DESIGN APPROACH

This document describes the design approach for the Gestura product, including design options considered, a system overview, and subsystem descriptions. Gestura is a gesture-based system to control smart home devices. The system is divided into six subsystems. The six subsystems are Gesture Recognition, User Interface, Power Supply, Sensor Module, Data Communications, and 3D Design. The product has the following constraints: Economic, Safety, Hardware, and Accuracy. The main requirements for the design include communication and accuracy. The product budget is $1000. The product ensures parents that the device cannot be controlled without parent's consent. The hardware of the device is customer-friendly and easy to assemble. The device is accurate within the dimensions of an average-sized U.S. room.

3.1. Design Options

For Gestura, the main challenge was how to collect data for gesture recognition. Two main design choices were considered. The first design option used a wristband that the user would wear to track the movement of the hand. The other approach involved using cameras and AI to detect gestures.

3.1.1. Design Option 1

The first design option considered was using a wristband with a gyroscope and an accelerometer that collected data to send to a microcontroller over Wi-Fi, which would interpret the data and signal an output to the smart home device. The advantages of this approach were that it was less expensive to make and provided more accurate data, but the disadvantage was that wearing a wristband could be uncomfortable or inconvenient for users. This approach was done in the Institute of Electrical and Electronics Engineers (IEEE) study by Alemuda and Lin [1].

3.1.2. Design Option 2

The option that the team chose was to use computer vision to recognize gestures. Two advantages are that this approach is user-friendly and that there is no extra technology that the user must wear. On the other hand, the disadvantages are that it is very costly, and extracting valuable data is much more difficult. The design team decided to tackle the second approach because the comfort and accessibility to the user in this approach outweigh the benefit of the low cost in the first approach.

3.2. System Overview

Figure 3-1 shows how Gestura works at a glance.

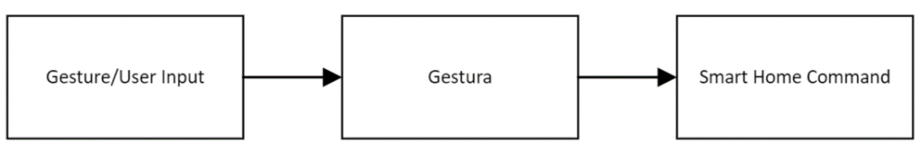

Fig. 3-1: Gestura System at a Glance (Level 0)

Gestura works by taking in a gesture through a camera. Gestura recognizes the gesture and outputs a command to a smart home device. The system works through wireless communications via Wi-Fi. All communications go through the control module. **Fig. 3-2** is an in-depth view of the system.

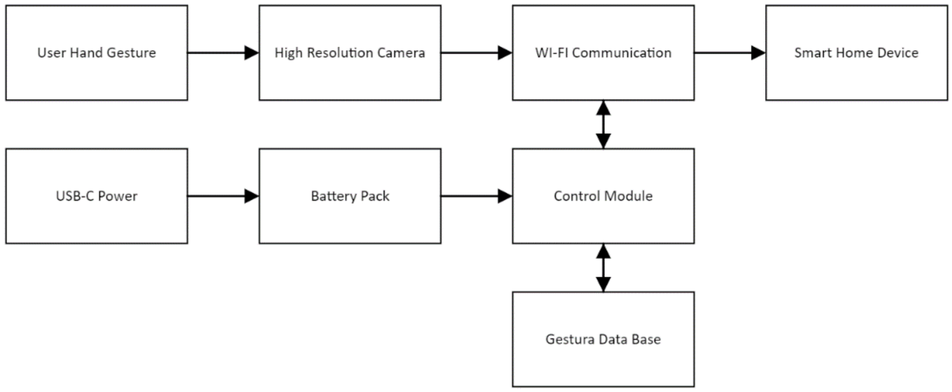

Fig. 3-2: Gestura System Diagram (Level 1)

The user's hand is captured through a camera and sent to the control module. The control module analyzes the video to compare it to its gesture library. The module outputs a command that is sent over Wi-Fi to the smart home device. This interaction is explained more in the following subsystem descriptions below.

