PIXEL Design Review 1
Precise Image eXtraction and Enhancement Lab

Jack Allardyce, Lindsey Canessa, Josiah Owens, Victoria Ryan, Delaney Smith
Professor Schafer
Senior Design
February 21, 2025

System Block Diagram

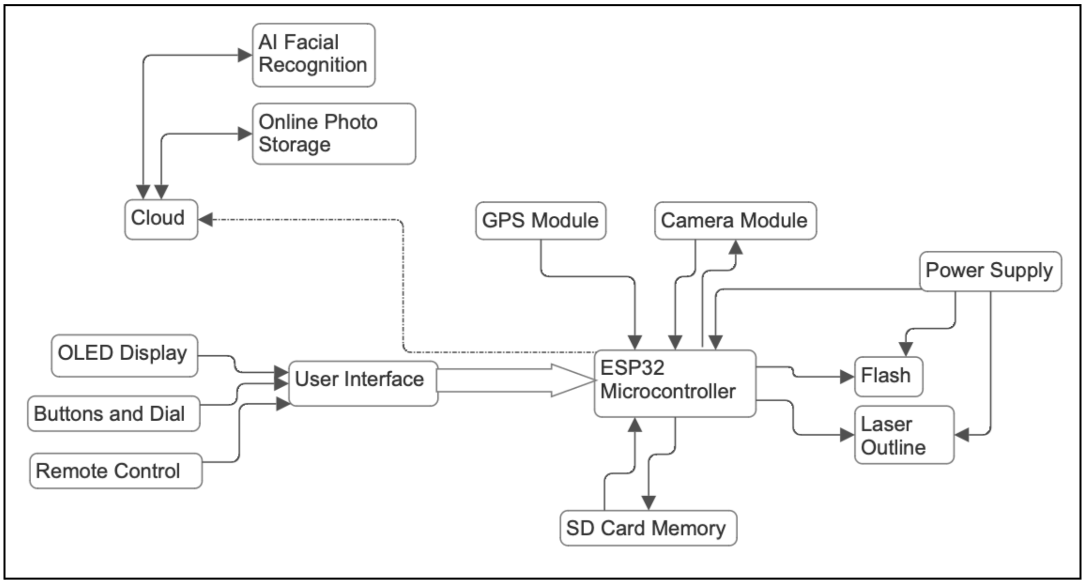

Figure 1. *System block diagram depicting all major devices of the project.*

List of all major subsystem designs:

1. Camera Control Module and Interface Design

   EPS32

   For wireless communication, we have determined that the technology of the ESP32-S3-WROOM-1-N16R8 microcontroller is best suited for our needs. It has built-in WiFi and Bluetooth capabilities. It eliminates the need for additional wireless modules, reducing costs and simplifying the design. It can handle data transmission and control operations efficiently and has 16MB of flash memory.

   The integrated Bluetooth Low Energy module in the S3 allows for wireless remote control, crucial for our camera feature of being able to remotely take a picture.

