Hand-Gesture Vending Machine

Hussein A. Shelleh, Jhamori Williams-Austin, Logan Terrell, Tyler Bornemann

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — A project built from considerations borne of the COVID-19 pandemic, the Hand-gesture Vending Machine is an attempt to develop an alternative consumer friendly prototype vending machine that remains contact-free. This is intended to be accomplished by making use of object detection machine learning and infrared motion detection. A person and their hand-gestures are observed after approaching the machine, which then identifies the gestures and requests the correct product. Through this request, sensors check for product availability before it is summarily dispensed via motor down a chute for the waiting customer.**

***Index Terms* — Vending Machine, Hand-gesture, Infrared Sensor, ESP32, Snack.**

I. Introduction

Vending machines are an old invention, one with over a thousand years of history; from the Ancient Greeks onwards, there have been many iterations of buying products in an automated fashion [1]. However, given the nature of these machines, physical contact with them remained a necessity until recent technological developments allowed for alternative options. Due to the COVID-19 pandemic, new concerns have arisen regarding contact points of sickness or disease. Through optics, there is a simple solution.

The Hand-Gesture Vending Machine is a four-option vending machine that operates entirely with no physical contact, making use of infrared sensors and object detection software to detect customers and read their actions and responses. As a failsafe, this includes a two-layer process - one for selection, and a second for confirmation. Once it is confirmed, the product selected is checked for availability, and if available is subsequently dispensed. This system will be evaluated for each selection, confirmation with product, confirmation without product, and rejection.

II. System Components

We There are many notable components within our prototype device. The purpose of this section is to give an overview of each, and how they are subsequently put together to define the greater product.

A. Printed Circuit Boards (PCBs) / Microcontroller

The PCBs in use for this project are twofold. The first is the main board, which possesses the plug for the power supply, slots for the custom voltage regulators to insert into, and the ESP32-WROOM-32D [2]

. The ESP32 is a powerful series of microcontrollers, and with just enough general-purpose Input/Output (GPIO) pins for our purposes, it is a perfect fit for the project. Additionally, this board has a set of connections set aside to lead into the various components - including the second board.

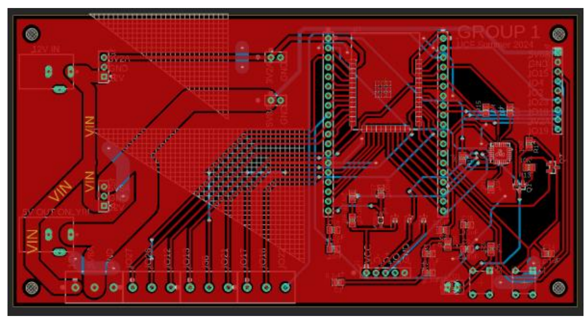

Fig 1. Main PCB.

The second PCB board we are working with is a secondary board of connection points and MOSFETs for the motors; this was done to minimize the size of the main board and to keep the overall stress on the main board as low as possible. This board does not possess a microcontroller of its own and is primarily a voltage and signal relay to minimize the use of high-voltage cables and risk of failure.

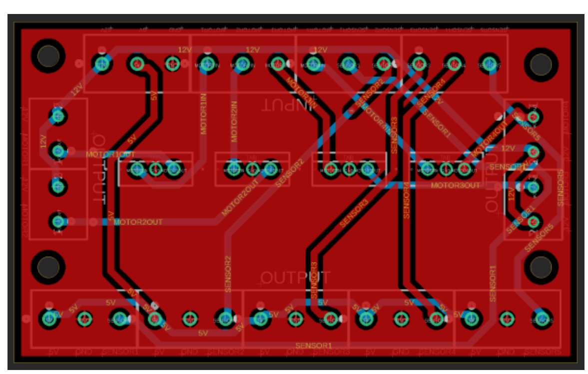

Fig. 2. Secondary PCB.

B. Single Board Computer

The Single Board Computer we are using is the NVIDIA Jetson Nano. Jetson Nano will be used to execute deep learning operations; more specifically, it will be used to classify what hand-gesture is being displayed by the user. The GPU on the Jetson Nano is a 128-core NVIDIA Maxwell architecture-based GPU, and the CPU is a quad-core ARM Cortex-A57 MPCore processor [3]. The Jetson Nano is powered by a five volt (5V) and four ampere (4A) barrel jack. The different interfaces on the Jetson that have been used are an HDMI, four USB Type A ports, and the USB micro-B port. These ports have been used for evaluating the Jetson Nano with the deep learning, hand-gesture model.

