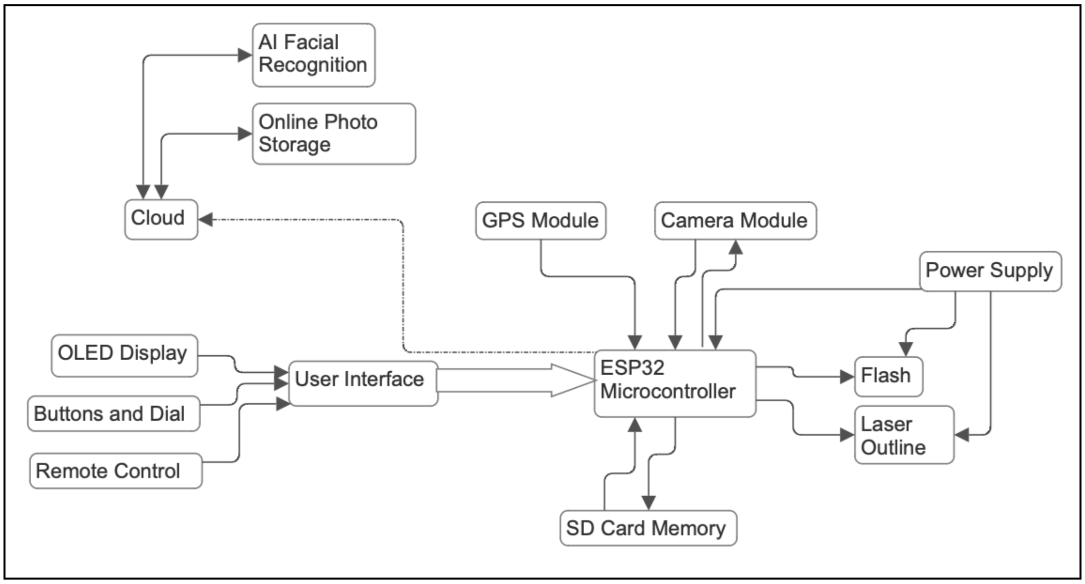
PIXEL Design Review 0
Precise Image eXtraction and Enhancement Lab

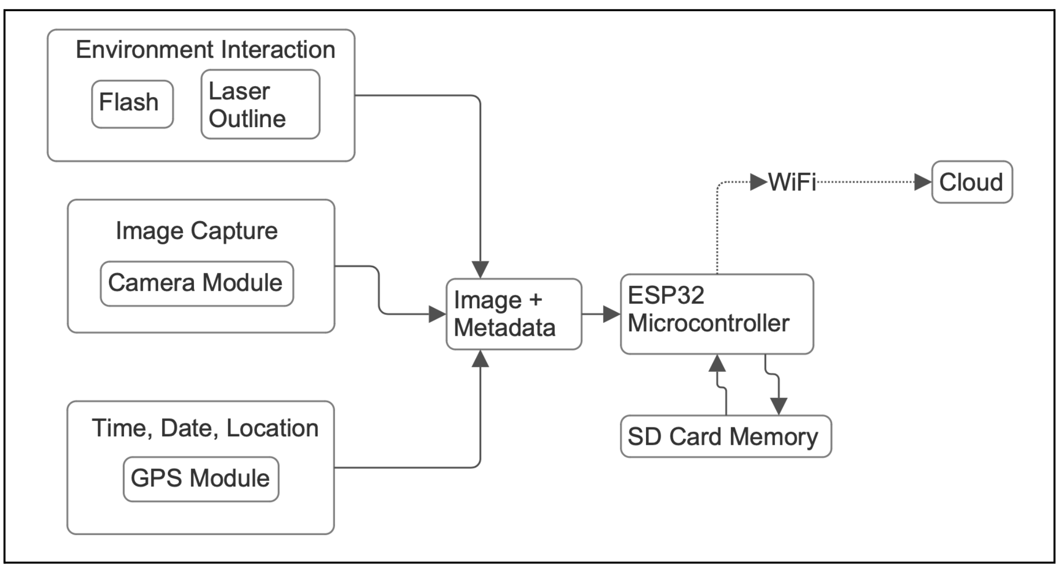
Jack Allardyce, Lindsey Canessa, Josiah Owens, Victoria Ryan, Delaney Smith
Professor Schafer
Senior Design
January 23, 2025

System Block Diagram



List of all major subsystems and requirements:

1. Camera Control Module and Interface

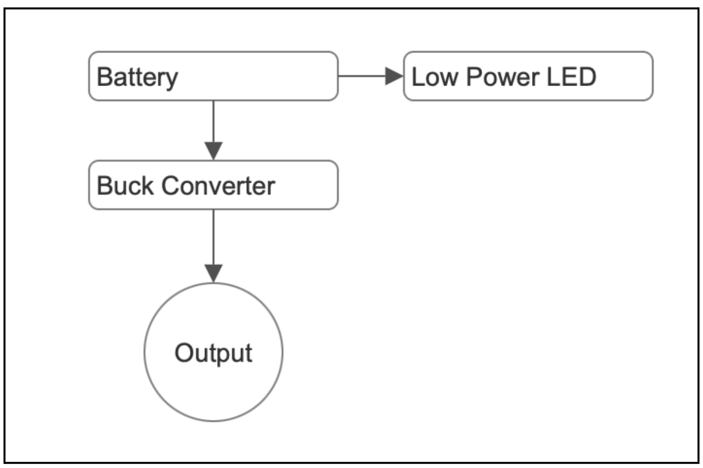


Requirements:

- The camera module must capture the photo and send it to the microcontroller via board wiring.
- The user must be able to press a button to enable the laser outline.