3.2.1. Microprocessor

For this project, the team considered two different options for the microprocessor. The first option was the Raspberry Pi 4 Model B, while the second option was an Allen Bradley programmable logic controller (PLC). **Table 3-1** provides the comparisons.

Table 3-1: Advantages vs. Disadvantages

Microprocessor	Price	Wireless connectivity	Supported Programming Languages
Allen Bradley PLC [2]	$255.00	None	Ladder Logic
Raspberry Pi 4 Model B [3]	$89.99	Built-in Bluetooth and Wi-Fi	Python, C, C++, Java, and many more

The Raspberry Pi offers multiple advantages compared to the PLC. First, the Raspberry Pi is more cost-effective. The Raspberry Pi is $165 cheaper than the PLC. Secondly, the Raspberry Pi can use multiple different programming languages, while the PLC can use only one. Additionally, for the PLC to be the team's main control module, it required extra products to connect to the team's different subsystems. The Raspberry Pi, however, already has built-in Wi-Fi and Bluetooth connectivity. Given that the team's project focused on residential use, the team found the PLC exceeded the project's needs, and the Raspberry Pi met the project's specific needs more effectively. **Fig. 3-3** is a picture of the team's chosen microprocessor.

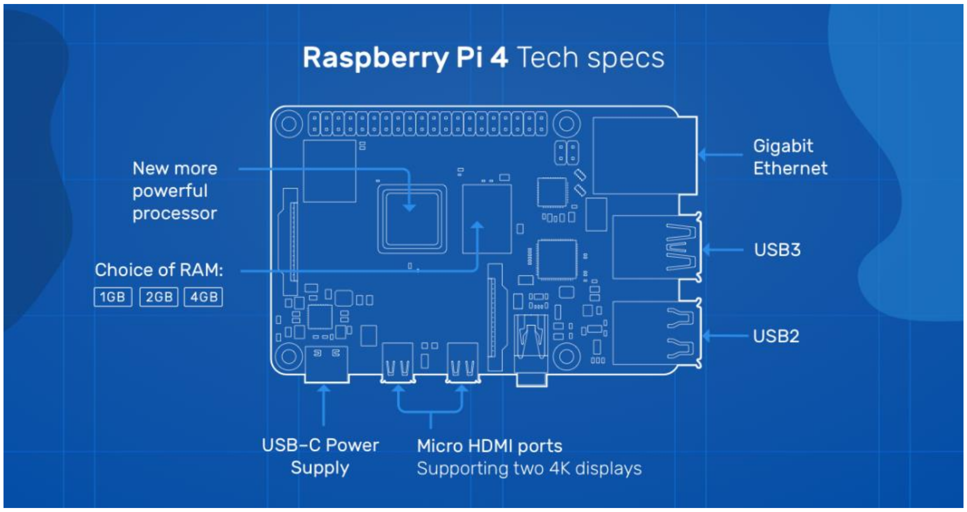

Fig. 3-3: Raspberry Pi 4 Model B [3]

The team chose to have 8 GB of RAM for the Raspberry Pi because of the importance that RAM has in AI and computer vision.

3.3. Subsystems

Gestura has six subsystems. Subsystem 1 is the gesture recognition subsystem. It detects and analyzes the gestures the user uses. Subsystem 2 is the user interface subsystem. The interface is on a tablet to allow the user to interact, connect smart home devices, and display user feedback. Subsystem 3 is the power supply subsystem that powers the LCD screen and Raspberry Pi for the tablet. Subsystem 4, the sensor module subsystem, detects and sends the information of the gestures to the Raspberry Pi. Subsystem 5 is the data communications subsystem, passing data between the control module, the smart home devices, and the cameras. The final subsystem is the 3D design subsystem. This 3D enclosure holds the Raspberry Pi, fan, LCD screen, power supply, and internal wiring for the tablet.

3.3.1. Gesture Recognition Subsystem

The gesture recognition subsystem is responsible for detecting gestures by using computer vision and analyzing the gestures using a machine learning algorithm to assign them to a certain function. Popular programming languages for computer vision and machine learning are C++ and Python [4]. The design team chose Python because of its extensive library [5] and the experience that the design team has with the programming language. **Fig. 3-4** shows a general flowchart of the gesture recognition subsystem.

