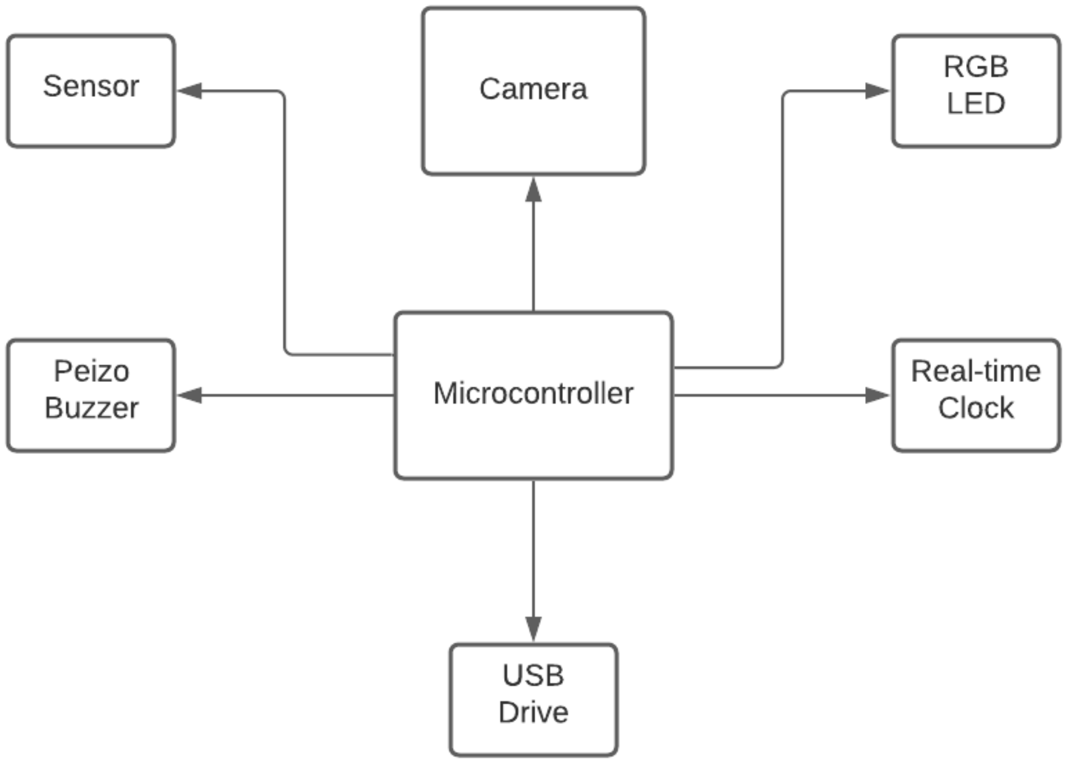


Figure 12

System General Description

The system will generally work by having it active when the user turns it on. The system will then not fully activate until 30 seconds have passed. Afterwards, the camera will take pictures every 30 seconds. If the sensor is tripped, then it will take a video recording that will last for one minute. The motion sensor will send this signal to the microcontroller, which will then tell the camera to start taking a video recording. When the sensor is tripped, the audible device, or alarm, will make a sound that will be loud enough to potentially scare away the intruder. As for the

LED's, there will be one to keep track as to how long it has been since the sensor was last tripped. The LED will need a clock system to activate, so the device will also have an internal clock to keep track of time. This clock system will be useful for also keeping track as to when pictures and videos were recorded onto the system storage. The final design measurements can be referred to in *Appendix III*, but they were slightly altered in the final design to add hinges to the box so the top could open up for easy access for the designer.