Furthermore, there is a camera connected to the Jetson Nano to receive a constant video feed when the Jetson is powered on. The software that the Jetson Nano has used for testing includes the PyCharm IDE, Ubuntu operating system, and Python as the programming language. In practice, the Jetson Nano will be powered on by the barrel jack when the front IR sensor detects a person at the machine. When the Jetson Nano is powered on, it will automatically run the hand-gesture recognition application. After the application completes, the chosen value of one, two, three, or four will be exported to the chip.

C. Power Supply

The power supply connected to the main board by a two and a half millimeter (2.5mm) barrel connector is the ALITOVE 12V 6A 72W AC to DC power supply adapter [4]. The power draw of all the components added up is 46.33W. Then accounting for the inefficiency of the voltage regulators brings us up to 51.47W. Which this power supply will manage with no problem with its maximum wattage of seventy-two watts (72W).

D. Voltage Regulator PCBs

There are two voltage regulators attached to the device, each designed to sit on the main board and take the current load in its place. The first regulator of the two converts the twelve volt (12V) voltage from the power supply down to five volts (5V) and up to eight amperes (8A). This is exceptionally high current, so having it be maintained on an independent board seemed to be the best option rather than risk breaking the main board. Also, the required trace width for 8A has been accounted for. The second regulator of the two steps down from the same twelve volts (12V) to three point three volts (3.3V) and two amperes (2A). This current demand is far lower, and while it should suffice to be on the main board, caution prevailed.

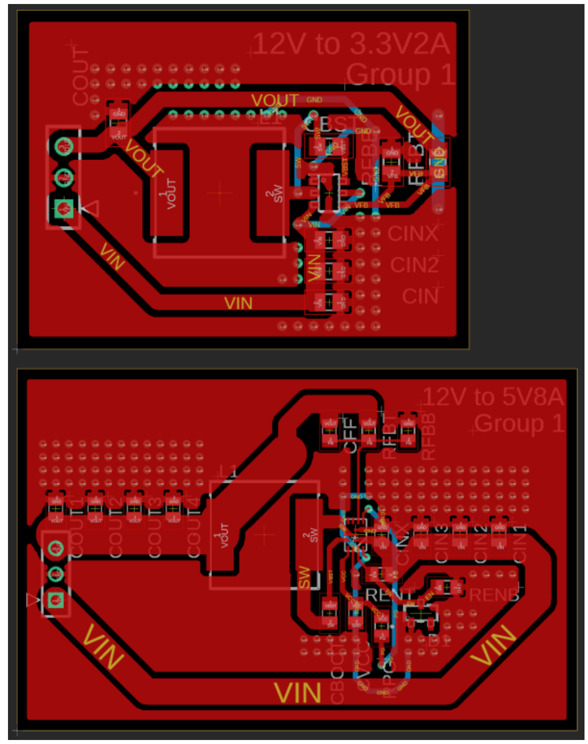

Fig 3. 3V2A (top) 5V8A (bottom) voltage regulator

E. IR Sensors

There will be a total of five E18-D80NK infrared (IR) sensors used for this project [5]. These sensors operate via a diffuse reflection style transmitter-receiver combination; this in particular allows for the modulated infrared signal transmitted by the sensor to be easily picked up regardless of the ambient light or temperature and should allow the sensors to remain functional so long as anything enters its range. Additionally, the IR sensors we have selected have an adjustable range between a few scant millimeters to just short of a meter. Finally, these sensors have a very intuitive high/low signal that allows for ergonomic GPIO pin usage.

There are two different purposes for these IR sensors. First, we will use the sensors to let us know when a person is detected. Our first IR sensor of the five will be a front facing sensor that accomplishes this task; it will activate the screen once something enters its range and initializes the entire system. Second, we will use the sensors to check for our available stock within the machine. The remaining four E18-D80NK IR sensors will be used to check for availability of the frontmost snack. In the case the item being selected is out of stock, this will result in the cancellation of the customer's request and a notice of no stock on the LCD screen.