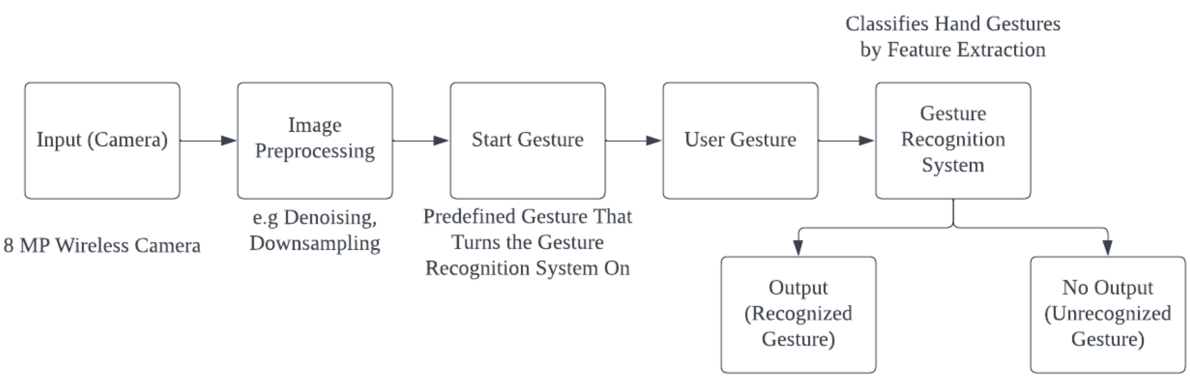

Fig. 3-4: Gesture Recognition Flowchart

An 8 MP camera was chosen because an increase in the quality of the video input results in a more accurate output in vision-based AI. As reflected by Cheng, Gestura needs high-quality input to be effective [6]. Image preprocessing, such as denoising and downsampling, is used to reduce the file size, which increases the efficiency and speed of gesture recognition. The “start gesture” is a gesture stored on the device that makes Gestura look for user gestures once the gesture is performed. This implementation was done to prevent unintentional activation of the device. User gestures are gestures that, once the confidence threshold for the gestures has been reached, activate the smart home device that they are programmed to affect. Training data is stored on the device with predefined gestures. The team is exploring adding user-programmable gestures in the following semester.

3.3.2. User Interface Subsystem

The user interface is on a tablet powered by the Raspberry Pi. The touch screen allows the user to connect to a home Wi-Fi network or new smart home devices. It also outputs if a gesture is detected and the corresponding command. **Fig. 3-5** offers a visual depiction of the user interface.

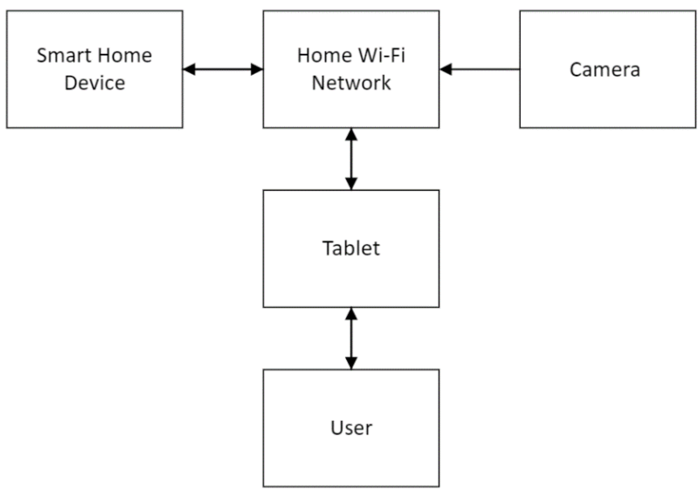

Fig. 3-5: User Interface Diagram

Amazon offers several different touch screens with resolution quality and sizes based on cost. The team decided that based on resolution and cost, the Hosyond 7-inch liquid crystal display (LCD) touch screen offers ample size and quality at a lower price compared to other options. The options are listed below in **Table 3-2**.