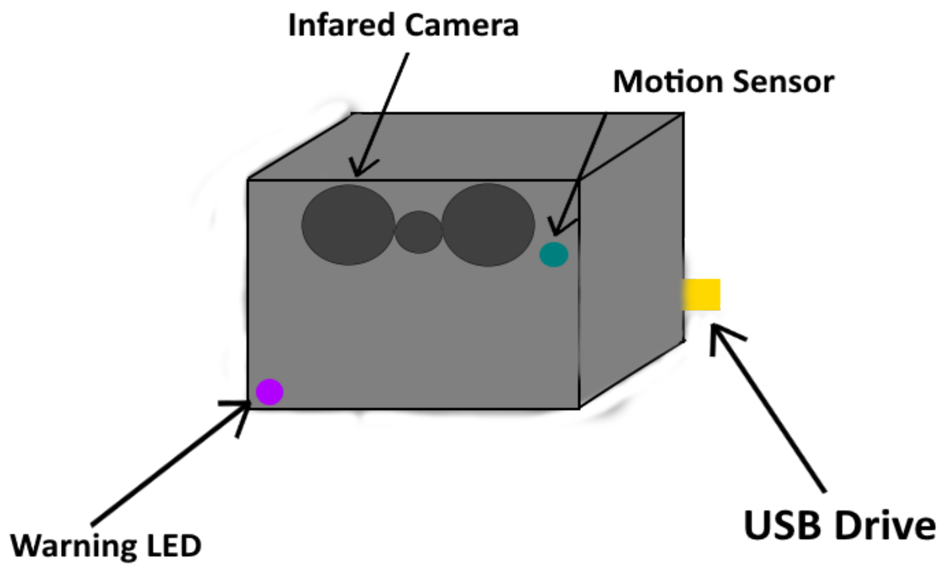


Figure 13

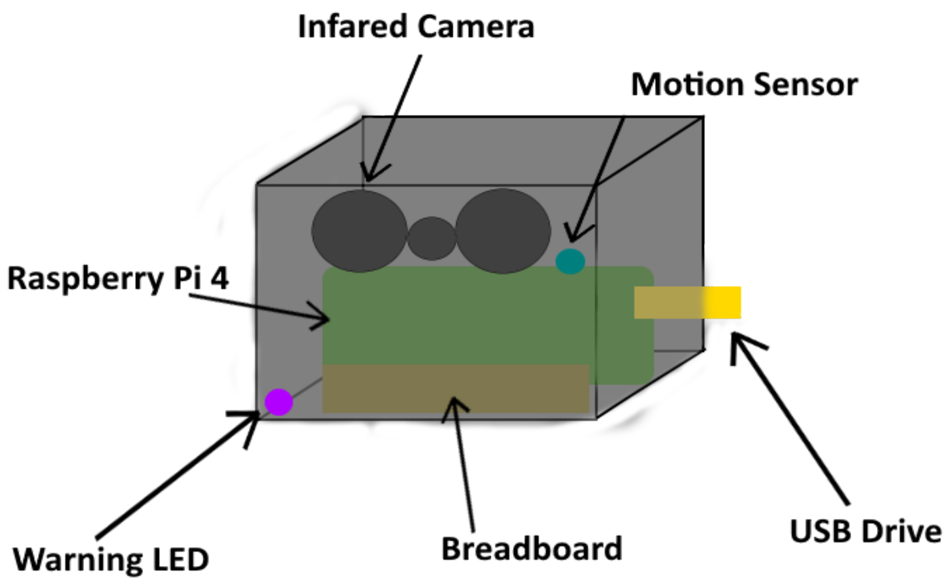


Figure 14

Major subsystems

The major subsystem will be the internal clock, which will allow the device to keep track of time. This will be useful for the LED that works based on time. There will also be a power system for the device, which will need to be connected to a power outlet in order to provide consistent power. This will work alongside the audible device when the system is turned on, so it will do a 30 second countdown to let the user know when the system will start functioning.

Hardware Development

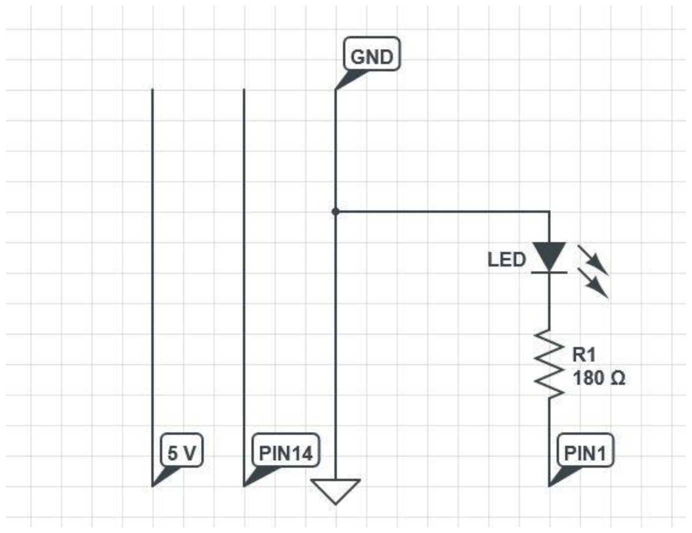


Figure 15

The hardware element that has been chosen for the development board is the Raspberry Pi 4. Other elements included will be a sensor that detects motion, which will be the HC-SR501 sensor. The RGB LED will be a standard one bought in bulk to allow for many trials in case one should break. This will be used to indicate when the motion sensor was last tripped in the final design.

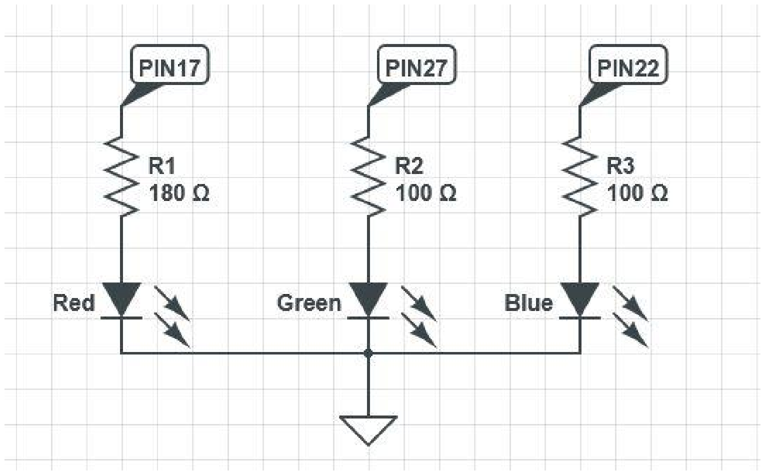


Figure 16

A camera is another element that will need to be added and the camera chosen is the Maker Focus. Despite its size, it is capable of taking videos and pictures in both day and night settings that will be clear to the user.

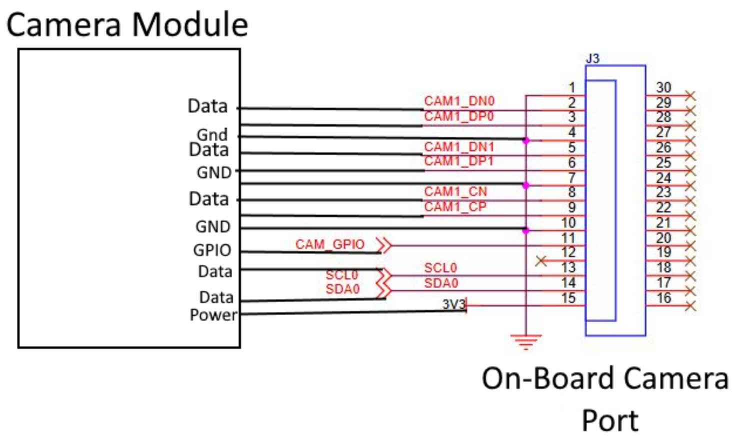


Figure 17

Then there will be a removable storage device, which will be a 2.0 USB. With those components in mind, they would need to be compatible with the development board in order for the system to function properly. The last component is an audible device, which is a piezo buzzer, that can act as an alarm for when the sensor is tripped, and as an alert system for when a device has been put into the system.