F. Camera

The camera connected to the NVIDIA Jetson Nano through an CSI-2 connected is a Raspberry Pi Camera Module 3 [6]. It is a new and compact camera from Raspberry Pi, featuring a Sony IMX 708 twelve-megapixel sensor with HDR and phase detection autofocus. With an impressive resolution of 11.9 megapixels and a variety of video modes, it allows us to explore the best performing video mode for our purposes. The CSI-2 specification ensures seamless compatibility and ease of use.

G. Lenses

The Hand-gesture Vending Machine is making use of four Edmund Optics lenses; these lenses are specifically the twelve millimeter diameter 32-014, the nine millimeter diameter 48-682, the nine millimeter diameter 45-292, and the nine millimeter diameter 32-968 [7][8][9][10]. These small sizes have required a custom-designed mount for the lenses - and for the camera, to keep the whole thing stable. The lenses themselves have no coatings, as there is no wavelength-specific optical element involved.

Fig 4. 3D-printed Mount for the Camera and Lens System.

H. Motors

The function for the G12-N20 DC motors will be to solely vend snack options out of the vending machine [11]. There will be a total of four motors for the four different snack options we have. The motors will be placed on the back of the vending machine with a turning coil attached to it so it will be able to vend the necessary item. We will be working with four small DC motors; despite their size they have an impressive torque of two kilogram-centimeters and a stall torque of sixteen kilogram-centimeters. Since we will barely be moving a single kilogram at a time with the turning coil, this setup will work out well for the prototype Hand-gesture Vending Machine.

I. Screen

The screen that will be displaying the user interface, interactive content, and showcasing product information is a 4-inch SPI LCD Display [12]. The screen will be activated once the front facing IR Sensors obtains a detection. Once the LCD display is activated there will be a home menu screen with four different options to choose from. When an item is chosen based on the hand-gesture a new screen will appear on the LCD display confirming your item selection. The LCD display will be connected to the microcontroller using SPI communication.

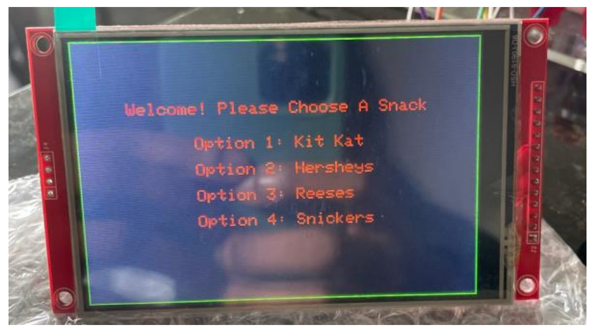

Fig. 5. Welcome Screen for the LCD Display.

J. Fans

There will be two to three Noiseblocker BlackSilent 12V DC fans inside of the Hand-gesture Vending Machine [13]. One as intake, one for output, and one strapped to the NVIDIA Jetson Nano's built-in heatsink. Our highest demanding component, the Jetson Nano, gets physically warm at idle so it is safe to assume we will need to have sufficient airflow to keep all our components in an enclosed space. The most important aspect of the fans was cost and noise. Our fans encompass a maximum rotational speed of 1200 RPM, 3-Pin power connector, and a noise level of sixteen decibels.

K. LEDs

The LEDs will be used for the Hand-gesture Vending Machine, contributing to its functionality and effectiveness. By using LEDs, the Hand-gesture Vending Machine can minimize power consumption while maintaining bright illumination. The chosen LEDs will be white LEDs. The white LEDs offer a balance between reliability, performance, and affordability. The white LEDs will be connected to the microcontroller. When the microcontroller turns on all the white LEDs will also turn on to illuminate the inside of the vending machine. The white LEDs offer high brightness levels, ensuring that the users can visibly view products on display.

III. HARDWARE SUBSYSTEMS

Using We define a subsystem as a group of related parts that serve a purpose that will be connected to our microcontroller for power or input and output.

A. Power Supply Subsystem

The power supply subsystem makes up the power supply and the two voltage regulator PCBs, essentially this is the power delivery of all the components. The power supply's twelve volts (12V) is connected to the fans and the two voltage regulators. One of the voltage regulators outputs three point three volts (3.3V), which goes to the microcontroller and subordinately, the LEDs. The other voltage regulator outputs five volts (5V), which goes to the NVIDIA Jetson Nano, the motors, the IR sensors, and the screen. The currents needed by all the components have been calculated and each point of the supply can handle a load greater than needed.