Table 3-2: Touch Screen Options

Product Description	Price	Size	Power Dissipation	Refresh Rate	Built-in Speakers
Hosyond 3.5 Inch 480x320 Touch Screen TFT LCD SPI [7]	$17.99	4.61 x 2.8 x 1.34 inches	0.65 W	10 Hz	No
Hosyond 7 Inch IPS LCD Touch Screen Display Panel 1024×600 [8]	$45.99	6.5 x 5 x 0.6 inches	3.1 W	60 Hz	No
GeeekPi 7 Inch Capacitive Touch Screen 1024x600 IPS LCD Display [9]	$59.99	6.65 x 3.93 x 0.32 inches	3.1 W	60 Hz	Yes

Although the Hosyond 3.5-inch touch screen is inexpensive at $17.99 and consumes only 0.65 W of power, its size and refresh rate make it difficult to use. From this, two options were considered. The Hosyond 7-inch touch screen offers many of the same features as the GeeekPi 7-inch touch screen at a lower price apart from built-in speakers, which are not needed for the project, so Hosyond 7-inch touch screen was chosen for its lower price.

3.3.3. Power Supply Subsystem

The Raspberry Pi and the LCD touch screen are powered by a 5-V rechargeable power supply. The power supply uses four 18650 batteries in series to achieve the longevity of the device. The device has 7000 mAh, and the power supply can be recharged by a universal serial bus (USB) Type-C charger. The cameras have a separate power supply. **Fig. 3-6** shows a general diagram of the power supply subsystem.

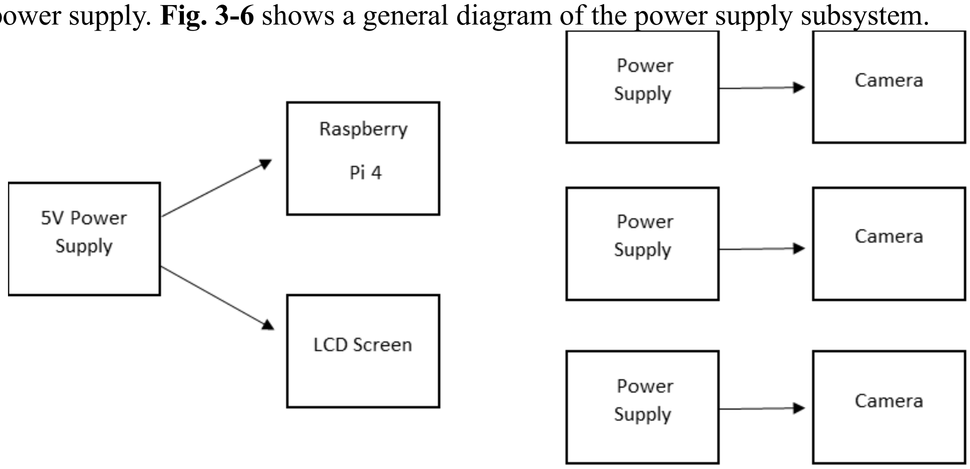

Fig 3-6: Power Supply Diagram

The voltage required by both the screen and the controller is 5 V. The team chose a rechargeable power supply to eliminate the hassle of replacing batteries. With the camera constantly monitoring for gestures, a long-lasting battery life is needed for the device. The team chose four 18650 batteries in series to achieve 7000 mAh. With 7000 mAh, the device can run for two to three days with moderate use before needing a charge.

3.3.4. Sensor Module Subsystem

The sensor module subsystem is responsible for taking in the input from the cameras of the system. This is done by using Wi-Fi to retrieve footage from the cameras, which is sent to the gesture recognition subsystem. **Fig. 3-7** below shows the flow chart from the input to the output of the subsystem.