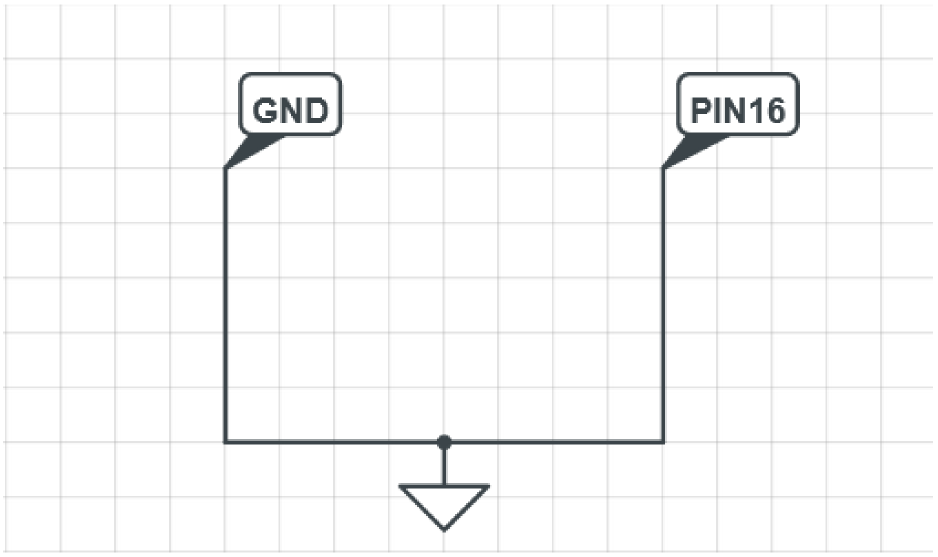


Figure 18

It will work alongside the motion sensor and camera by using multiple processes. Here it is connected to a ground pin and a GPIO pin. It is not connected through a resistor as not a lot of current runs through the small part. Figure 19 shows a state machine diagram for how the motion sensor and camera work with one another.

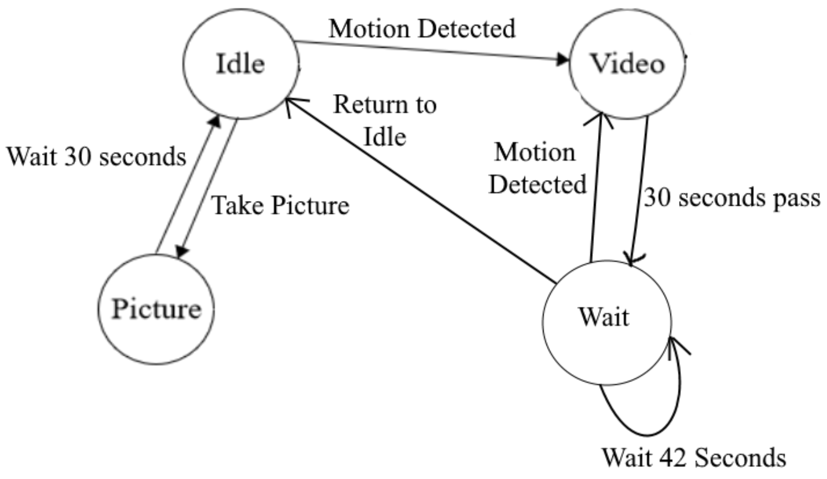


Figure 19

Figure 20 shows how the piezo buzzer works in the system, but is shown to be in a different state machine diagram. This better demonstrates how it works alongside the other components.

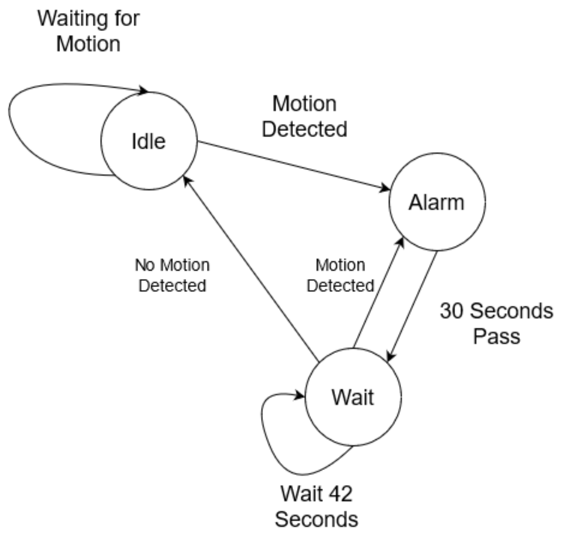


Figure 20

Alongside these two processes of the motion being detected and the alarm going off, the notification LED is also running alongside them. This happens if motion has been detected within the past 24 hours and will be reset each time motion has been detected. If no motion has been detected within the next 24 hours, then it will turn off. Figure 21 demonstrates via state machine as to how each of the states work.