Camera Module

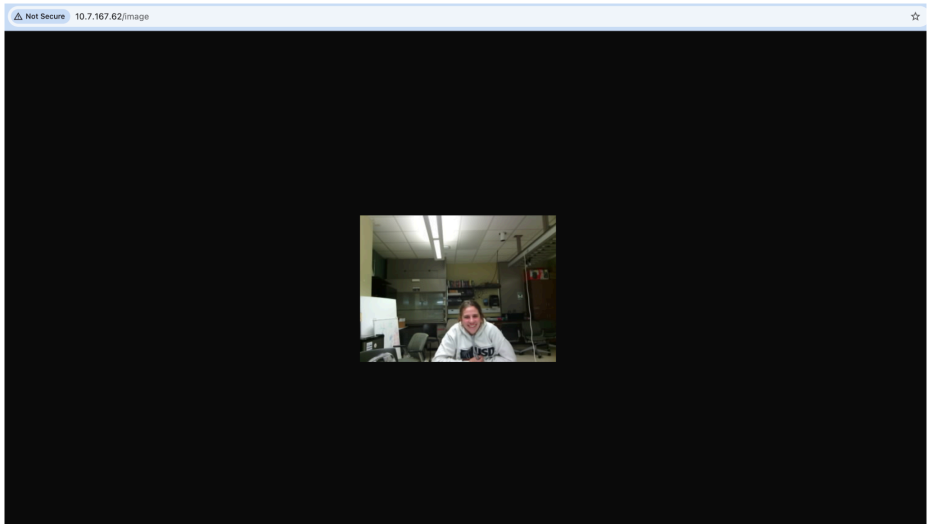

Figure 2. *Web Server displaying photo taken with Arducam, sent over wifi*

Major components of camera module subsystem and the function they offer:

- The Arducam Mega is the image capture device for this project and will be connected to the ESP32 S3 microcontroller via a header pin on the board.
- It was chosen for its extensive libraries and SDK with significant documentation. It also uses minimal pins (6) and interfaces with the S3 over SPI (SPI mode 0). It requires a constant 3.3 V power supply and does not require additional decoupling capacitors. The camera module has successfully been tested with following functionalities:
    - Capturing an image and sending it over serial communication
    - Establishing a WiFi connection on the ESP32
    - Transmitting the captured image to a web server, where it was successfully displayed as shown above in Figure 2
- Arducam Mega
    - Captures image based on shutter button input (manual button on camera (1) or bluetooth remote shutter (2))
    - Communicates with the ESP over SPI0
    - Transmits captured image for display on the viewfinder and upon user input either gets deleted or sent to SD card or over Wifi to google drive for post-processing, facial recognition, and sorting

Laser Lines

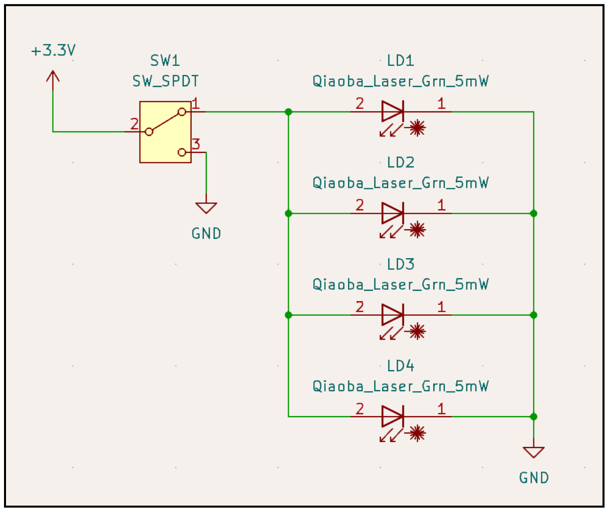

Figure 3. *Schematic of laser outline: four lasers and a switch.*

Major components of subsystem and the function they offer:

- [The lasers] are 5mW green lasers. Green is the most easily visible color of laser, so it should show up better compared to the red lasers tested previously.
- The lasers' input voltage is controlled by a switch for ease of use.

Essential connections on the major components:

- The lasers are connected in parallel.
- The lasers are powered by 3V-3.7V, so they can be connected to the 3.3V logic level from the power subsystem.