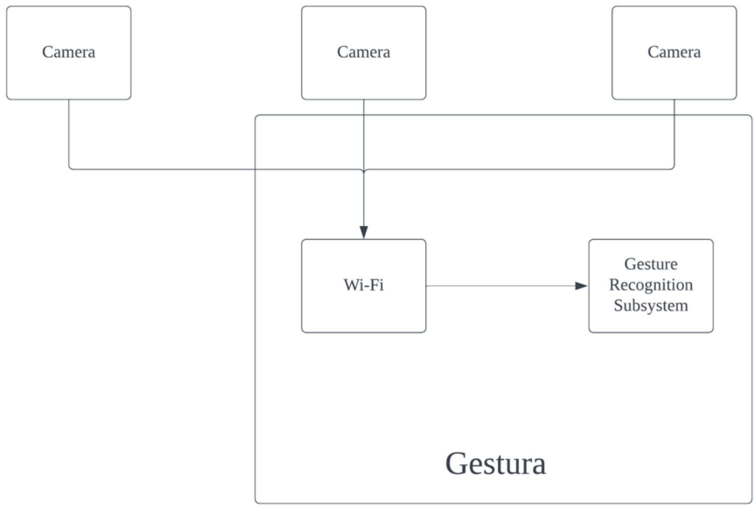

Fig. 3-7: Sensor Module Flowchart

The code is in Python using the CV2 library. This library takes input of each frame of the recording of the camera. This method can be used for multiple cameras. Each is sent to the gesture recognition subsystem to look for a start gesture. Once the start gesture is recognized by a camera, Gestura takes input from only that camera. **Fig. 3-8** demonstrates this interaction.

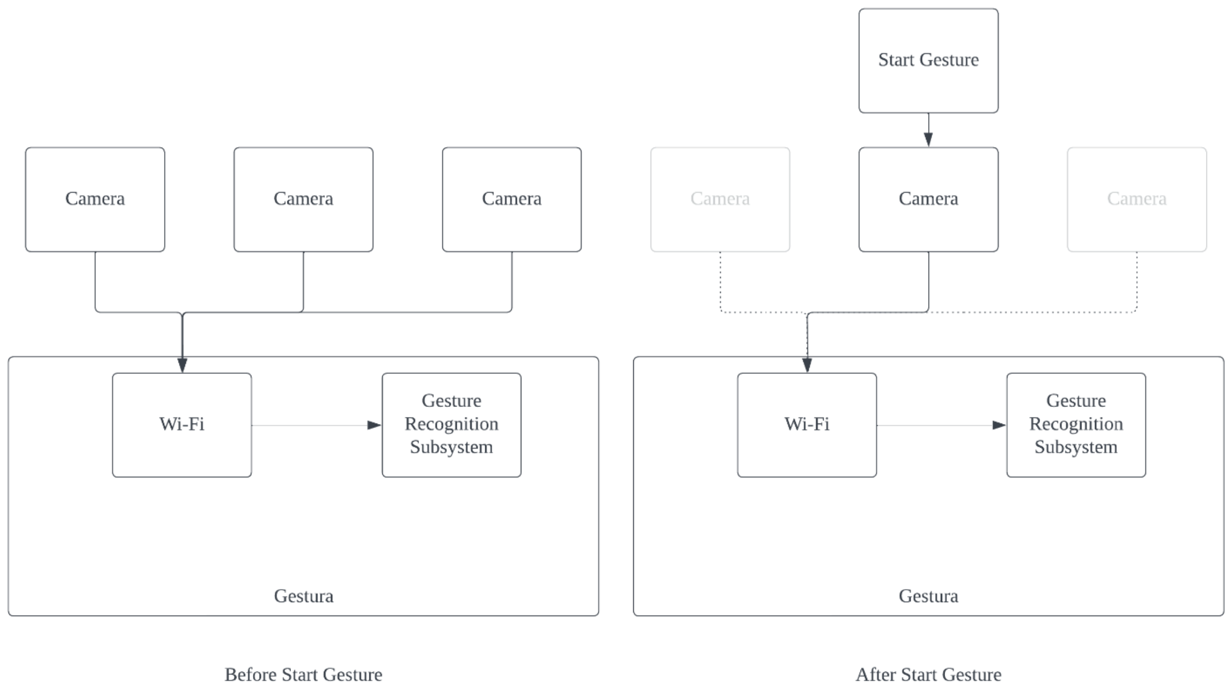

Fig. 3-8: Start Gesture Interaction

This implementation speeds up the response time of Gestura by efficiently using processing power where it is important.