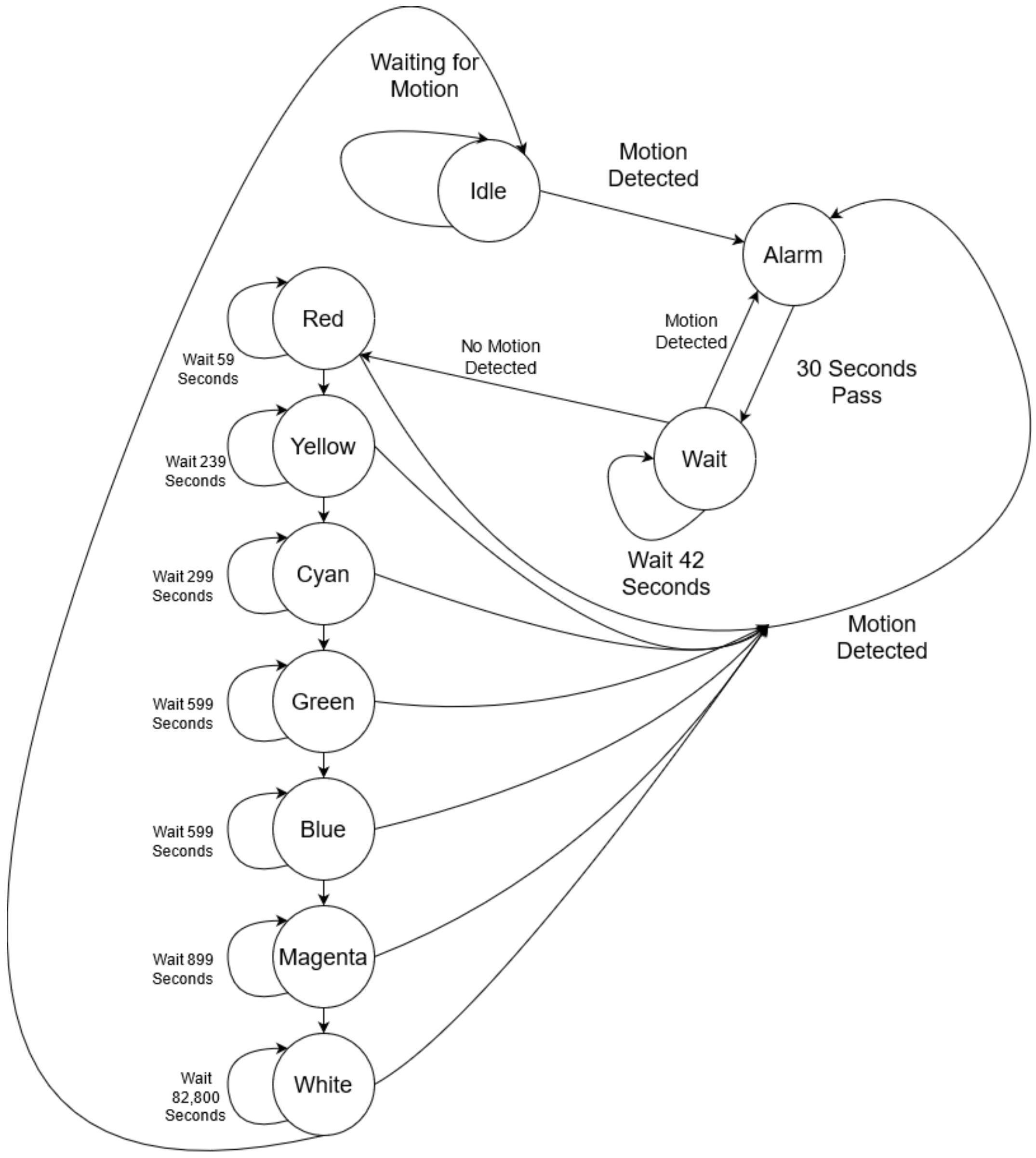
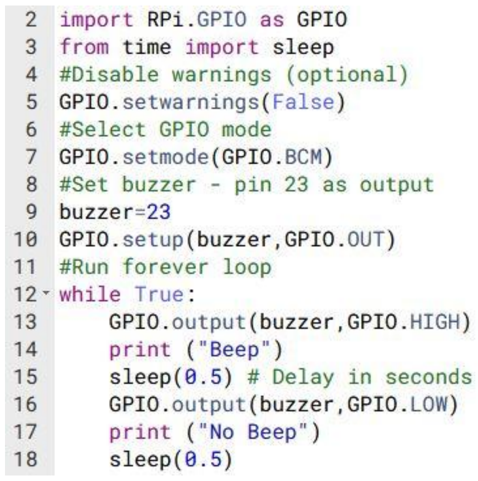


Figure 21

Software Development

For the software development part of the system, what will be used will be the Raspberry Pi 4 development IDE as that is the type of development board that was chosen. Python is also the most common scripting language used for the Raspberry Pi, so that will be chosen as there will be many resources for it. Experience in Python is currently limited, but progress has been made in developing test code for the motion sensor (see Appendix I), and RGB LED (see Appendix II). The camera[9] and the RTC clock[10] were both more difficult to set up, and online resources were needed in order to properly set them up. These required the Raspberry Pi 4 console to set up, which is very similar to the Linux console.

The next step was the piezo testing, which was determining how loud to make the sound of it so it could at least be heard through a wall and ten feet away from that wall.



```python
import RPi.GPIO as GPIO
from time import sleep
#Disable warnings (optional)
GPIO.setwarnings(False)
#Select GPIO mode
GPIO.setmode(GPIO.BCM)
#Set buzzer - pin 23 as output
buzzer=23
GPIO.setup(buzzer,GPIO.OUT)
#Run forever loop
while True:
    GPIO.output(buzzer,GPIO.HIGH)
    print ("Beep")
    sleep(0.5) # Delay in seconds
    GPIO.output(buzzer,GPIO.LOW)
    print ("No Beep")
    sleep(0.5)
```

Figure 22

From here, the frequencies were then tested to determine how loud to have the piezo buzzer be. The findings were recorded in a separate document as of now, but will later be migrated to this document for the testing section. What was found is that the faster the delay between turning on and off the output, the quieter the buzzer would be. The longer the delay, the louder it would be, up until a certain point. It was determined that the delay would be at 0.3 seconds, so it would be loud enough to be heard from another room, but not be irritating.

Another item that was integrated was the camera to be able to do the following: take pictures without user input into a command line, be able to take pictures and have them be saved in a .jpeg format, take videos without user input into a command line, and be able to record videos and have them be saved as .mp4. Figure 23 shows the flow of this and how this process can be more in line with software.