The twelve volt (12V) Power supply is connected to the main PCB with a standard two point five millimeter (2.5mm) barrel jack. The voltage regulators are separate boards to make the replacement of one easy. The Jetson Nano gets its five volt (5V) supply from another barrel jack coming off the main board. Then the rest of the components get their power off the main board or from the secondary board, which gets its power from the main board. The two voltage regulator PCBs can be seen attached to the main PCB in Figure 6.

Fig 6. Main PCB powered only by the wall.

B. Front IR Sensor Subsystem

The front IR sensor subsystem consists of one of our IR sensors facing externally. The IR sensor will read HIGH if a potential customer comes close to the machine. When a customer comes up to the machine, the microcontroller counts for a specified amount of time before turning off to not keep turning on and off as they stand there. The sensitivity of the IR sensor can be controlled on the back of the sensor to avoid a false positive.

C. Fan Subsystem

The fan subsystem is made up of three fans. Two of which will be directly connected to our 12V source, the other will be connected to the Jetson Nano which has its own PWM system to control the fan depending on the load. This constant supply of fresh air should be sufficient in protecting the electronics that we will be putting in an enclosed space.

D. Internal IR Sensor Subsystem

The internal IR sensor subsystem is the other four IR sensors. One for each of our options. These sensors will be used to detect if an object is in stock or not. If the IR sensor reads LOW, then the item is out of stock and should not be able to be sold to the customer. The sensitivity of the IR sensor can be controlled on the back of the sensor to avoid a false negative. The sensitivity is adjusted so the customer could not make the machine read in stock just by standing in front of the option. The power delivery and GPIO connections have been moved to the secondary board to make connections more secure and easy.

E. Camera and Single Board Computer Subsystem

The camera and single board computer subsystem are the Raspberry Pi camera module 3 and the NVIDIA Jetson Nano. The NVIDIA Jetson Nano has a CSI-2 connector that the camera is plugged into. The NVIDIA Jetson Nano will take images from the camera and using artificial intelligence, it will recognize gestures . Two different stages will be a selection stage and a confirmation stage. If the confirmation stage is okayed by the user, then the selection will be sent back to the microcontroller.

F. Motor Subsystem

The motor subsystem consists of four motors controlled by four IRF510 MOSFETs, which are low-drive voltage components. The microcontroller sends a 3.3V signal to the gate of each MOSFET, which allows current to flow from the drain to the source, completing the circuit from the 12V power supply to the motor and then to ground. The power delivery and GPIO connections have also been moved to the secondary board to make connections more secure and easy.

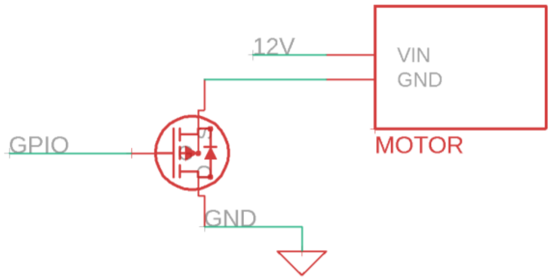

Fig 7. MOSFET and Motor Configuration.

IV. Software

This section will cover the software of our Hand-gesture Vending Machine project. The software itself can be split into two main parts: The embedded code, and the machine learning algorithm.

A. MCU Code

The embedded software behind the Hand-gesture Vending Machine will be flashed onto our PCB using the C language. The microcontroller that is being used in this project is the ESP32. This particular microcontroller offers numerous GPIO pins, SPI pins, and I2C pins. The SPI pins will be used to connect the LCD display for the user interface. Using most of the peripherals the software will control this. The LEDs, LCD Display, IR Sensors, fans, and DC motors will be connected to the ESP32. The connections will be made through wiring each component to the PCB.

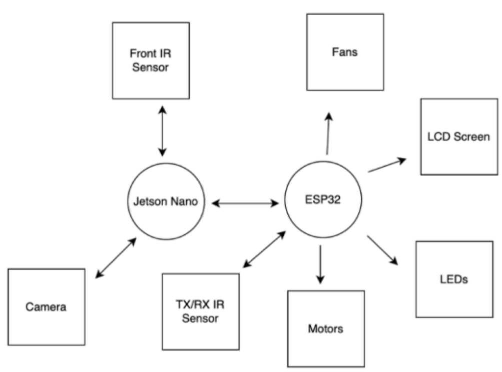

Fig. 8. Microcontroller Component Connections Layout

The connection layout pictured above can depict how the software will work with the different components. As you can see from the layout there will be many different components communicating with the MCU at one time.