3.3.5. Data Communications Subsystem

The system uses Bluetooth and Wi-Fi to communicate. It uses Bluetooth to connect to new smart home devices and Wi-Fi to communicate with the cameras and smart home devices. Wi-Fi is also used to connect to a home network. **Fig. 3-9** provides a general overview of the data communications subsystem.

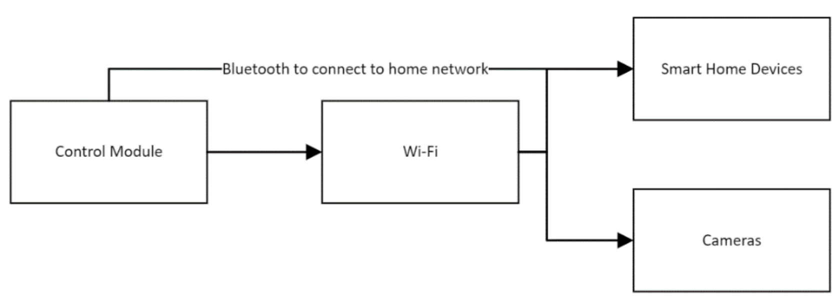

Fig. 3-9: Data Communications Diagram

The cameras and smart home devices use the home network to communicate with the control module. The gesture recognition system is on the control module, which translates the signals into usable commands.

3.3.6. 3D Design Subsystem

The 3D design subsystem holds the Raspberry Pi, fan, LCD screen, power supply, and internal wiring in an enclosure for the tablet. The ports of the Raspberry Pi are also accounted for in the design by leaving open holes to access the ports easily and to plug in wiring. **Fig. 3-10** displays all components that are part of the 3D design subsystem.

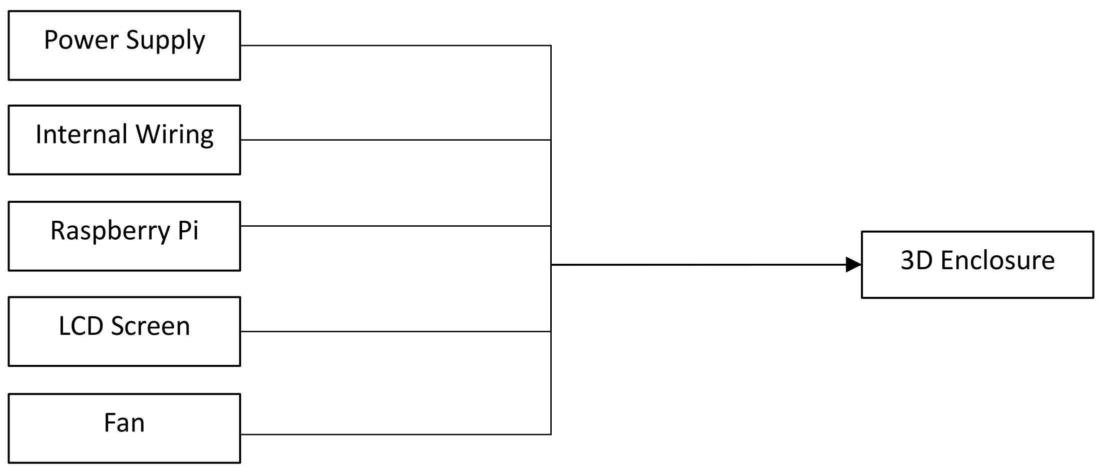

Fig. 3-10: 3D Design Diagram

The enclosure is designed to hold the components that are in **Fig. 3-10.** Its design is in the form of a tablet to make it easy to handle and use for the user.

3.4. Level 2 Prototype Design

For Gestura, the team is planning to encase the control module, cables, battery pack, battery charger module, and so forth in a 3D-printed case. The case has a touch screen monitor on it to act like a tablet. The team intends to use acrylonitrile butadiene styrene (ABS) filament to make the case. ABS plastic is a sturdy enough material for the case to endure moderate wear and tear.

3.4.1. Level 2 Diagram

Building upon the Level 0 and Level 1 diagrams previously shown, a diagram that shows how all these subsystems work together is shown in **Fig. 3-11**.