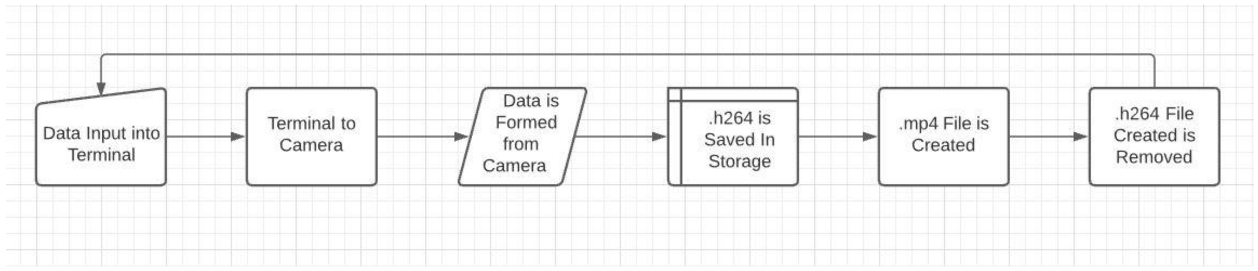
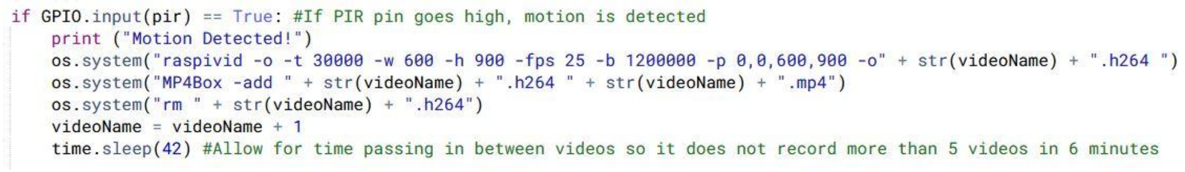

Data Input into Terminal
Terminal to Camera
Data is Formed from Camera
.h264 is Saved In Storage
.mp4 File is Created
.h264 File Created is Removed

Figure 23

As shown, there is data input into the terminal, which then talks to the camera to take a picture or a video. Then the file is saved onto the external storage by use of another command line input, then a .mp4 file is saved and the original .h264 video file is deleted. As for the external storage device, it is meant to be recognized by the system and then have the files taken by the camera stored onto it. There was also an indicator made within the code to tell the system when the storage would be close to getting full. As of now, it works, but it will be updated in Spring Term so it works more efficiently.

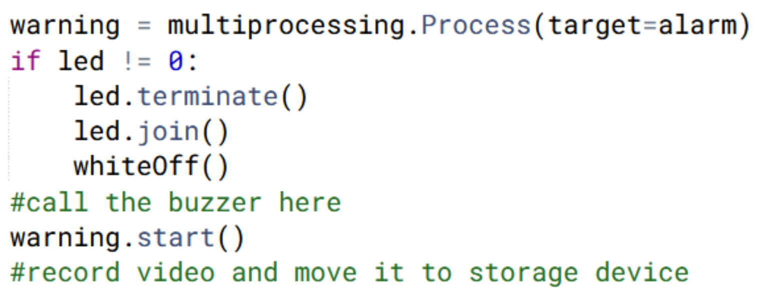
In order to have some of the code work, certain functions are called. In Figure 24, the system call for os is used so it can talk with the terminal without the user having to manually input the commands themselves.



```
if GPIO.input(pir) == True: #If PIR pin goes high, motion is detected
    print ("Motion Detected!")
    os.system("raspivid -o -t 30000 -w 600 -h 900 -fps 25 -b 1200000 -p 0,0,600,900 -o" + str(videoName) + ".h264 ")
    os.system("MP4Box -add " + str(videoName) + ".h264 " + str(videoName) + ".mp4")
    os.system("rm " + str(videoName) + ".h264")
    videoName = videoName + 1
    time.sleep(42) #Allow for time passing in between videos so it does not record more than 5 videos in 6 minutes
```

Figure 24

There are also multiple processes working alongside this to allow for multiple items to be used at once without crashing the system. The system call ‘multiprocessing’ is used to allow for the multiple processes to work synchronously instead of asynchronously. Figure 25 demonstrates how the alarm is set off before running the code shown in Figure 24 above.



```
warning = multiprocessing.Process(target=alarm)
if led != 0:
    led.terminate()
    led.join()
    whiteOff()
#call the buzzer here
warning.start()
#record video and move it to storage device
```

Figure 25

Here it also shows that the warning LED usage is terminated and reset to be used again when motion is no longer detected. After the video is taken, the alarm will be terminated and the warning LED will turn on to alert the user that motion has been detected recently. That is when the system will also go back to take pictures until motion has been detected again.

Overall System Development Progress

In regards to other updates, some of the requirements were removed. These were ones that included requiring the use and implementation of an infrared LED as the camera purchased supplemented this requirement. This made those requirements redundant, so they were removed. If they are needed to be added back, then there is a backup requirements document that has them, but that will be unlikely.

There was also a sub requirement removed, which was talking about how the decibel count is still to be determined until the appropriate sound range has been determined. This was removed because it was difficult to determine how high the decibel count was during testing, so the only part for the piezo now is if its sound range can reach up to a certain distance. The distance tested was to see if it could be heard from about 10 feet away at minimum from another room, and it could. The change to the requirement was then made so that this would be more of what was intended to be gained from doing this. This allows for the user to hear it go off without it possibly disturbing those further away. It is also intended to be loud enough to startle a possible intruder, which is what was intended from the start.

In regards to Spring term, what was worked on was the housing. There was a different material chosen that was more expensive than the original material, but it was easier to work with in terms of making a smaller housing unit for the system. The material chosen was abs plastic as it is a softer plastic that was easier to cut and make the needed openings in the middle for the camera and the sensors to be in. The housing was the only part worked on as documentation needed more focus for Spring term. The testing design is one of these documents and can be referred to in *Appendix IV* and *Appendix V* for testing situations, with some hypothetical ones. As for progressing, a memo template is shown in *Appendix VI* to show the progression of the project, mostly during the Winter term of 2021. Then for completing the requirements, overall, about 75% of them were completed. This is less than what was expected, but due to many different situations that impeded progression of the project, this amount that was completed is good. Overall, the progress of the project was somewhat successful and not a failure; however, it does leave room for improvement.