GPS Module

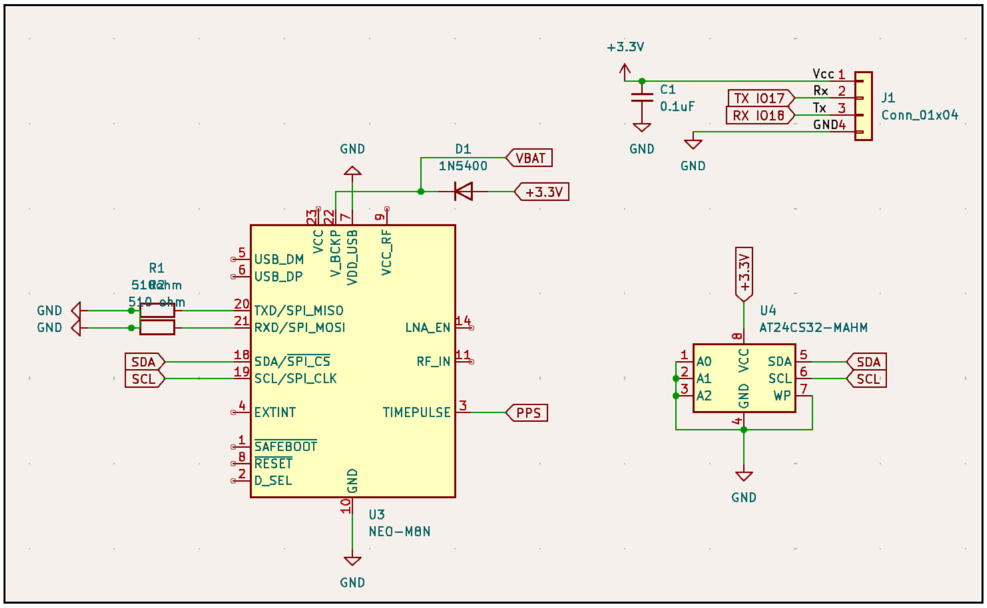

Figure 4. *Schematic of GPS module: GPS is interfaced with EEPROM found on chip and powered with 3.3V.*

Major components of subsystem and the function they offer:

- The NEO GPS Module provides accurate location and timing data and supports both UART and I2C communication
- AT24CS32 EEPROM consists of protected memory used to store GPS configuration and calibration data
- Voltage regulator provides steady 3.3V output from higher input voltage (see power subsystem)

Essential connections on the major components:

- The GPS Module connects to the ESP32 through the Vcc, TX, RX, and GND pins to provide regulated power supply and UART serial communication
- An antenna connects to the GPS chip via U.FL connection, allowing for communication with satellites to extract desired data

LED Flash

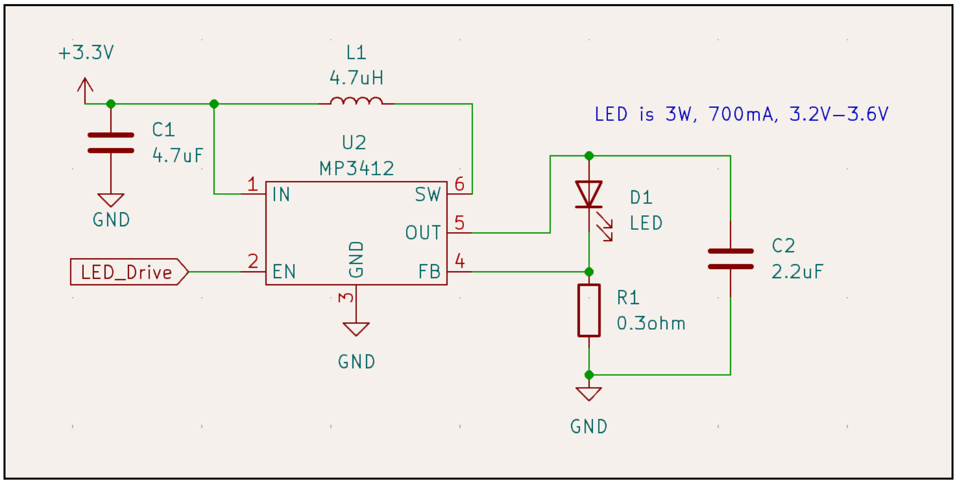

Figure 5. *Schematic of camera flash: LED & LED driver with appropriate inductor, resistor, and capacitors.*

Major components of subsystem and the function they offer:

- The LED is a high power COB device; it's a 3W, 700mA, and 3.2-3.6V LED.
- The driver is the MP3412EJ-LF-Z from digikey; it has input range of 0.8V-4.4V, output range of 2.5V-5.0V, and is controlled with PWM through the EN pin.

