Interactive Mirror Display

By

Hiraal Doshi

Stephen Hurwit

Pamela Sanan

Final Report for ECE 445, Senior Design, Spring 2019

TA: Nicholas Ratajczyk

1 May 2019

Project No. 29

Abstract

The ability to interact with devices on a regular basis allows users to perform a large amount of tasks ranging from scheduling to streaming a video. While smart devices facilitate the completion of simple tasks, such as scheduling, each device would increase the amount clutter around the house. Smart devices around a consumer's home would facilitate the management of an increased amount of devices, but current devices on the market have limited functionality.

The Interactive Mirror Display provides convenient access to information and media through a discrete gesture and speech controlled device. This device makes use of passive infrared sensors and proximity sensors to detect gestures from the users, and speech recognition software in order to recognize specific commands. The sensors communicate with a microcontroller which forwards data to the Raspberry Pi. This is then used to determine which action the user wishes to perform. The speech commands are received through a microphone on the Raspberry Pi and are similarly used to perform actions by the system. The actions then reflect the appropriate response to the user interface. Through the implementation of these aspects, the Interactive Mirror Display provides a unique device with increased interactivity. Given that a mirror is a common household item, the device is a solution that does not add to the complexity of the user's daily activities.

Contents

1 Introduction

1.1 Objective

Technology has become a ubiquitous presence in our lives. We have many devices which take up space and clutter up our living environments, providing us with convenient access to interesting features and functionalities in exchange. However, it can be difficult to manage and find space for so many devices, especially those that present information visually.

We propose to address this issue by developing an interactive mirror display which provides context sensitive information and media access to the user through a discreet and unobtrusive device. We intend to integrate gesture recognition, through the use of visual sensors, and voice commands, through the use of Alexa integration, which will allow the user to interact with the device and access further information. The sensors will interface with a microcontroller, the user interface software will be run on a Raspberry Pi, and we will integrate a camera, microphone, and speakers.

1.2 Background

As technology is progressing, we are slowly entering a futuristic age. Smart devices are gaining popularity and are a part of peoples daily lifestyles and households. There is a clear interest from the public in these devices (51% of consumers in the United States are most excited about smart home technology), but many have concerns about cost, privacy and potential clutter with the accumulation of these devices [1]. We would like to design a smart mirror that can address some of these concerns. The appeal of a mirror is that it is already an everyday household object; it is discrete and concealed as a smart device. Furthermore, we want to be able to encompass many technologies to reduce clutter, so the mirror will be capable of presenting visual and audio content to the user directly from the same device. It will offer voice integration and gesture recognition in order to facilitate intuitive control of the device.

There are several other smart mirrors[2] with voice integration on the market today, however they are often little more than large Android devices hidden behind a mirror that do not effectively take advantage of their mirror form factor. As a reach goal, we hope to make our mirror more innovative by using computer vision and machine learning to integrate the capability to detect certain visual features of the users appearance. Based on these detections, the mirror can suggest commercial products that might be of interest to the user. As for privacy concerns, the camera and microphone on the mirror will be under the complete control of the user through the settings and the use of a simple off switch. Overall, we want to implement a smart mirror that is effective, user-friendly and secure.

1.3 High-Level Requirements

- The mirror should be able to recognize gestures, including left, right, up, and down swipes and hovering a hand over one of three proximity sensors, and perform the appropriate corresponding action.
- The mirror should be able to recognize speech and interpret specific voice commands. We will specifically be integrating a Google speech recognition library that will listen for specific hotwords.
- The mirror should have a functional and practical display, which adjusts accordingly in order to ensure the information is clearly visible.

2 Design

The overall design of the device is split into submodules which each provide a specific functionality. The user interface module provides the hardware required to effectively communicate information to the user, including a display and status LEDs. The audio I/O module enables the user to interact with the device via voice commands through a microphone and provides speakers through which the device can respond and provide audio media. The visual sensor module makes use of a camera, passive infrared (PIR) sensor array, and proximity sensors to enable gesture detection and perform visual analysis on the user, as well as typical photo and video functionality. The control unit consists of a microprocessor which will process input from the sensor modules and relay information to the Raspberry Pi which runs the virtual user interface software. The power module ensures that the devices in the rest of the modules receive the power that they need to operate. The composition of and interaction between these modules is presented visually in Figure 1.