Lessons Learned

When taking on this project, there were many aspects that were learned throughout the process. Organizational progress was made in terms of what kinds of tools were needed to organize a schedule, help plan out what kinds of materials were needed for the project, and also money management. The costs ended up being more than double the amount of what was expected. This was due to trying to find the proper parts that were needed and then there was not any planning for buying spare parts, such as an extra camera or an extra development board. This is due to how expensive they were, so in the future, extra parts will be accounted for in the budget, even if they are expensive. It is that or plan to purchase a cheaper part that would have the same functions needed. Another aspect that was learned was to accommodate for more time with some aspects of the project and not expect to complete everything on time due to delays. The situations that the project was developed in were not the best as they were unpredictable. That does not mean the project was not doable as it was. The work environment sometimes was difficult at times and not as flexible as it was needed; therefore, in the future, less time should be accommodated as given to plan for some unpredictability where the project might not progress as much as needed. Then there will also be times where more time will be accommodated to give some aspects of the project more time to be worked on.

Conclusion

Overall, the biggest takeaway from learning from working on Senior Project is to expect unpredictability and be more flexible in situations where they are not in one's favor. In the future, this knowledge will be applied to further projects. This project was a good learning experience and it was a good experience to learn how projects in the work environment work.

Requirements

1) The device shall have a camera on it.

2) The resolution of the camera shall be a minimum of 600 by 900 megapixels for both pictures and videos.

3) The device shall take a picture every 30 seconds.

4) The pictures shall be taken in .jpg format.

5) The device shall be able to record videos.

6) Videos shall be recorded when the sensor on the device detects movement.

  -These videos shall be taken in a 30 second interval.

  -Another video shall be taken after the first video is done recording, but only if the sensor has gone off again when the first video is taken.

7) The videos shall be recorded in .mp4 format.

8) The device shall allow for only five videos to be recorded within six minutes.

9) The device storage shall contain both videos and pictures only.

  -Any other file type shall be erased automatically.

10) The device shall be able to receive the pictures and videos taken by the camera and then send them to the USB drive for storage.

11) The storage shall be a 2.0 USB drive.

  -It shall hold up to 32 gigabytes of memory.

12) The device shall only allow up to 30 gigabytes of the device's storage for photos and videos to be used.

  -It shall only take up to this amount to take into account how the USB drive will not allow the user to make use of all of the memory given.

13) When the device is turned on, it shall start taking videos or photos after thirty seconds.

  -It shall stop taking pictures when it begins to record videos.

  -It shall be expected that it will take photos first, unless movement was detected within the 30 seconds of being turned on.

14) The device shall be able to store up to 2 months worth of only pictures.

-The number of pictures it shall be able to store up to will be 28,927,182 pictures.

15) The device shall be able to store up to 32,000 30 second videos, but of just videos alone.

16) If both videos and pictures are taken into account, then the device shall be able to store up to, on average, 10,716 30 second videos and 19,284,788 pictures.

17) The user shall be allowed to remove the USB drive to access the files stored on it.

-The microcontroller shall meet the requirements to successfully remove the USB drive without corrupting the data.

18) The system shall be able to know when the USB drive is removed from the device in order to stop taking pictures until the USB is put back into the device again.

-The system shall interrupt all functionality until the USB drive has been detected again.

19) The device shall be able to detect an outside USB storage device, so the information it receives from the camera will not be stored onto the microcontroller.

20) The device shall only accept a USB drive that has 32 gigabytes of memory.

-If the USB is under 32 gigabytes, then the device shall give an audible warning that will last for three seconds, and the flash drive shall not be accepted by the system.

-If the USB is more than 32 gigabytes, it will be treated as if it were a 32 gigabyte flash drive.

21) The device shall wipe the pictures and videos starting at the oldest date once the device meets its storage limit.

-Videos shall be wiped first.

-Pictures shall then be wiped second.

22) The device shall have a sensor that detects movement.

-The movement will need to be within 10 feet of the sensor in order to activate.

23) The device shall have a real time clock.

24) The board shall have a real-time module that will act as the clock.

25) The device shall save the time information on the pictures and videos to tell the user when they were taken.

26) The device shall notify the user if the sensor has been tripped via an RGB LED.

27) The device shall have an RGB LED that is located on the outside of the device so it is visible to the user.

28) The light color of the LED on the device shall indicate the time frame as to when the sensor was last tripped.

-Red shall indicate that it was one minute ago.

-Yellow shall indicate that it was less than five minutes ago, but more than one minute.

- Cyan shall indicate that it was less than ten minutes ago, but more than five minutes.

-Green shall indicate that it was less than twenty minutes ago, but more than ten minutes.

-Blue shall indicate that it was less than thirty minutes ago, but more than twenty minutes.

-Magenta shall indicate that it was less than forty-five minutes ago, but more than thirty minutes.

-White shall indicate that it was forty-five minutes ago or more.

-LED shall turn off after 24 hours or until the next instance of motion has been detected.

29) The device shall be hooked up to an electrical outlet so it can be constantly running without the need to change or recharge a battery.

-The microcontroller shall be required to be able to have a port that allows for the device to be connected to an electrical outlet.

-The device shall have a plug to be able to connect to an AC power outlet.

-The device shall be able to remain on while it is plugged in.

30) The device shall indicate a thirty second timer when it is initially turned on.

31) The timer shall be an audible device that will beep once a second until thirty seconds have passed.

-This shall be used to alert the user to move away from the device to not set it off themselves.

32) The device shall have an audible sound made from a small part connected to the microcontroller when the sensor is tripped.

-The sound shall last for 30 seconds after the sensor has been tripped.

-The sound shall be heard from 10 feet away from another room.

33) The device shall have a microcontroller that's appropriately selected to properly maintain the device's functionality.

-The device that has been chosen is a Raspberry Pi 4 B.

34) The memory of the functions shall be stored via flash.

35) The device's microcontroller shall be selected to have a sufficient amount of I/O ports.

36) The device shall be programmed via Raspberry Pi 4 IDE and Python.