Essential connections on the major components:

- The ESP32 connects to the EN pin of the driver. The duty cycle of the PWM determines the current and thus brightness of the driven LED.
- The input and output pins of the driver have decoupling capacitors (with values given by datasheet).
- The FB resistor, called R_SENSE, determines the current of the LED given by I_LED = 0.2V / R_SENSE.
    - With the 700mA LED, the resistor is about 0.3 ohms. Is this value small enough to justify omitting the resistor?

SD Card

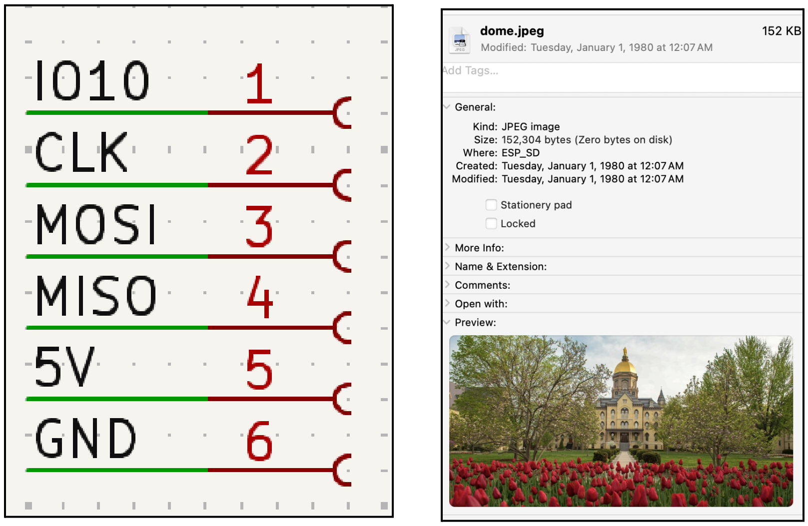

Figure 6. *SD card pins and demonstration of SD card image quality preservation*

Major components of subsystem and the function they offer:

- The SD card module allows the images from the camera to be saved, even if the camera is not connected to WiFi.
- A 32 GB, FAT microSD card is used to store the images so that there is enough space for multiple images to be stored at once and the FAT formatting is necessary for interfacing with the ESP32.

Essential connections on the major components:

- The SD card is connected to the ESP32 with an SD card module for SPI interfacing with the chip select pin on IO10.
- The SD card module needs to be powered with 5V.
- The specific connections for the SD card module can be seen in Figure 6, above. In the final product, only the SD card holder will be used in the board, not the entire SD card module.
- Figure 6 also shows the image that is uploaded from the ESP32 to the SD card when the SD card is plugged into a laptop. The image and byte size match the original image.