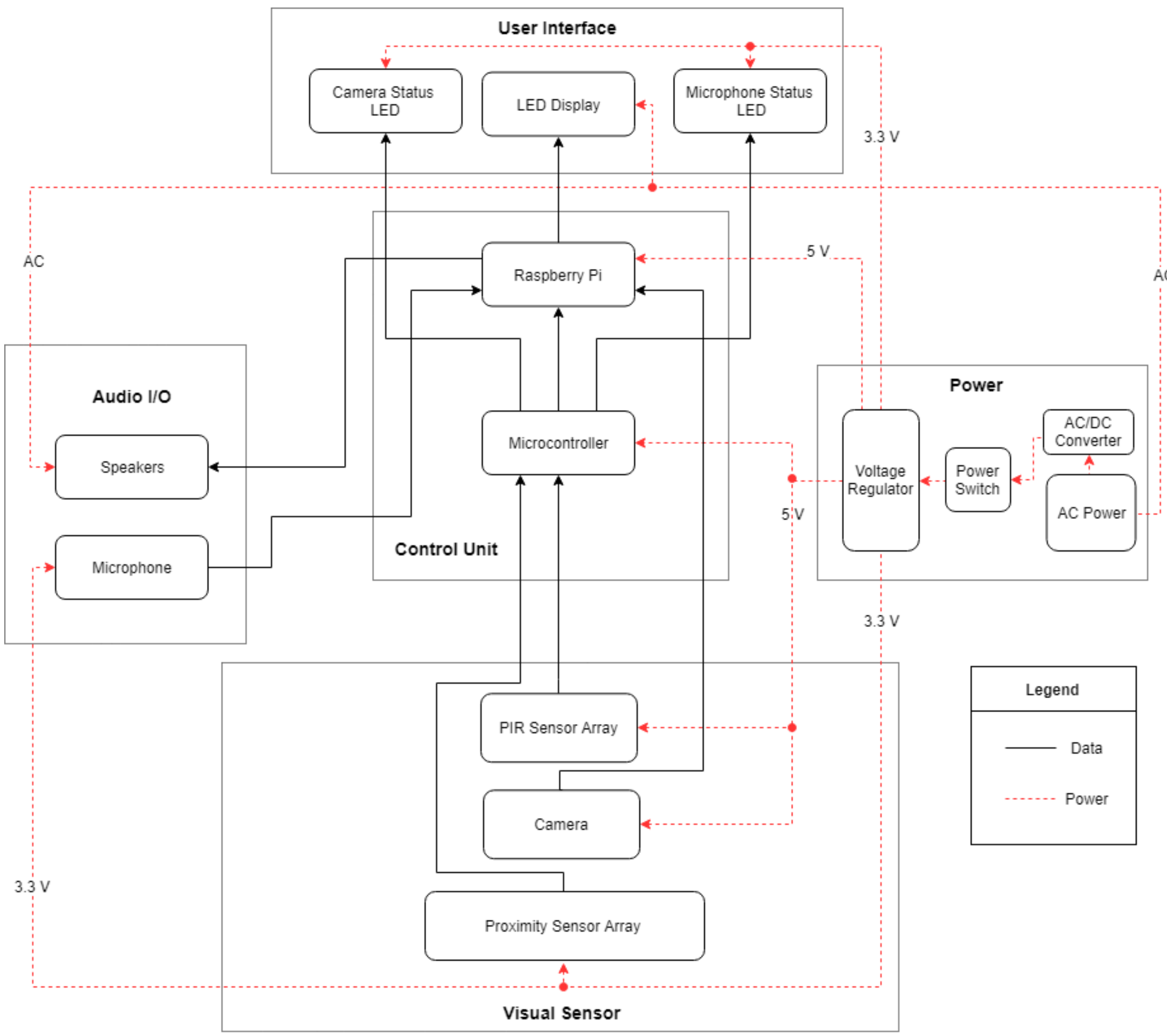

Figure 1: Interactive Mirror Display Block Diagram

The physical design of the device consists of a single self-contained unit. The front consists of a two-way mirror surrounded by a bezel at the bottom of which the two status LEDs is inlaid. The back of the device consists of a monitor which fills the upper portion of the display and a sensor module below which includes the camera, PIR sensors, proximity sensors, and microphone. By placing the sensor module at the bottom of the device, we ensure that they have an unobstructed view through the two-way mirror while still being concealed effectively. The bottom of the frame contains holes to ensure that sound travels to the microphone and from the speakers effectively as well as the power switch. The physical layout and dimensions of the device are presented visually in Figure 2.

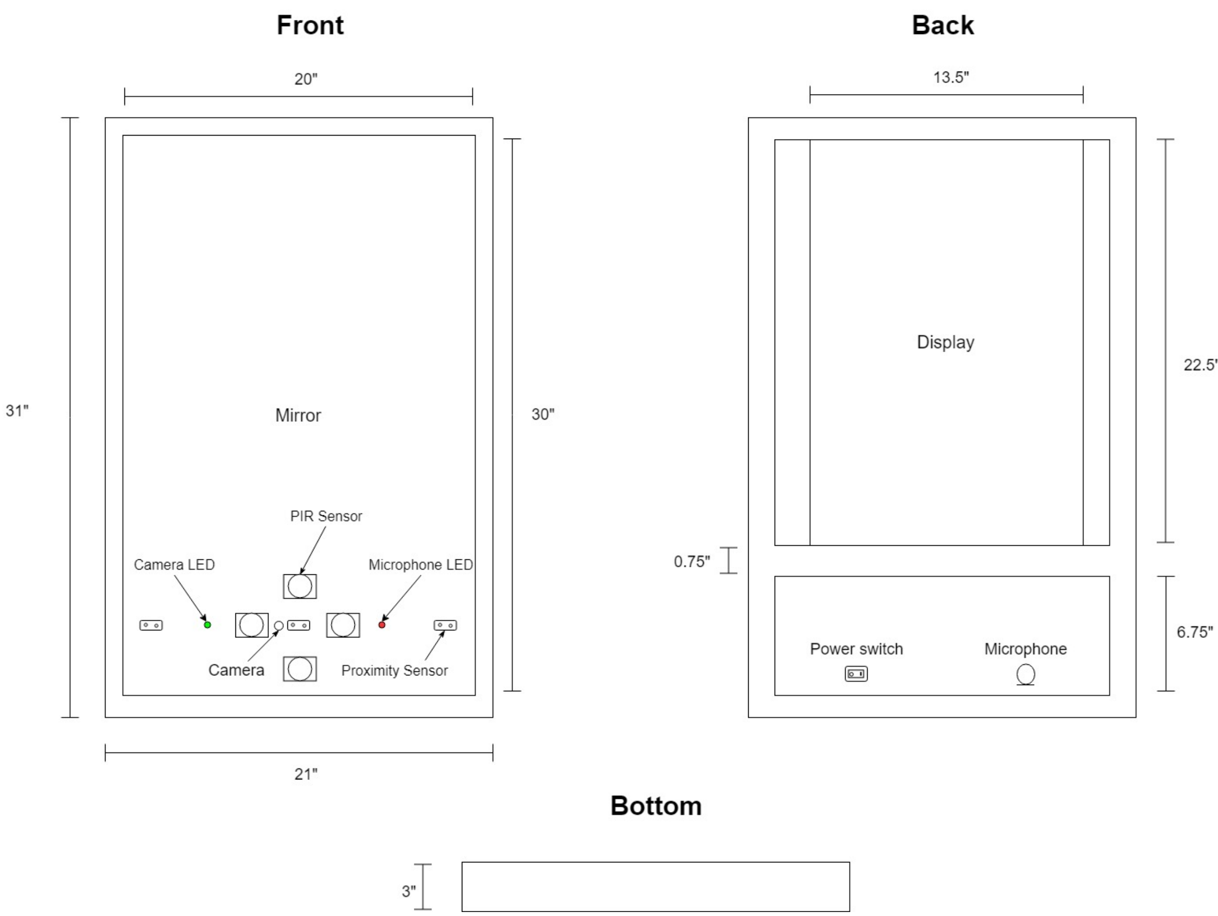

Figure 2: Interactive Mirror Display Physical Diagram

2.1 Control Unit

The control unit contains a microcontroller and a Raspberry Pi. The microcontroller is responsible for coordinating the data flow between all of the sensors and outputs. It processes the data it receives and communicates with the Raspberry Pi over UART, which is responsible for providing the virtual user interface.

2.1.1 Microcontroller

The microcontroller (ATmega328p [3]) receives sensor data from the PIR and proximity sensors and interpret it as specific gestures including swipes and palm holds. It sends the gesture data to the Raspberry Pi over the serial port. It is also responsible for activating the status LEDs when the camera or microphone is in use.

Figure 3 shows the pinout diagram for the ATmega328p. We made use of the serial communication pins, I^2C pins, and analog pins of this device to communicate with the sensors and the Raspberry Pi.