37) The system shall be encased in housing and be designed to be attached to a wall.

38) The user shall not be allowed to modify the following:

-The sensor.

-The camera.

-The power cables.

-The device that makes an audible sound.

-The microcontroller that powers the device.

-The housing itself.

39) The user shall be able to modify the following:

-The USB drive to be able to remove it to access the files stored on it.

Glossary

Package Deal - a term to define how a product has more additions, but are not listed in the full price because the user pays for them separately.

Add-ons - additions made to a product that the user can choose to purchase as they are not in the base product.

RGB - red, green, and blue is what it is short for. This refers to a system for representing the colors, usually to be used on a computer display.

FPGA - short for field-programmable gate array, it is an integrated circuit designed to be configured by the user [11].

IDE - short for integrated development environment, it is a software application that provides comprehensive facilities to computer programmers for software development [12].

Appendix

```
#!/usr/bin/python

import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD) #Set GPIO to pin numbering
pir = 8 #Assign pin 8 to PIR
led = 10 #Assign pin 10 to LED
GPIO.setup(pir, GPIO.IN) #Setup GPIO pin PIR as input
GPIO.setup(led, GPIO.OUT) #Setup GPIO pin for LED as output
print ("Sensor initializing . . .")
time.sleep(2) #Give sensor time to startup
print ("Active")
print ("Press Ctrl+c to end program")

try:
	while True:
		if GPIO.input(pir) == True: #If PIR pin goes high, motion is detected
			print ("Motion Detected!")
			GPIO.output(led, True) #Turn on LED
			time.sleep(4) #Keep LED on for 4 seconds
			GPIO.output(led, False) #Turn off LED
			time.sleep(0.1)

except KeyboardInterrupt: #Ctrl+c
	pass #Do nothing, continue to finally

finally:
	GPIO.output(led, False) #Turn off LED in case left on
	GPIO.cleanup() #reset all GPIO
	print ("Program ended")
```

Appendix I

```
import time, sys
import RPi.GPIO as GPIO

redPin = 11   #Set to appropriate GPIO
greenPin = 15 #Should be set in the
bluePin = 13  #GPIO.BOARD format

def blink(pin):
        GPIO.setmode(GPIO.BOARD)

        GPIO.setup(pin, GPIO.OUT)
        GPIO.output(pin, GPIO.HIGH)

def turnOff(pin):
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(pin, GPIO.OUT)
        GPIO.output(pin, GPIO.LOW)

def redOn():
        blink(redPin)

def redOff():
        turnOff(redPin)

def greenOn():
        blink(greenPin)

def greenOff():
        turnOff(greenPin)

def blueOn():
        blink(bluePin)

def blueOff():
        turnOff(bluePin)

def yellowOn():
        blink(redPin)
        blink(greenPin)
```

```
def yellowOff():
        turnOff(redPin)
        turnOff(greenPin)

def cyanOn():
        blink(greenPin)
        blink(bluePin)

def cyanOff():
        turnOff(greenPin)
        turnOff(bluePin)

def magentaOn():
        blink(redPin)
        blink(bluePin)

def magentaOff():
        turnOff(redPin)
        turnOff(bluePin)

def whiteOn():
        blink(redPin)
        blink(greenPin)
        blink(bluePin)

def whiteOff():
        turnOff(redPin)
        turnOff(greenPin)
        turnOff(bluePin)

def main():
        while True:
        redOn()
        time.sleep(1)
        redOff()
        time.sleep(1)
        greenOn()
        time.sleep(1)
        greenOff()
        time.sleep(1)
        blueOn()
        time.sleep(1)
        blueOff()
        time.sleep(1)
        yellowOn()
        time.sleep(1)
        yellowOff()
        time.sleep(1)
        cyanOn()
        time.sleep(1)
        cyanOff()
        time.sleep(1)
        magentaOn()
        time.sleep(1)
        magentaOff()
        time.sleep(1)
        whiteOn()
        time.sleep(1)
        whiteOff()
        time.sleep(1)
        return

main()
```

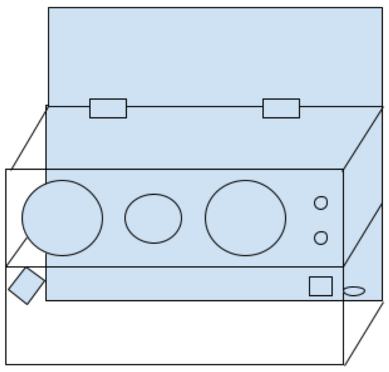
Appendix II

Box Dimensions
Length: 2 ⅞”
Width: 1 ¾”
Height: 3”

Camera Length is 2 ½”, but it is closer to the right and is ⅛” away from the edge. Big holes are ½” and the smaller one is 7/16”. They are spaced 7/16” apart.

USB:
¼” by ¾”

⅜” up from the base that starts from ¼” to ⅜”. Is 1/16” from the box edge.



Motion Sensor is 11/16” by 11/16”. It starts 11/16” above the base. LED starts 1 ¾” from the base directly above the motion sensor, and is ⅛” in diameter. The piezo buzzer is 2 ½” above the base and has a diameter of 7/16”. These are all ⅛” away from the edge and camera.

There is a small port on the backside right for the AC adapter. It is about ⅛” by ¼” and is 3/16” away from the edge.