2. Power Design

Batteries, Voltage Regulation, and Low-Power LED

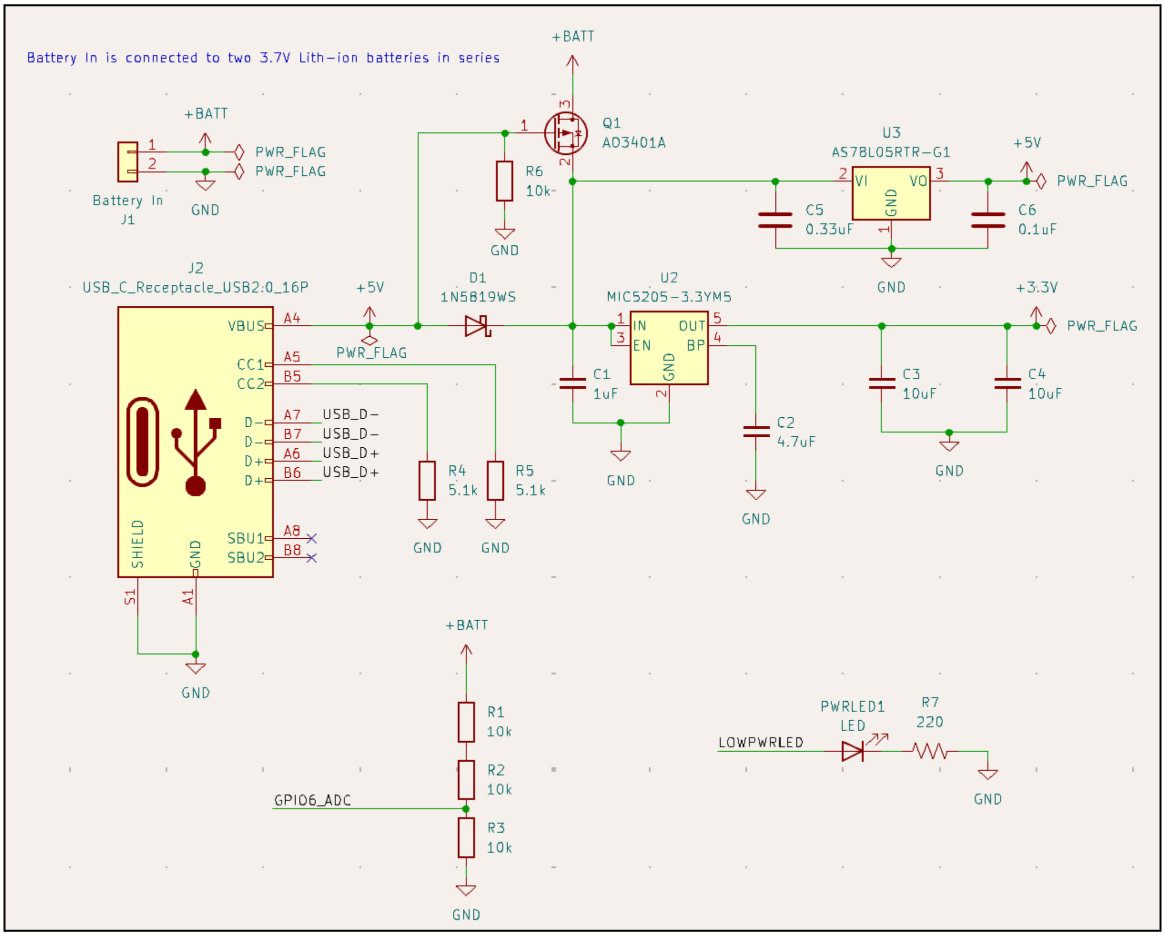

Figure 7. *Schematic of power subsystem: usb power supply and battery power supply (and components for switching between), 5V LDO voltage regulator, 3.3V LDO voltage regulator, voltage divider for battery level detection, and LED to signal low power.*

Major components of subsystem and the function they offer:

- Lithium-ion batteries: two rechargeable 3.7V batteries connected in series to act as portable power source.
- 5V LDO regulator which steps down the 7.4V to a steady 5V.
- 3.3V LDO regulator which steps down the 7.4V to a steady 3.3V.
- USB-C connector to power and/or program the board.
- Voltage divider to read the battery voltage.
- LED to signal low battery voltage, which is programmed to turn on if the ESP32 detects the voltage going below a certain amount.

Essential connections on the major components:

- The USB and battery power path consists of a schottky diode and a PMOS. If both the USB and batteries are connected, the board uses the USB as power. If just the batteries are connected, the board uses them as power.
- The USB-C connector uses the resistors connected to the CC1/CC2 pins to detect orientation. The connector's shield and ground pin are connected to ground.
- The voltage divider uses three resistors in series to determine the current level of battery voltage. One of the ESP32 ADC pins is connected above one resistor (otherwise the battery at full charge would exceed the maximum analog voltage for the pin).
- The voltage regulators receive power from either the USB-C connection or the batteries. They both use decoupling capacitors at the input and output pins, with values specified by their respective datasheets.