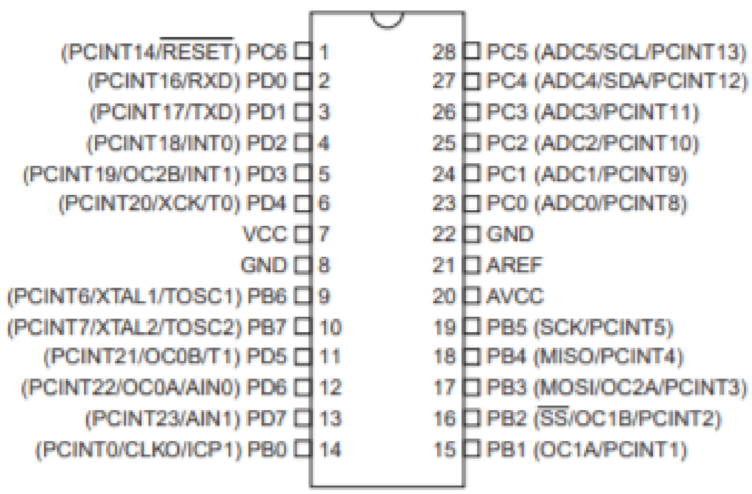

Figure 3: ATmega328P Pinout [4]

2.1.2 Raspberry Pi

The Raspberry Pi 3 [5] is responsible for running the virtual user interface which provides visual and audio media to the user through the Raspberry Pi's wireless network chip. It runs an application on the platform Electron which provides the full-featured user interface and receives inputs from the microcontroller which communicate processed sensor data and controls.

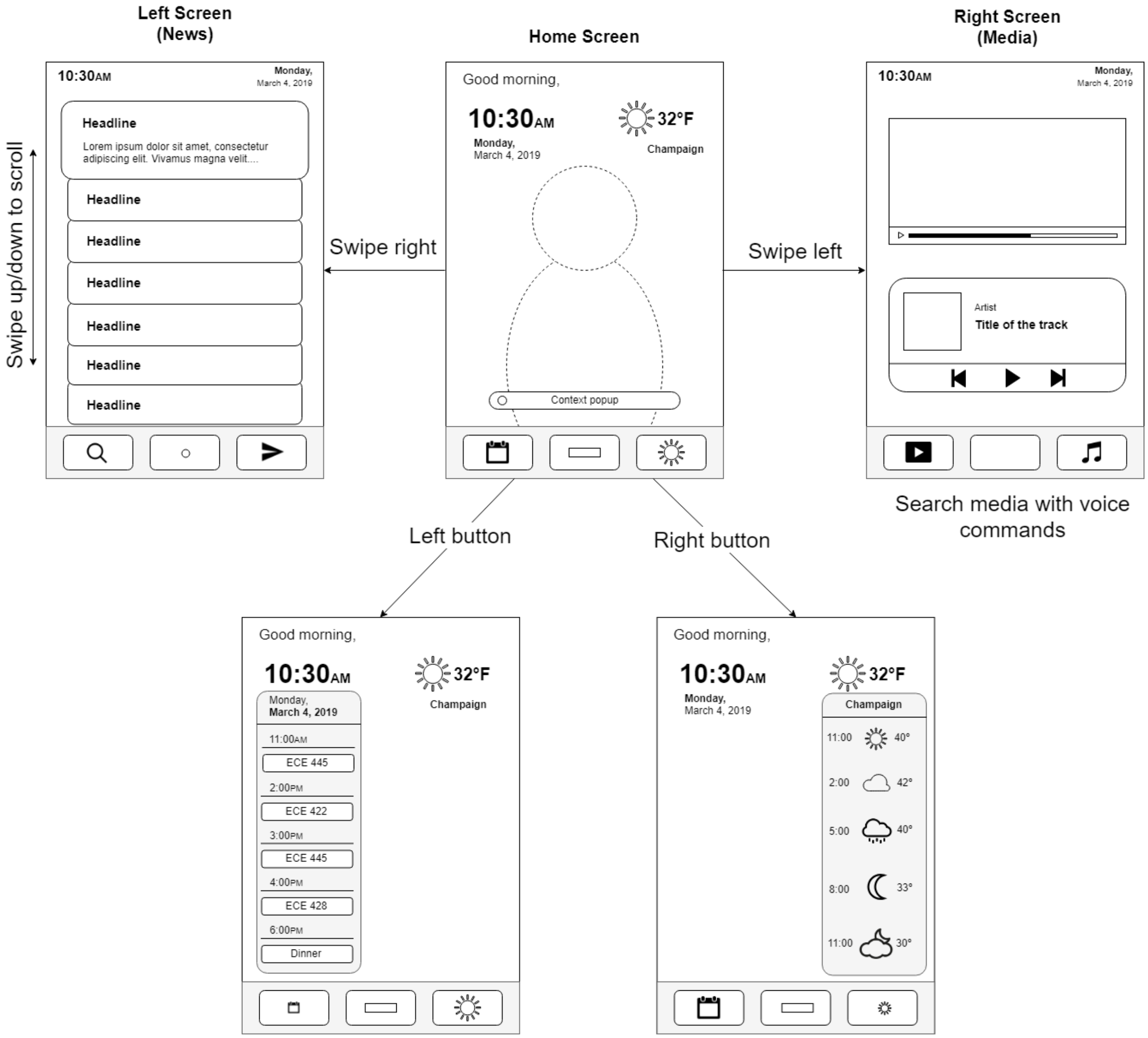

Figure 4: User Interface Diagram

Figure 4 contains a prototype of the user interface application which runs on the Raspberry Pi and is displayed on the monitor. It demonstrates some of the content that the device will provide and illustrates how the gestures will be used to navigate between screens and contexts. Each of the buttons contained within the overlay at the bottom of the screen will be activated by the corresponding proximity sensor, and the swipe gestures will be recognized by the PIR sensor array.

2.2 User Interface

The user interface supplies the hardware necessary to communicate information to the user effectively. The virtual user interface is presented on the display monitor and two status LEDs notify the user when the camera or microphone are active. The user interface was written in JavaScript, HTML and css (Node.js).

2.2.1 Display Monitor

The display monitor is the main interface through which information and content will be provided to the user. We used a 1080p monitor to provide a clear picture at a common resolution with a reasonable cost. Data is supplied to the display by the Raspberry Pi over HDMI, and power is supplied through the wall outlet.

2.2.2 Camera Status LED

The camera status LED indicates when the camera is currently enabled so that the user is always aware that it is watching. Additionally, a mechanical switch is incorporated to control whether or not the camera is being utilized. The camera status LED is powered on by a GPIO pin on the microcontroller.

2.2.3 Microphone Status LED

The microphone status LED indicates when the microphone is currently enabled so that the user is always aware that it is listening. The microphone status LED is powered on by a GPIO pin on the microcontroller.

2.3 Audio I/O

The audio module takes input through the microphone and output through the speakers in the monitor in order to provide voice control and audio media playback capabilities. Originally, the design incorporated an Alexa SDK in order to listen to voice commands. However, the Amazon Alexa library is not very flexible with adding new voice commands that would be more suitable for the mirror. Therefore, the new design incorporates a Google Speech Recognition library so the voice commands can be customized and be integrated with the mirror in a user-friendly way.

2.3.1 Microphone

The microphone (Mini USB Microphone [6]) is used to collect data as the user speaks in order to process voice commands. It sends the data to the Raspberry Pi over UART and is powered through the Raspberry Pi.

2.3.2 Speakers

The speakers output audio to the user including video and music audio. Data is provided by the Raspberry Pi over HDMI.