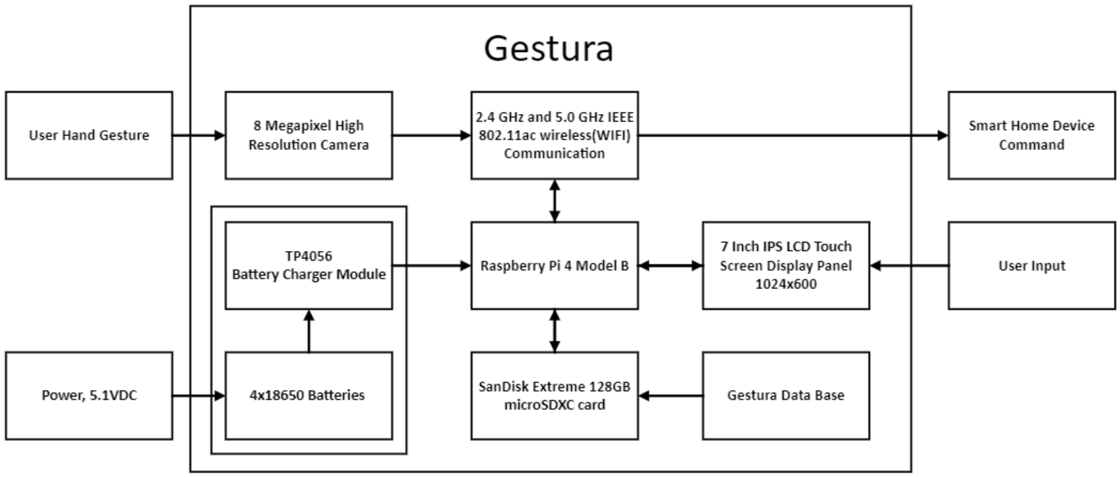

Fig. 3-11: Diagram for Gestura (Level 2)

Each subsystem provides a vital role in Gestura's functionality. The choices made herein provide a foundation of how Gestura works. The next step is to validate these concepts with rigorous testing, addressing any problems as they arise.

References:

[1] F. Alemuda and F. J. Lin, "Gesture-based control in a smart home environment," 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 2017, pp. 784-791, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.120.

[2] “PLC hardware - Allen Bradley 2080-LC30-24QBB series B, surplus in ...,” PLC Hardware, https://www.plchardware.com/Products/RA-2080-LC30-24QBB-B-NSS.aspx (accessed Oct. 24, 2023).

[3] Raspberry Pi, “Raspberry Pi 4 Model B specifications,” https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/ (accessed Oct. 23, 2023).

[4] Y. Noema, “Top 3 programming languages for implementing a computer vision system,” Medium, https://medium.com/imagescv/my-top-3-programming-languages-for-implementing-a-computer-vision-system-2aa0a3ad0a2a (accessed Oct. 20, 2023).

[5] M. R. Munawar, “C++ vs. Python: Choosing the best language for computer vision development,” VisoByte, https://www.visobyte.com/2023/07/cpp-vs-python-choosing-the-language-for-computer-vision-development.html (accessed Oct. 20, 2023).

[6] D. (Wei-Y. Cheng, “Ai with high IQ (image quality).,” Ambarella, https://www.ambarella.com/blog/ai-with-high-iq/ (accessed Oct. 22, 2023).

[7] “Hosyond 3.5 inch 480X320 touch screen TFT LCD SPI display ...,” https://www.amazon.com/Hosyond-480x320-Screen-Display-Raspberry/dp/B0BJDTL9J3 (accessed Oct. 22, 2023).

[8] “Hosyond 7 Inch IPS LCD Touch Screen Display Panel 1024×600 Capacitive Screen ...,” https://www.amazon.com/Hosyond-Display-1024%C3%97600-Capacitive-Raspberry/dp/B09XKC53NH (accessed Oct. 22, 2023).

[9] “GeeekPi 7 inch LCD touch screen for Raspberry Pi, 1024x600 ...,” https://www.amazon.com/GeeekPi-Raspberry-1024x600-Display-Portable/dp/B0CJWXWJ6K (accessed Nov. 3, 2023).