The way the software will work for the Hand-gesture Vending Machine starts with the front-facing infrared sensor. When the IR sensor detects an object - in this case, the object should be a person - the LCD display will turn on with a welcome menu screen. When there is nobody detected, the LCD display will be turned off. The fans connected to the microcontroller will be on the moment the microcontroller turns on to make sure no overheating occurs from the jetson nano and other components.

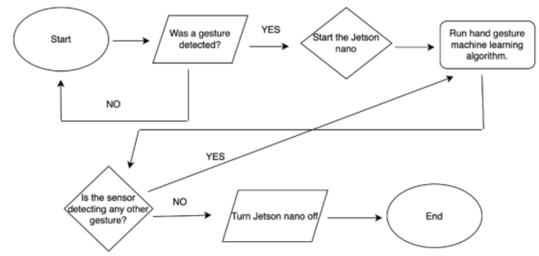

Fig. 9. Front IR Sensor Flowchart.

The LEDs will be on the inside of the machine to ensure visibility of the items inside. Our IR Sensors will be used for detecting if an item is in stock, reading binary 1s and 0s. When a hand detection is detected the LCD display screen will change, and the motors will move to dispense an item.

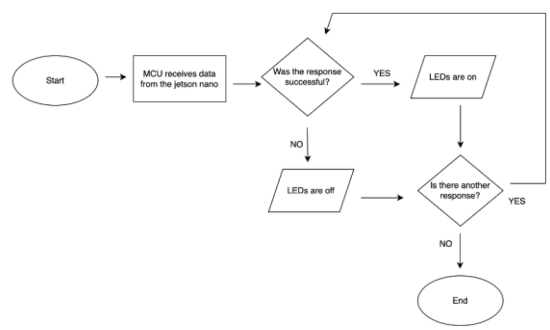

Fig. 10. LED Software Flowchart.

B. Jetson Nano Code

The model that was chosen for the deep learning portion of the Hand-gesture Vending Machine was the Convolutional Neural Network (CNN); the CNN specializes in image classification. Neural networks are a collection of node layers that make up a system of communication like one of the neural connections of the brain, with the node layers including the input layer, the hidden layer, and the output layer. The convolutional neural networks are made up of a feedforward neural network, meaning that this type of neural network does not include a loop and it only goes one way. Two features that differentiate the CNN and other neural networks are filters and pooling layers. A filter is a matrix made to determine things like colors or edges to further understand the data. A pooling layer's purpose is to compress the data, so the overall application does not become overly heavy and slow due to processing a large amount of data.

The system of communication consists of an input layer, a hidden layer, and an output layer. The input layer takes the input parameters, processes them, and passes its output to the hidden layer. The hidden layer takes the input from the input layer and further anatomizes the data that was received and learns the intricacies of each factor in the received data. There can be multiple layers within the hidden layer and in this project, there are two. This output from the hidden layer is passed to the output layer, where the output layer makes the final predictions for the input's question.

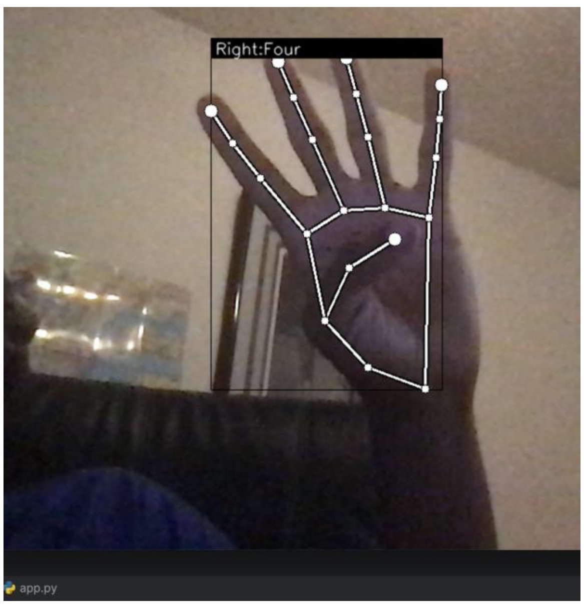

Fig. 11. Object Detection – Right hand: Four.