2.4 Visual Sensor

The visual sensor system provides various visual information to the processor in order to facilitate gesture recognition. These gestures include swiping left, swiping right, swiping up, swiping down, and holding your hand in front of a sensor for a period of time.

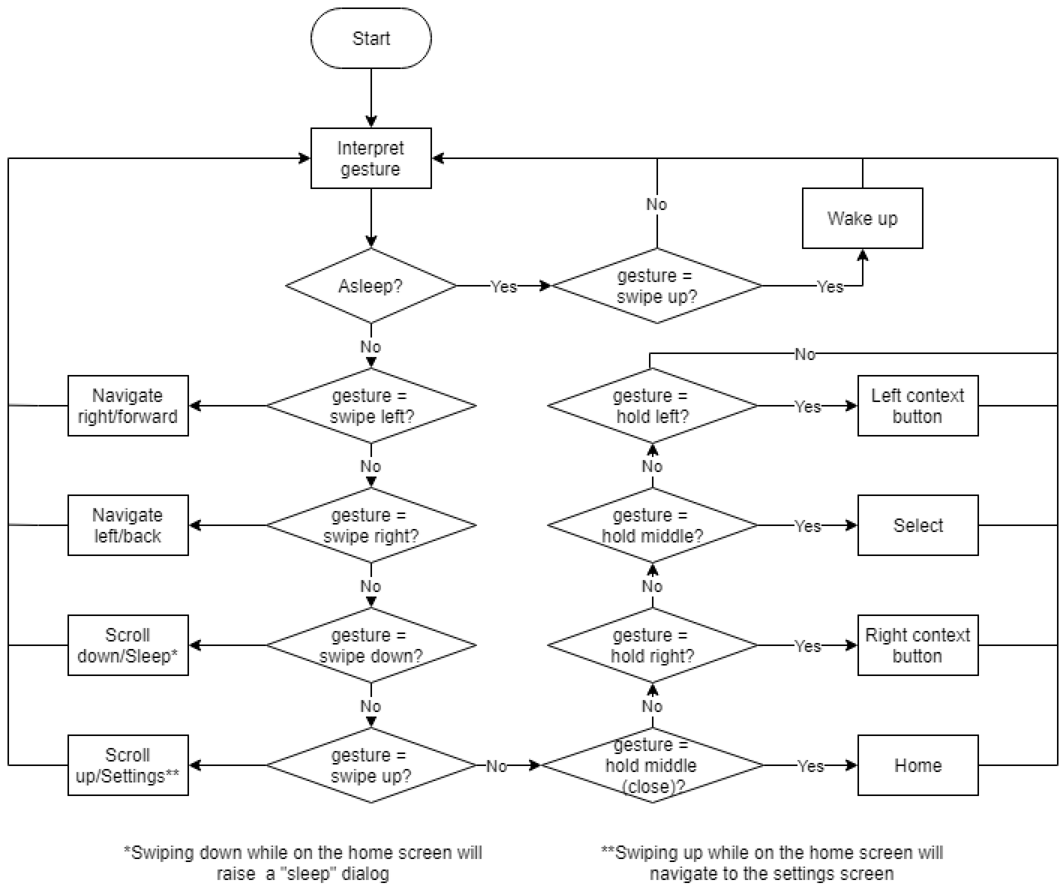

Figure 5: Gesture Control Algorithm

The flow chart in Figure 5 outlines the gesture recognition system that will be implemented for the Interactive Mirror Display. The mirror reads a gesture, then performs a task accordingly. If the mirror is asleep, then it will turn on if the gesture is recognized as an upward motion. If the mirror was not asleep in the initial state, then it will interpret a sequence of gestures to either navigate or perform certain actions, such as select an item or a button. The flow chart illustrates the state space of the mirror, and which of the eight gestures can be used to transition from a given state to another.

When the mirror is on and the user wishes to transition it to sleep mode, the user must perform a gesture swipe and button hold to get to the power down page. Before going into sleep mode, the mirror will ask the user for confirmation to do so to avoid accidentally going to sleep.

2.4.1 Camera

The original design was for the camera (Raspberry Pi Camera Module V2 [7]) to provide a real time stream of visual information to the microprocessor which is used to send pictures and videos. The feed is then sent to the Raspberry Pi for processing. The new design had the camera stream be activated by the Raspberry Pi itself therefore eliminating the microcontroller as a "middle man" and avoiding unnecessary delays. In the user interface, in order to turn on the camera, the user can perform a button press to activate it. Once the camera has taken the picture or video, the camera automatically turns off which is indicated by the camera status LED.

2.4.2 PIR Sensor Array

The PIR sensors (HC-SR501 [8]) detect motion within their field of view and output a digital signal accordingly. We use an array of these sensors in order to detect motion, specifically hand gestures such as swipes. There are four PIR sensors in the array in order to determine directionality of the swipe. In order to interpret these gestures, we use the delay between the activation of the sensors in the array to determine the direction of the motion.

2.4.3 Proximity Sensors

The infrared proximity sensors (VCNL4200 [9]) are used to enable selection of the two auxiliary action buttons when the user places their hand within 30cm of the sensor for over 1 second. The distance data is sent to the microcontroller over the I^2C bus. The design includes three of these sensors: one each for a left button press, right button press and a middle buttons press.

Figure 6 shows the wiring diagram for the VCNL4200. The I^2C communication pins are connected to the microcontroller in order to receive distance data when a user places their hand near the proximity sensor.

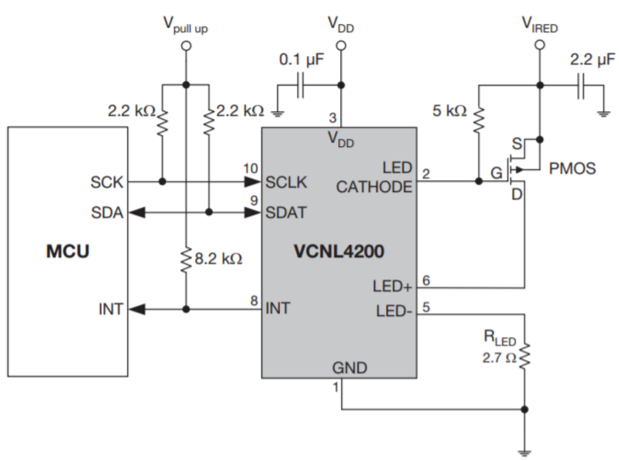

Figure 6: VCNL4200 Pinout [10]

2.5 Power

The power module is responsible for adapting the power provided by the wall outlet and supplying power to all of the other components.

2.5.1 AC Power

AC power is supplied via a wall outlet.

2.5.2 AC/DC Power Converter

The power converter converts the AC power supplied by the wall (120V) to DC power usable by the devices in our design (12V).

2.5.3 Voltage Regulator

The voltage regulator regulates the voltage supplied by the power converter at the specific voltages required for the various devices throughout the design. The proximity sensors require 3.3V and the PIR sensors and microcontroller require 5V. Two surface mount linear regulators were included in the initial design to be placed on the PCB. However, the design was later modified to include two through hole linear regulators after the initial surface regulators were burned during the soldering process.