Appendix III

Mary Swords - Test Plan and Results.xlsx

Appendix IV

Mary Swords - Proposed Test Plan Overview

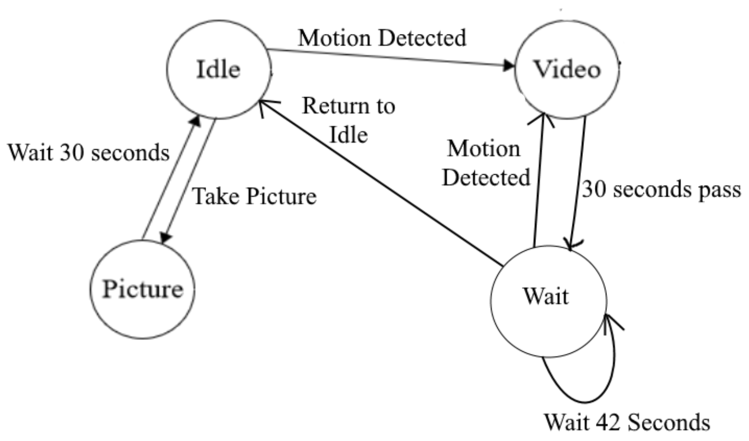
Appendix V

Memo

To:Kevin Pintong
From:Mary Swords
Date:2/5/2021
Re:Week 5 Memo

One of the items that has been completed for the deliverables is the first deliverable has been completed. This includes that the camera is able to take 30 second videos, and knows when to stop taking videos and take pictures instead. It also means that it will know if the sensor has been tripped during the recording that the current recording will not be interrupted. This is demonstrated with the state machine diagram below.

State Machine Diagram



What is also demonstrated in the state machine diagram is for the requirement to not allow more than five videos to be taken within 60 minutes. The math is as follows.

Math

$$60 \times 6 = 360 \text{ seconds}$$

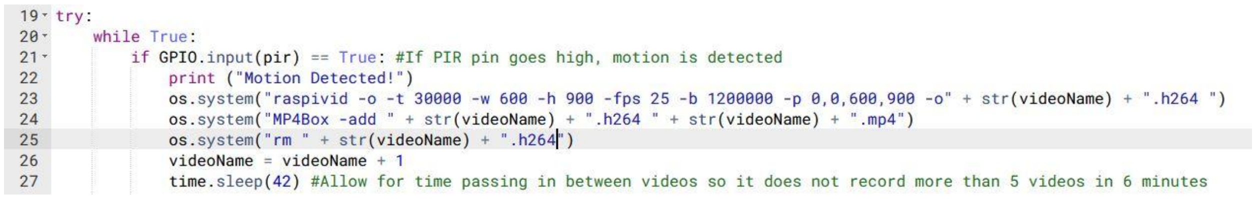
$$5 \times 30 = 150 \text{ seconds}$$

$$360 - 150 = 210 \text{ seconds}$$

$$210 \div 5 = 42 \text{ seconds}$$

Then the last item that was completed was to have the videos be saved as a .mp4 file. This was done by installing the correct drivers onto the Raspberry Pi, and then changing the input line of code.

Code Example



```
try:
    while True:
        if GPIO.input(pir) == True: #If PIR pin goes high, motion is detected
            print ("Motion Detected!")
            os.system("raspivid -o -t 30000 -w 600 -h 900 -fps 25 -b 1200000 -p 0,0,600,900 -o" + str(videoName) + ".h264 ")
            os.system("MP4Box -add " + str(videoName) + ".h264 " + str(videoName) + ".mp4")
            os.system("rm " + str(videoName) + ".h264")
            videoName = videoName + 1
            time.sleep(42) #Allow for time passing in between videos so it does not record more than 5 videos in 6 minutes
```

With all of those items completed, the last item that is currently being worked on is the piezo buzzer. There are still issues with determining the correct frequency for it, so there could also be a potential design change for it. It is not for certain, so an update for this will be given by the end of Week 6. Once the piezo is properly integrated, then it will be added to the code where it will be used alongside the motion sensor.

Appendix VI

References

[1]“Security alarm,” *Wikipedia*, 05-Jun-2020. [Online]. Available: https://en.wikipedia.org/wiki/Security_alarm. [Accessed: 06-Jun-2020].

[2]R. Edwards, "Crime and the Coronavirus: What You Need to Know | SafeWise", *SafeWise*, 2020. [Online]. Available: https://www.safewise.com/blog/covid-19-crimes/. [Accessed: 08- Jun-2020].

[3]“Best Home Security Systems of 2020,” *U.S. News & World Report*. [Online]. Available: https://www.usnews.com/360-reviews/home-security. [Accessed: 06-Jun-2020].

[4]“SimpliSafe: Home Security Systems,” *United States*. [Online]. Available: https://simplisafe.com/. [Accessed: 06-Jun-2020].

[5]“DIY Home Security Systems,” *Frontpoint*. [Online]. Available: https://www.frontpointsecurity.com/. [Accessed: 06-Jun-2020].

[6]“Wyze Cam V2,” *Wyze Cam | 1080p HD Smart Home Camera With Free AWS Cloud*. [Online]. Available: https://wyze.com/wyze-cam.html. [Accessed: 08-Jun-2020].

[7]Vizza, L., 2020. *What Google And Adobe Can Teach Us About Feature Creep*. [online] Toptal Design Blog. Available at: <https://www.toptal.com/designers/ux/teach-us-about-feature-creep> [Accessed 6 June 2020].

[8]B. Zaidi, “Why you should avoid most add-ons, optional features of term insurance plans,” *The Economic Times*, 12-Jun-2017. [Online]. Available: https://economictimes.indiatimes.com/wealth/insure/why-you-should-avoid-most-add-ons-optional-features-of-term-insurance-plans/articleshow/59081216.cms?from=mdr. [Accessed: 06-Jun-2020].

[9]L. Ada, “Adding a Real Time Clock to Raspberry Pi,” *Adafruit Learning System*, 2012. [Online]. Available: https://learn.adafruit.com/adding-a-real-time-clock-to-raspberry-pi. [Accessed: 04-Dec-2020].

[10]*Night camera Raspberry pi*. 2020.

[11]“Field-programmable gate array,” *Wikipedia*, 06-Jun-2020. [Online]. Available: https://en.wikipedia.org/wiki/Field-programmable_gate_array. [Accessed: 08-Jun-2020].

[12]“Integrated development environment,” *Wikipedia*, 23-May-2020. [Online]. Available: https://en.wikipedia.org/wiki/Integrated_development_environment. [Accessed: 08-Jun-2020].