3. User Interface Design

Display

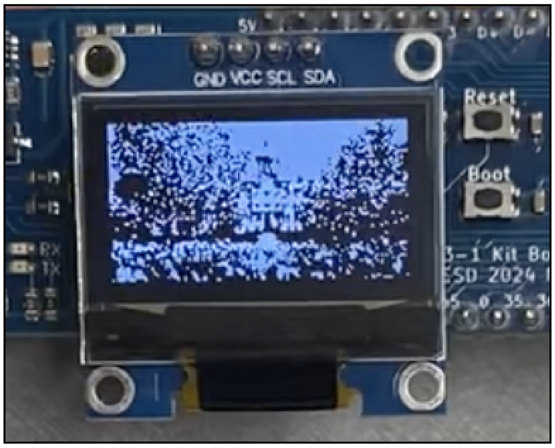

Figure 8. *User interface display test*

Major components of subsystem and the function they offer:

- The display acts as a viewfinder so that the user can see the field that they are taking a picture of.
- The display also shows the taken image so that the user can see the pictures and decide to upload or delete them.
- From testing, OLED screens update faster than LCD screens, but the available small OLED screens do not have enough resolution for this project.
- There are many available LCD screens of a good size and resolution for this project.
- The ST7789V LCD screen has a maximum speed of 62.5 MHz, and the user can still see each line being drawn one at a time at this speed, which can be seen in this video.
- A touchscreen display would enhance the user experience and limit the number of physical buttons needed. TFT LCD screens have options for either resistive or capacitive touch, but OLED touchscreens are much more limited.

Essential connections on the major components:

- The screen needs to be powered with 3.3V.
- For a SPI screen, the reset pin is on IO8 and the backlight pin is on IO7. The data and clock lines will be used for SPI interfacing.
- For a parallel TFT screen, there would be 8 different data lines on 8 different GPIO pins. 16 pins in total would likely be used for this display.