2.5.4 Power Switch

The power switch turns the device completely on or off. The mirror may also be woken with gestures and/or voice commands, but in order to eliminate privacy concerns and allow the users complete control over what features are enabled, the switch is provided to turn the whole device off when not in use.

3 Verifications

Figure 7 contains a photograph of our completed device with the display active behind the two-way mirror. The sensor array can be seen at the bottom of the mirror with the PIR sensors affixed to the front and the proximity sensors placed behind the mirror looking through holes drilled in the screen.

Figure 7: Photo of Completed Device

3.1 Control Unit

The control unit successfully takes sensor input data and changes the display accordingly. The proximity sensors output data to the microcontroller ATMega328-P over the I^2C bus. Since there are multiple proximity sensors that have the same address, an I^2C MUX was required so that they could share the same I^2C bus. Next, the microcontroller performs an I^2C read to receive distance data from the proximity sensor. The microcontroller also polls the GPIO pins to determine when the PIR sensors are active. Then, the microcontroller sends this data to the Raspberry Pi over UART with a baud rate of 9600. A host program written in Python then transmits messages to the user interface over a TCP connection. Finally, the user interface displayed on the Raspberry Pi changes after receiving the appropriate sensor data.

3.2 User Interface

The user interface works as expected. It was written in Javascript and successfully ran on the Raspberry Pi through the platforms Electron and Node.js. The interface includes a personal schedule, news, weather, music and videos for the user. The news and weather information was pulled from common JavaScript APIs that contain the most relevant news and weather data based on location. For the schedule, we chose to integrate a Google Calendar API since it is the most common personal schedule seen in use. For videos, a Youtube API and widget for JavaScript was implemented that allows users to search for specific videos or playlists. The last API used was SoundCloud that allows users to search for specific playlists using voice commands.

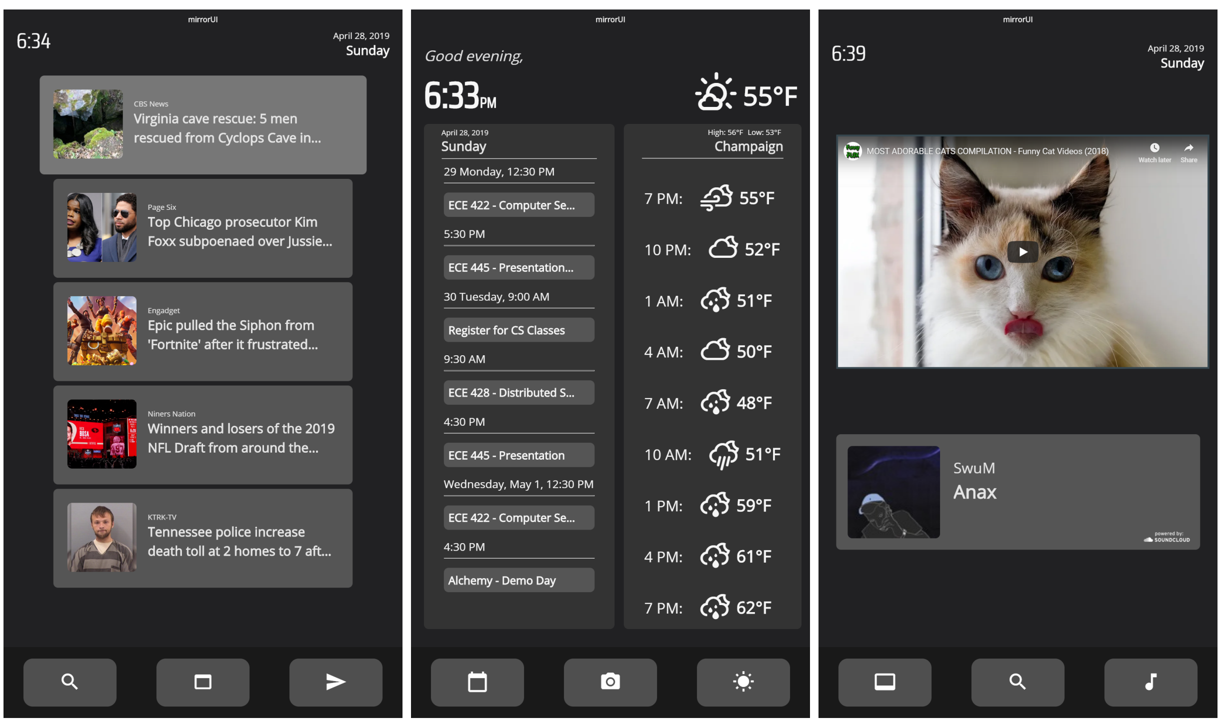

Figure 8: Screenshots of Completed User Interface

Figure 8 contains screenshots of the three main pages in the completed design. The leftmost page contains the current headlines, the center page contains the schedule and weather, and the rightmost page contains the media displays and controls.

In order to search for music or videos, we successfully integrated voice commands into our interface. A package called ”annyang” was installed that uses Google Speech recognition to listen for specific commands. The commands we preset were: ‘play playlist “Taylor Swift”’, ‘play video “cats”’, ‘serach news “Illinois”’, ‘play/pause music’, ‘skip/previous song’.

The first three commands can only be activated if the search button is pressed on the user interface. This means that if search was never pressed, the microphone will never be listening for these commands in the first place. The last 4 music commands are passive commands, meaning the microphone is listening for them at all times. This can be enabled or disabled in settings depending on what the user feels comfortable with.

There are two status LEDs on the bottom of the mirror indicating whether the microphone or camera was enabled. To test if the LED ever lit up at all, we set up a test circuit as shown in figure 9. Both LEDs lit up proving they worked. To test visibility, a DC voltage of 3.3V was provided to each LED in order to make sure it was bright enough to be seen from one meter away. The results were as follows:

Table 1: LED Verification

LED	**Visible at 1m (Y or N)**	**Voltage (V)**
LED	Visible at 1m (Y or N)	Voltage (V)
Red	Y	3.3
Green	Y	3.3

Then, based on the use of microphone or camera, the microcontroller sent out data to the LED either turning it on or off.

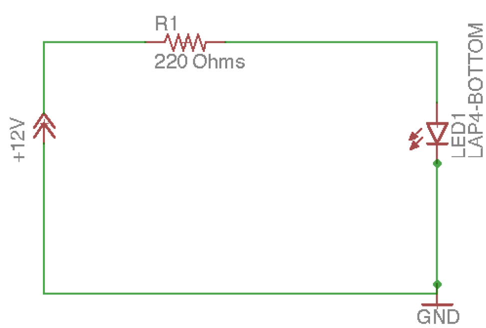

Figure 9: LED Test Circuit - Turns LED on when connected to battery and off otherwise

3.3 Audio I/O

The microphone and speaker were individually tested on the Raspberry Pi to make sure they could function at the human hearing range as needed for our Smart Mirror. All requirements from the table were met. The only issue was the microphone had difficulty picking up sound from far away. The main reason for this is because the microphone was connected to the Raspberry Pi which was hidden behind the mirror. To improve this, we could either buy a more expensive microphone that has a larger range or make a hole in the bottom of the mirror for the microphone to peek out.