The Convolutional Neural Network had to be trained to recognize the different hand-gestures, with the acceptable hand-gestures being one, two, three, four, thumb up, and thumb down. The way this was executed was first it had to be understood that the camera module read in frames. Secondly, the hand had to be broken down into twenty different points; on each finger, there was a point placed at the base of every finger, at the tip of every finger, and on every joint of every finger. There also existed a point at the wrist of each hand. The way each hand-gesture was trained was there were thousands of snapshots taken of each of the 6 hand-gestures and, simultaneously, labeled. These snapshots saved where each of the twenty points were relative to each other and related that snapshot to the selected name. Our data consisted of around 4100 of these snapshots.

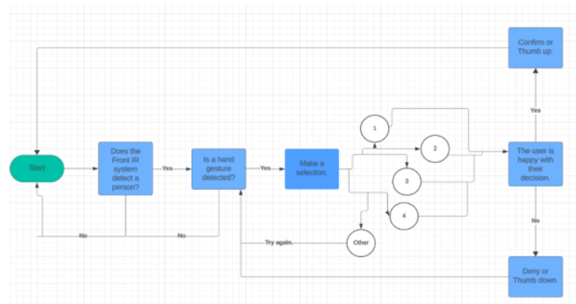

Fig. 12. Object Detection Flowchart.

The object detection logic follows a simple flowchart. Of the hand-gestures, there are two groups. There is group one, which consists of one, two, three, and four. There is group two, which consists of thumb up and thumb down. Firstly, it is determined if a hand-gesture is present in the current frame. If a hand-gesture is not present, then the camera runs until the system itself turns off or there is a hand-gesture present. If a hand-gesture is present, there will be a clock, that counts to three, that begins. If this counter can reach three, the hand-gesture that was registered is examined if it is in group one. If the hand-gesture shown does not exist in group one, then the user is prompted to present another hand-gesture. If the hand-gesture shown exists in group one, then they are prompted to confirm or deny their decision. The user now needs to gesture either a thumb up or thumb down. If the gesture the user selects does not belong to group 2, then the user is prompted to select a gesture belonging to group 2. If they select a thumb down, the flowchart takes them to the beginning where they are prompted to show a hand-gesture belonging to group one. If they select a thumb up, the decision is exported to the PCB and the object detection program is closed. If any gesture, that is shown to the camera, is dropped out of frame, then the counter resets to zero.

Some of the notable libraries that will be used are the Numpy library for dealing with the large matrices in this project, the Opencv library for handling everything about the camera and camera feed, the Keras library for dealing with the neural networks' aspect of this project , and the Tensorflow library for further support of neural networks.

V. Fabrication and Assembly

Given the Hand-gesture Vending Machine is a prototype, it should at the very least resemble the vending machines it is intended to compare to. As such, we have worked towards constructing a proper frame and mechanical components to house all the pieces of the overarching design. This meant motor mounts and coils, a lens system and camera mount, and - if all goes well - a complete box-shaped vending machine.

Fig 13. 3D-printed motor mount and motor coil.

The motor mounts, motor coil, and lens system and camera mount were all designed within 3D modeling software, primarily SOLIDWORKS. They were then printed out using 3D-printers, tested for functionality, and tweaked accordingly until we were satisfied with them.

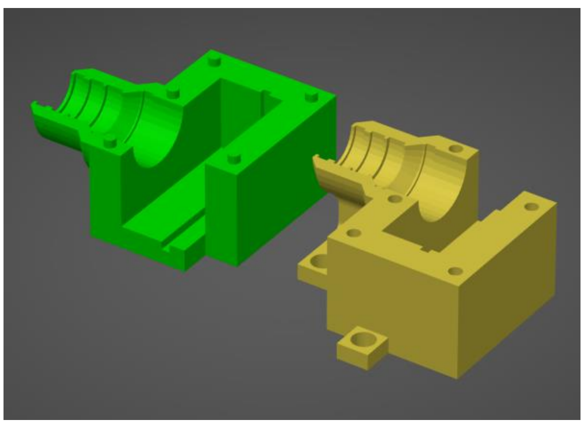

Fig 14. 3D-printed lens system and camera mount.