Buttons

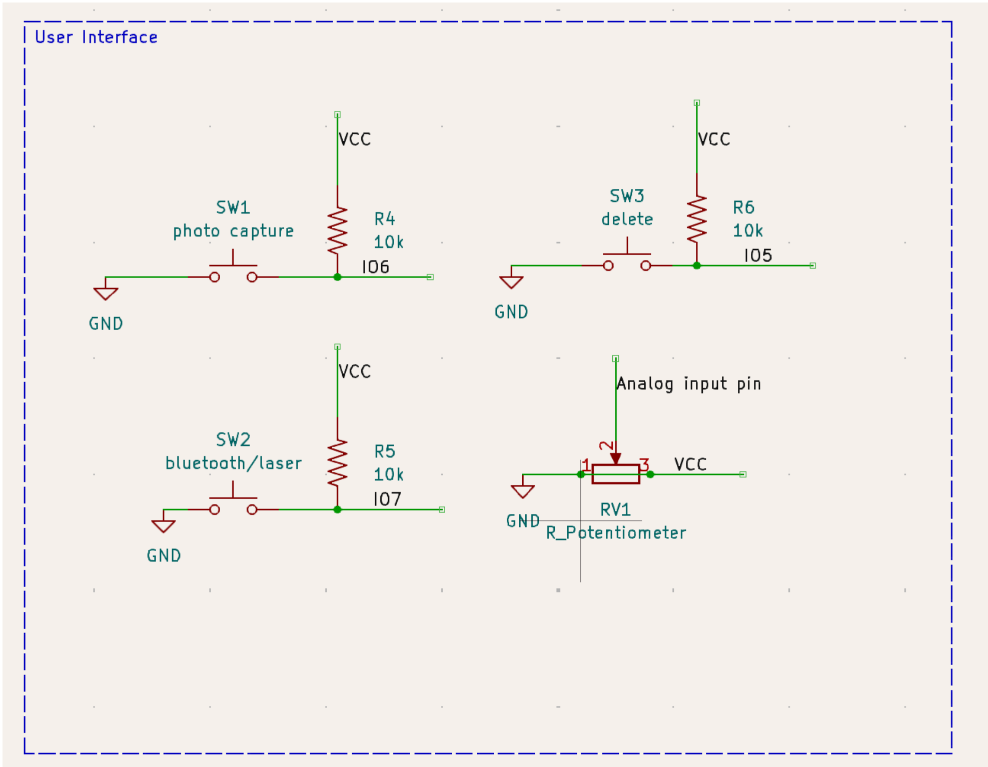

Figure 9. *Schematic of user interface: three push buttons and a potentiometer.*

Major components of subsystem and the function they offer:

- SW1 is the button for signaling the camera to take a picture.
- SW2 is the button to put the ESP32 into bluetooth-pairing mode.
- SW3 is the button to tell the ESP32 to delete the last image.
- Potentiometer is used to tell the ESP32 what PWM to use for the camera flash.
- Will also need a power on/off button to conserve battery power.
    - This also means there will need to be a way for the ESP32 to shut off the power.

Essential connections on the major components:

- Each button is connected to a pull-up resistor connected to the 3.3V logic level. Pressing a button pulls the value low.
- The potentiometer is connected to an ADC pin of the ESP32 and is powered by the 3.3V logic level.

WiFi Connection

Connecting the ESP32-S3 to WiFi:

- Use the USB-C connector to upload WiFi credentials to the ESP32-S3.
- Connect the user's phone to the ESP32-S3 via bluetooth to input WiFi credentials.

    - Espressif has developed a free app for WiFi provisioning called "ESP BLE Provisioning."
    - There are many online resources detailing how to configure the ESP32 to use the app (e.g., this online tutorial).

BT Remote

Major components of subsystem and the function they offer:

- The Bluetooth remote enables a way to trigger the Arducam Mega camera module using a wireless shutter button
- The system utilizes the S3's integrated Blutooth Low Energy Module to establish a connection with the BT remote and receive a capture command
- After Bluetooth mode is enabled by user pressing button, they can switch laser lines on, and the wireless remote button sends a BLE signal when pressed
- S3 acts as a BLE peripheral, scanning for Bluetooth remote devices
- When the recognized BT remote is detected, the S3 establishes a connection and listens for incoming commands from the remote.
- When the remote button is pressed, the remote sends a predefined BLE signal
- The S3 receives this BLE trigger signal and initiates image capture by sending a command to the Arducam mega over SPI
- Image is saved and processed the same way as when a picture is taken using the camera's local shutter button, but with a longer wait time for deleting the photo