3.4 Visual Sensor

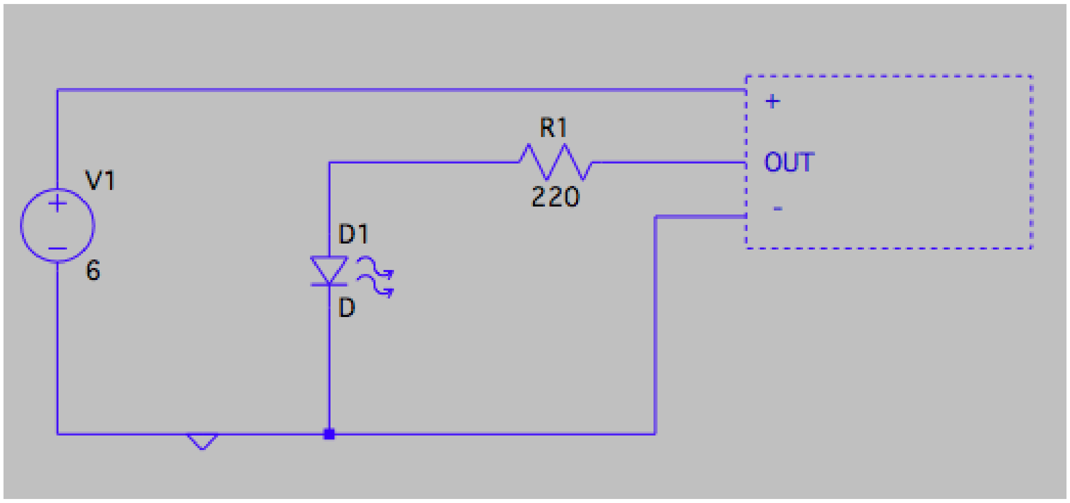

Figure 10: PIR Test Circuit - LED turns on when the sensor detects a hand in range

Each PIR sensor was tested individually using the circuit illustrated above to ensure that each sensor was functioning correctly. They went from a low voltage of approximately 0.01V to a high voltage of approximately 3.24V when an object was placed over a sensor, which represents that the sensor being tested was detecting motion.

The proximity sensors were calibrated by setting the appropriate I^2C command registers to initialize the device. Data was then read from the "Proximity Sensor" command register periodically to retrieve the distance data from the sensor. We experimentally determined what value corresponded to 30cm by measuring with a tape measure and hardcoded this value as a cutoff in the program. We found that this reliably detected when something approached within 30cm of the sensors.

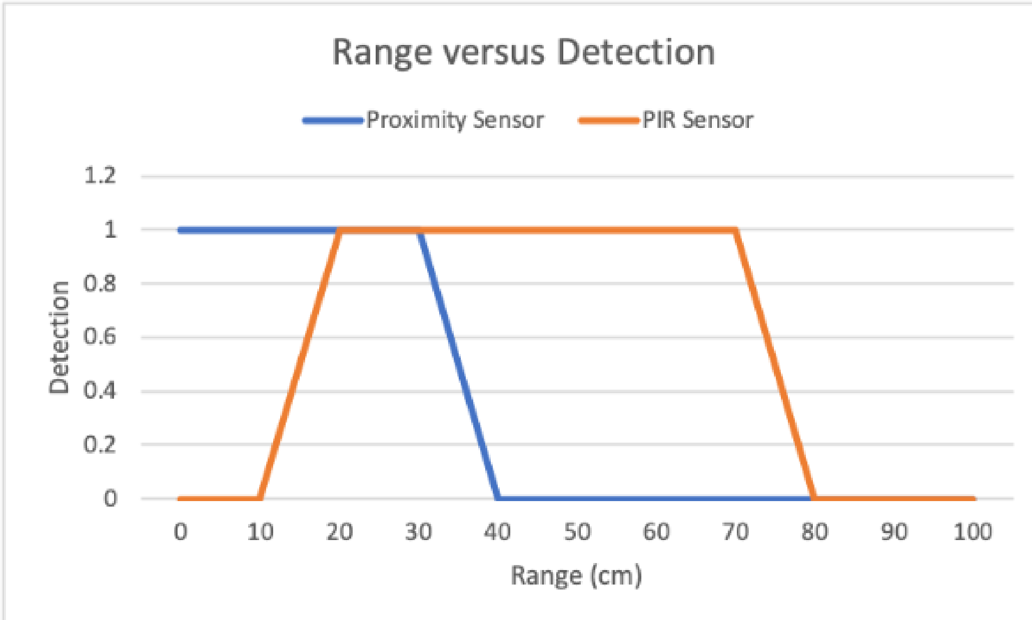

Figure 11: Visual Sensor Detection Range

Figure 11 contains a plot of the detection range of the two sensors that were included in our design after we had modified them appropriately. “Detection” in this case refers to the frequency with which the sensor detects a presence at the specified distance.

Table 2: Visual Sensor Verification Results

Sensor Type	Operating Voltage (V)	Range (cm)
PIR Sensor	5	10 - 80
Proximity Sensor	3.3	0-30

3.5 Power

Our power submodule works as expected. The wall power is properly converted to 5V and 3.3V as needed by our devices. The exact results are shown in the tables below.

Table 3: AC/DC Power Converter Verification Results

AC/DC Converter	V_{in}	V_{out}
	120	12.18

Table 4: Mechanical Switch Verification Results

Mechanical Switch	V_{in}	V_{out}
Off	12.18	12.18
On	12.18	0.01

Table 5: Voltage Regulator Verification Results

Voltage Regulator	V_{in}	V_{out}
3.3V	12	3.313
5V	12	5.01

4 Cost and Schedule

4.1 Cost Analysis

4.1.1 Labor

The cost of development will consist of 15 hours of work per week for 3 people at \$40/hour, as shown in Equation 3.

$$3 \cdot \$40/\text{hour} \cdot 15\ \text{hours/week} \cdot 9\ \text{weeks} \cdot 2.5 = \$40,500 \tag{1}$$

4.1.2 Parts

Table 15 illustrates the parts which will be required to complete the design and the quantities and costs associated with each. We have provided an estimate of both the retail cost for the prototype that we will build in class this semester, as well as the bulk costs that would be applicable for larger scale production.