When it comes to the full frame for the Hand-gesture Vending Machine, however, there have been difficulties with determining a suitable size for the frame. We did not want to make it too large, nor too small - but without getting started with making the frame and installing components, this had taken a significant amount of time to decide on. At this time, the full ‘box’ at forty centimeters (40cm) on each side will be built with medium-density fiberboard (MDF) due to its blend of cheap cost, moderate durability, and relative ease to cut without expensive tools. There will be slots for the snacks, drilled-in plexiglass, the PCBs, screen, Jetson Nano, power supply, external IR sensor, and camera mount - while still having some space inside in the case components need to be swapped out. The top of the prototype will specifically open up in order to facilitate these replacements - or to restock as needed.

VI. Conclusion

In conclusion, this project is an attempt at an alternative vending machine - one which minimizes physical contact in the name of minimizing the risk of spreading disease through touch. Through the efforts we have made, we believe the design we have put together is a positive step forward for the future.

VII. Members

Hussein A. Shelleh is currently a senior student at the University of Central Florida. He is seeking two degrees; one in Electrical Engineering, and another in Photonic Science and Engineering. Once he graduates, Hussein would like to spend time visiting family overseas.

Jhamori Williams-Austin is currently a senior student at the University of Central Florida seeking a degree in Computer Engineering. Jhamori would like to travel.

Logan Terrell is currently a senior student at the University of Central Florida seeking a degree in Computer Engineering. Once he graduates, Logan would like to get a job centered around embedded systems engineering.

Tyler Bornemann is currently a senior student at the University of Central Florida seeking a degree in Electrical Engineering. Once he graduates, Tyler would like to learn and use more CAD software.

Acknowledgement

The authors wish to acknowledge the assistance and support of Dr. Arthur Weeks, Dr. Chung-Yong Chan, and Dr. Lei Wei, for without the time they had spent with us, we may not have reached where we are now.

The authors also wish to acknowledge Dr. Aravinda Kar and Dr. Peter J. Delfyett, for their advice and

recommendations regarding the optical components of our device.

REFERENCES

[1] "From holy water to luxury cars, vending machines dispense all," GovTech Singapore, Feb. 18, 2022. https://www.tech.gov.sg/media/technews/history-of-vending-machine

[2] "ESP32-WROOM-32D & ESP32-WROOM-32U Datasheet," Espressif Systems. https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf

[3] "NVIDIA Jetson Nano Software Documentation," NVIDIA. Available: https://docs.nvidia.com/jetson/index.html

[4] "ALITOVE 12V 6A 72W AC to DC Adapter 100-240V AC to DC 12 Volt 6 Amp Transformer with 5.5x2.5mm Plug" https://www.alitove.net/product/power-supply-adapter/12v-power-supply?product_id=67

[5] "Hands-On Technology User Guide: E18-D80NK Adjustable Infra-Red (IR) Proximity Sensor." Available: https://handsontec.com/dataspecs/sensor/E18-D80NK%20IR%20Sensor.pdf

[6] "Raspberry Pi Camera Module 3 Product Brief," Raspberry Pi. https://www.mouser.com/datasheet/2/635/camera_module_3_product_brief-3241376.pdf

[7] "12mm Dia. x 12mm FL Uncoated, Double-Convex Lens," Edmund Optics. https://www.edmundoptics.com/p/12mm-dia-x-12mm-fl-uncoated-double-convex-lens/2090/

[8] "9mm Dia. x -18mm FL, Uncoated, Double-Concave Lens," Edmund Optics. https://www.edmundoptics.com/p/9mm-dia-x18mm-fl-uncoated-double-concave-lens/8934/

[9] "9mm Dia. x 45mm FL Uncoated, Double-Convex Lens," Edmund Optics. https://www.edmundoptics.com/p/9mm-dia-x-45mm-fl-uncoated-double-convex-lens/5776/

[10] "9mm Dia. x 18mm FL Uncoated, Double-Convex Lens," Edmund Optics. https://www.edmundoptics.com/p/9mm-dia-x-18mm-fl-uncoated-double-convex-lens/2737/

[11] "Hands-On Technology User Guide: G12-N20 Geared Mini DC Motor." Available: https://www.handsontec.com/dataspecs/motor_fan/GA12-N20.pdf

[12] "LCD MODULE QD-TFT3956 SPECIFICATION QD-TFT3956," 2021. Available: http://www.lcdwiki.com/res/MSP4022/QD-TFT3956_Specification_V1.0.pdf

[13] "NoiseBlocker BlackSilent Pro." NoiseBlocker. Available: https://www.blacknoise.com/datas/downloads/datasheets/TData_BSP120_122012_de_en.pdf