4. Cloud / Post Processing

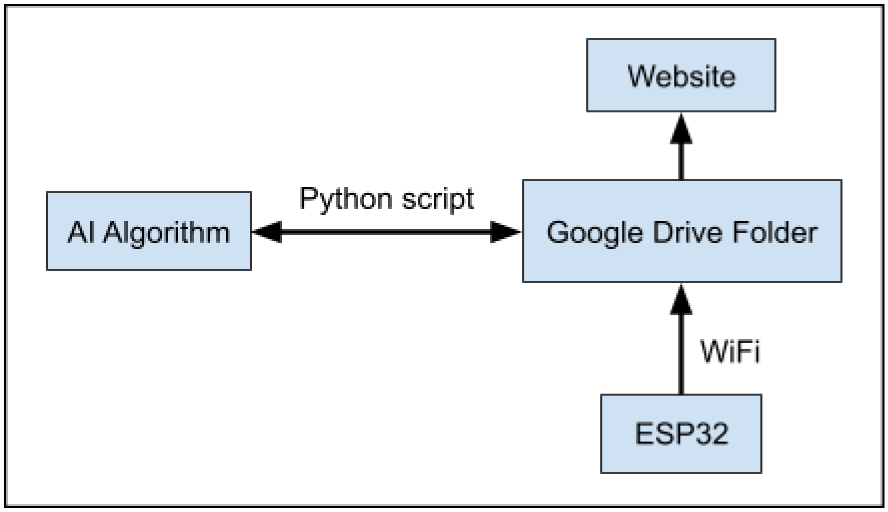

Figure 10. Workflow diagram of cloud/post processing subsystem.

Major components of subsystem and the function they offer:

- Google Drive folder to store ESP32 files in the cloud
- Website to display photos
- AI facial recognition script to identify and sort faces

Description of subsystem workflow and remaining work:

- The photos need to be uploaded from the ESP32 and then accessed by both the AI facial recognition algorithm and the website, so it makes sense to store the images somewhere in the cloud–a Google Drive folder.
- Google Apps Scripts can be used to facilitate connections to get the photos from the ESP32 to the drive and then between the drive and the Python script.
- The Python script first downloads the images and uses a dictionary to sort them. The code goes through the images to detect faces, store them in the dictionary, and then compare new faces to the stored one. The dictionary keys are the unique people while the values are the list of images that person appears in. The code currently only recognizes the dominant face in each photo, so a photo with multiple people would only be sorted into one person's list. Further work needs to be done to explore how to identify all faces in a photo. The dictionary is then used to create new folders and copy each person's photos to their folder, which are then uploaded back to drive.
- The Google Drive folder can then supply the images to a website, whether it's a google site, the senior design site, or some other page.

- The workflow would also need to be optimized so that it only analyzes new images and compares them to the existing dictionary–it should not go through every image any time a new image is uploaded.

<u>Plan for Demonstration for Design Review 2 (Working Subsystems):</u>

1. Camera Control Module and Interface Design
    a. Connect a push button to an S3 board to show that manually pushing a button triggers the Arducam to capture an image, ensuring an image is taken when it is pressed.
    b. Integrate autofocus libraries into the Arducam code and test image quality
    c. Integrate the code so that when the "take picture" button is pressed, the flash at its relative potentiometer value also goes off. If timing issues arise, the STROBE pin on the flash driver can be enabled and utilized.
    d. The laser outline is matched to the frame of the camera. Set up the laser outline and the camera. Take a picture and see if that image framing matches the laser outline in the room.
    e. Show that the GPS module is able to get the outdoor location data and display longitude, latitude, and time to the serial monitor.
    f. Show a circuit, built with a breadboard and/or protoboard, that flashes the chosen LED at a particular brightness.
    g. Show that an image taken by the camera is able to be stored to an SD card.
2. Power Design
    a. Show the rechargeable batteries stepped down to the 5V and 3.3V levels using the chosen voltage regulators. This can be done using a protoboard.
    b. Ensure that the ESP32 can read the voltage level of the batteries (or a power supply for demonstration), and turn on an LED once the voltage is low enough.
3. User Interface Design
    a. When an image is taken on the camera, it should be shown on the display. Edit the current image display code to read in an image from the camera module, not just a previously downloaded image from a laptop.
    b. Once an image can be received from the camera on the display, work on real time processing for the live view finder.
    c. Implement code for the buttons so that an image can be deleted or uploaded.
    d. Wire and code the potentiometer and the flash driver to test the flash brightness and feasibility.
    e. Test touchscreen capabilities which could be used instead of buttons for pin management and ease of use.
4. Cloud / Post Processing
    a. When an image is taken, it should be uploaded over WiFi to the Google Drive folder.
    b. Python script should organize images into a dictionary and all new images should be compared to existing files.
    c. Use Google Drive/Cloud credentials to access folders.

d. Google Drive folder should send images to the website, before and after being sorted by person.