Table 6: Parts Costs

Part	Retail Cost	Bulk Purchase Cost
Microcontroller (ATmega328p)	\$1.96	\$1.62
Raspberry Pi 3	\$35	\$35
Display Monitor	\$50	\$50
Camera (Raspberry Pi Camera Module V2)	\$29.95	\$22.50
Microphone (Mini USB Microphone)	\$5.95	\$4.76
Proximity Sensors (VCNL4200) x3	\$8.85 (\$2.95)	\$5.70 (\$1.90)
PIR Sensor (HC-SR501) x5	\$8.99	\$8.99
Assorted components	\$9.30	\$9.00
Total	\$150	\$137.57

5 Conclusion

5.1 Accomplishments

A functional system was completed with slight modifications to the initial design, which were implemented due to unexpected occurrences during the building process. The physical components were integrated and functioned as expected, including the PCB, the proximity sensors, and the majority of the other submodules. The data from the proximity sensors was accurately communicated to the microcontroller through the I^2C bus, which was then used to determine what action the user wanted to perform. The graphical user interface was developed and reflected changes according to the aforementioned input from the proximity sensors. Input from both the sensors and the voice commands detected through the speech recognition software generated accurate responses.

5.2 Uncertainties

The PIR sensors did not function as expected in the completed system because of a delay in the reset time. The delay was initially five seconds, but it was reduced to approximately a single second by shorting a resistor. However, this delay is still relatively large, resulting in delayed interactions with the user. Additionally, the PIR sensors were highly sensitive and detected unintended actions from the user.

5.3 Ethical considerations

The ethical and safety concerns that our project presents relate to data management and privacy. As the ACM Code of Ethics (1.6) [11] explains, we need to ensure that potential consumers know whether or not their personal information will be collected or monitored. To specifically address this issue, we will explicitly state how and when the mirror will be gathering and using information. To work toward the goal of full disclosure, we will incorporate LEDs used to alert the consumer that the mirror and/or camera is on. Additionally, we need to ensure that all data and metadata gathered will be kept confidential in order to comply with the ACM Code of Ethics (1.7) [11]. This means that we will not share data that can identify a consumer with third-parties (only specified personnel will have limited access), unless there is a clear violation of the law. In such a case, we will inform the consumer that their data may be shared with proper authorities.

Another concern we will be addressing is the security of our systems. The Interactive Mirror Display will have the potential ability to gather and monitor information about the consumer, we will be implementing certain security measures. In order to provide the consumer with a way to manage the usage of the mirror, we will be implementing a button to turn the mirror on or off. This provides a measure of physical security, and is simple enough for the general consumer to understand, which complies with the ACM Code of Ethics (2.9) [11]. Furthermore, we will be designing a system that implements common networking security protocols to ensure that all data being sent to a server is secured to a satisfactory level. In the case of a data breach, we will notify all parties affected in the "most expedient time possible and without unreasonable delay" [11]. By implementing and following these standards, we would be adhering to the Illinois Personal Information Protection Act (815 ILCS 530, et seq.) and the Federal Trade Commission Act (15 U.S. Code 41, et seq.) [11].

While our device does not raise many safety concerns, it is possible that damage could be caused by improper mounting of the device, or if the screen were to shatter. In order to reduce the likelihood of these events and mitigate the damage that could be caused if they did occur, we will provide two mounting hooks, one on either side of the device, to ensure stability and support. We will also use acrylic coated in two-way mirror film rather than actual glass in order to avoid shattering.

Another potential concern is that the consumer would come into contact with some of the circuitry. To avoid direct contact with the electrical components, the components were enclosed by the actual acrylic described above and the back of the display.

5.4 Future work

In order to improve upon our visual sensor submodule, we would buy more expensive PIR sensors that are more stable than the cheap ones we used in our project. We would experiment with the spacing and the field of view of these new PIR sensors in order to get that to be more accurate. This way the PIR sensors are more reliable for a gesture recognition system. Another option would be to try to use a completely different type of sensor such as a gesture sensor chip or just an array of proximity sensors in order to detect swipes.

We would also expand our user interface to include more applications on the mirror that the user would find helpful. For example we could include Google Maps and traffic information so the user could get ready in the morning and be able to see how long it would take to get work at the same time. Other applications we would possibly add include checking your email or adding stock market information. Furthermore, we would add more voice commands that one could use to interact with the mirror for ease of use.

Along with expanding our user interface, we would also develop a companion mobile app to go along with our smart mirror. The app would give the user the ability to customize their mirror to their liking and be able to log in to specific applications so they could pull their data.

Another improvement we can have to the mirror is using the camera to do computer vision and machine learning to interpret gestures and button presses instead of using the sensors we have. This would make the mirror be able to detect swipes more accurately. Furthermore, the camera and computer vision could be used to suggest products to the user. For example, if the camera is picking up on rashes on the user that were not there previously, it can recommend certain anti-itch or anti-rash creams.

For the actual hardware construction of the mirror, we would use two-way glass instead of acrylic and film like we currently have so the mirror is not warped and the display is clearly visible no matter the lighting. The reason we went with acrylic and film was because it was much cheaper than two-way glass, but if budget restrictions are lifted in the future, the two-way glass is a much better alternative.

References

[1] "Impact of Smart Home Technology - IoT Innovation," Web page, IoT Innovation, accessed February 2019. [Online]. Available: https://internet-of-things-innovation.com/insights/the-blog/impact-of-smart-home-technology/#.XFtTyxlKiqA

[2] "The 6 Best Raspberry Pi Smart Mirror Projects You Can Make," Web page, MakeUseOf.com, 2018, accessed February 2019. [Online]. Available: https://www.makeuseof.com/tag/6-best-raspberry-pi-smart-mirror-projects-weve-seen-far/

[3] "ATmega328, P Product Description," Web page, DigiKey, accessed February 2019. [Online]. Available: https://www.digikey.com/product-detail/en/microchip-technology/ATMEGA328P-PU/ATMEGA328P-PU-ND/1914589

[4] "ATmega328," Datasheet, Atmel, 2009, accessed February 2019. [Online]. Available: https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf?fbclid=IwAR2WZGy-vAtoW_0tXzsM_vME84chuFD0gCNQRNyZ-wkCdcsYvCdljCg8dhg

[5] "Raspberry Pi 3 Product Description," Web page, Adafruit Industries, accessed February 2019. [Online]. Available: https://www.adafruit.com/product/3055?src=raspberrypi

[6] "Mini USB Microphone Product Description," Web page, Adafruit Industries, accessed February 2019. [Online]. Available: https://www.adafruit.com/product/3367

[7] "Raspberry Pi Camera Module V2 Product Description," Web page, SparkFun Electronics, accessed February 2019. [Online]. Available: https://www.sparkfun.com/products/14028

[8] "HC-SR501 PIR Sensor Product Description," Web page, Amazon.com, Inc, accessed February 2019. [Online]. Available: https://www.amazon.com/DIYmall-HC-SR501-Motion-Infrared-Arduino/dp/B012ZZ4LPM/ref=sr_1_3?s=electronics&ie=UTF8&qid=1550559971&sr=1-3&keywords=pir+sensor

[9] "VCNL4200 Proximity SensorProduct Description," Web page, Mouser Electronics, accessed February 2019. [Online]. Available: https://www.mouser.com/ProductDetail/Vishay-Semiconductors/VCNL4200?qs=gt1LBUVyoHnG9%252beWUQ9nRg%3D%3D

[10] "VCNL4200," Datasheet, Vishnay Semiconductors, 2017, accessed February 2019. [Online]. Available: https://www.vishay.com/docs/84430/vcnl4200.pdf

[11] "ACM Code of Ethics," Web page, ACM, 2018, accessed February 2019. [Online]. Available: http://www.linxtechnologies.com/products/rf-modules/lt-series-transceiver-modules/

Appendix A Requirement and Verification Tables

Table 7: Microcontroller Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Operating voltage of 5 V ± 5%	1. Attach input leads to voltmeter 2. Observe and ensure that the value does not leave the range of 5 V ± 5%	Y
Data rate of 9600 baud ± 5%	1. Write simple program which sends data packets with alternating bit values 2. Attach data line to oscilloscope 3. Set oscilloscope to trigger on pulse 4. Measure duration of shortest pulse and ensure that the reciprocal falls within 9600 baud ± 5%	Y

Table 8: Raspberry Pi Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Operating voltage of 5 V $\pm$ 5%	1. Attach input leads to voltmeter 2. Observe and ensure that the value does not leave the range of 5 V $\pm$ 5%	Y
Max current of 1A $\pm$ 1%	1. Attach input leads to multimeter 2. Ensure that the current through the load is at 1A $\pm$ 1% using a multimeter (positive lead attached to 10A and negative lead attached to COM) in series with a 12V battery	Y
Data rate of 9600 baud $\pm$ 5%	1. Write simple program which sends data packets with alternating bit values 2. Attach data line to oscilloscope 3. Set oscilloscope to trigger on pulse 4. Measure duration of shortest pulse and ensure that the reciprocal falls within 9600 baud $\pm$ 5%	Y

Table 9: Display Monitor Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
50 mA $\pm$ 1% to connect display to Raspberry Pi via HDMI port	1. Attach input leads to ammeter 2. Ensure that the current through the load is at 50 mA $\pm$ 1% using a multimeter (positive lead attached to 10 A and negative lead attached to COM) in series with a 12 V battery	Y
Power consumption of 22 W $\pm$ 5%	1. Test the power by plugging the monitor into a Kill a Watt meter 2. Make sure the power reading is 22 W $\pm$ 5%	Y
Supports up to 1080p resolution	1. Check the on screen control in the settings of the display to check if 1080p is supported	Y

Table 10: Camera Status LED Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
The status LED should be visible from 1 m away and have a drive current of 10 mA $\pm$ 5% - 20 mA $\pm$ 5%	1. Set up test circuit shown in Figure 9 to verify that current is between 10 mA $\pm$ 5% - 20 mA $\pm$ 5% 2. Ensure LED is visible from 1 m away	Y

Table 11: Microphone Status LED Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
The status LED should be visible from 1 m away and have a drive current of 10 mA $\pm$ 5% - 20 mA $\pm$ 5%	1. Set up test circuit shown in Figure 9 to verify that current is between 10 mA $\pm$ 5% - 20 mA $\pm$ 5% 2. Ensure LED is visible from 1 m away	Y

Table 12: Microphone Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Audio bandwidth of 300 Hz - 5 kHz	1. Plug microphone in and play audio signals ranging from 300 Hz - 5 kHz 2. Check to see if microphone picks up those signals	Y
Max current draw of 1 A $\pm$ 1%	1. Attach input leads to multimeter 2. Ensure that the current through the load is at 1 A $\pm$ 1% using a multimeter (positive lead attached to 10 A and negative lead attached to COM) in series with a 12 V battery	Y

Table 13: Speakers Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Integrated speakers in monitor (powered by monitor): sound should span 30 Hz $\pm$ 10% to 17 kHz $\pm$ 5%	1. Output an audio file to the monitor that has frequencies ranging from 33 Hz to 16,150 Hz 2. Check to see if the speakers from the monitor successfully output all the audio and one can hear it	Y

Table 14: Camera Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Max current draw of 250 mA $\pm$ 1%	1. Attach input leads to multimeter 2. Ensure that the current through the load is at 250 mA $\pm$ 1% using a multimeter (positive lead attached to 10 A and negative lead attached to COM) in series with a 12 V battery	Y
Pixel resolution between 2592 x 1944 pixels and 3280 x 2464 pixels	1. Take a picture using the camera 2. Check properties of the image file to verify pixel resolution matches the requirement	Y
Can support 640p, 720p and up to 1080p video mode	1. Take a video using the camera 2. Check properties of the video file to verify resolution matches the requirement	Y

Table 15: PIR Sensor Array Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Needs 5 V $\pm$ 1% power (powered by ATmega328)	1. Attach input pin to voltmeter 2. Observe and ensure that the value does not leave the range of 5 V $\pm$ 1%	Y
Can detect motion within the range of 10 cm $\pm$ 5% to 30 cm $\pm$ 5% (it is acceptable for the range to extend past this limit as long as it functions as expected within the specified range)	1. Set up the circuit shown in Figure 10 on a breadboard and wait 30-60s for the PIR to stabilize 2. During this time, the LED may blink - wait until the LED is off 3. Move hand from 0 cm to 30 cm incrementing your position by 10 cm every 10 seconds. When the sensor detects motion, the output pin goes high to 3.3V and lights up the LED 4. Ensure that LED turns on while within the specified range	Y
Field-of-view of 20°$\pm$ 0.5°	1. Set up test harness consisting of two dowels connected at one end and separated by 21 such that they are properly aligned with the tangent from the base of the sensor 2. Set up the circuit shown in Figure 10 on a breadboard and wait 30-60s for the PIR to stabilize	Y

Table 16: Proximity Sensor Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Has a range from 10 cm ± 5% to 30 cm ± 5%	1. Configure proximity sensor for desired range 2. Set up the circuit shown in Figure 10 where OUT is the interrupt pin 3. Move hand from 0 cm to 30 cm incrementing your position by 10 cm every 10 seconds. When the sensor detects presence, the output pin goes high to 3.3 V and lights up the LED 4. Ensure that LED turns on while in the range and off while outside of it	Y
Operation voltage from 2.5 V ± 5% to 3.6 V ± 5%	1. Attach input leads to voltmeter 2. Observe and ensure that the value does not leave the range of 2.5 V - 3.6 V	Y

Table 17: AC/DC Power Converter Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Convert the 120V AC to 12V ± 5% DC.	1. Attach input leads to voltmeter 2. Observe and ensure that the value does not leave the range of 120 V ± 5%	Y

Table 18: Voltage Regulator Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Regulate the voltage to 5 V ± 5% for the microcontroller and PIR sensors	1. Attach output to voltmeter 2. Observe and ensure that the value does not leave the range of 5 V ± 5%	Y
Regulate the voltage to 3.3 V ± 5% for the proximity sensors and status LEDs	1. Attach output to voltmeter 2. Observe and ensure that the value does not leave the range of 3.3 V ± 5%	Y

Table 19: Mechanical Switch Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
The power switch should power off the entire system	1. Once switch is flipped, ensure that all devices are no longer receiving power	Y