- The laser outline must be set to bounds of image to be taken and must be powered appropriately; the outline must be visible to the people in the self-taken image and the lasers must be class IIIA for eye safety.
- The camera flash must be adjustable via a potentiometer dial, which must be connected to ESP32 analog input pin; the flash should go off right after the user presses the button to take the image.
- The GPS device must be able to record the metadata via satellite when an image is taken; if satellite data is not available, the time, date, and location must be gathered using WiFi.
- The image and metadata must be sent to the microcontroller to upload to the cloud via WiFi; if no internet connection is available, information must be stored in the SD card memory.

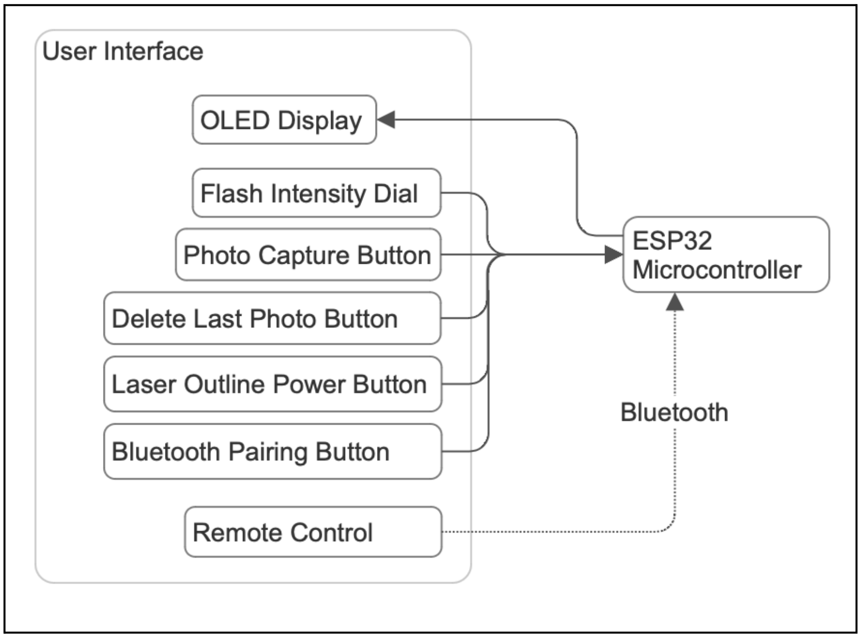
2. Power



Requirements

- Two 3.7-volt lithium ion batteries in series will be used to power the microcontroller, lasers, LEDs, and flash.
- The battery voltage must be stepped down to the desired output voltage.
- The camera battery must be able to be safely recharged and must be protected from over-discharging.
- There must be an LED to indicate when the camera battery is low.
- The camera electronics must not overheat or discharge too much heat that would be uncomfortable to the user.
- The battery life of the camera must be on par with other existing digital cameras, lasting at least 3 hours.
- The batteries must be recharged with a commercial charger.

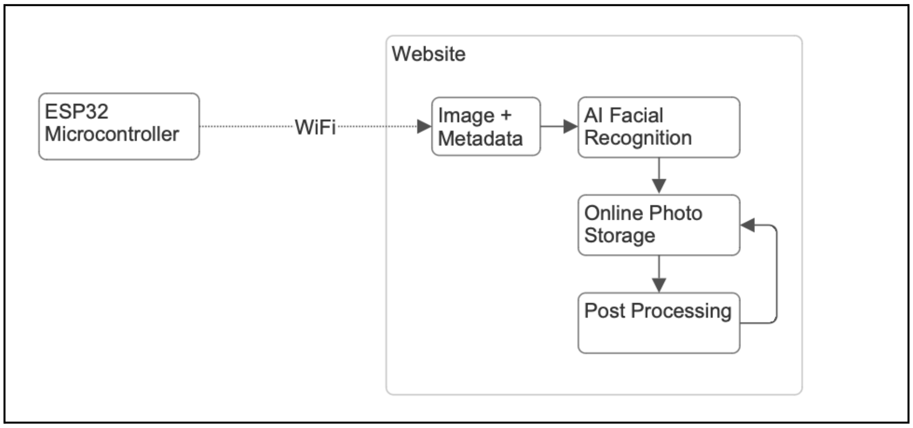
3. User Interface



Requirements:

- The OLED display must be wired to the microcontroller and should preview the image before it is taken.
- All buttons must be connected to the microcontroller, must be easily pressed, and should be labeled so the user knows the functionality of each one.
- The remote control must interface with the microcontroller via bluetooth.
- Pressing the photo capture button or the remote control must allow the user to capture a photo.
- The user must be able to press a button to enable the laser frame.
- The remote control must be able to send a signal over Bluetooth in real-time when its button is pressed.
- The camera must be able to display the correct image after capture on an OLED screen, and the pictures taken must be an accurate representation of the field of view and color that the user sees through the viewfinder.
- The camera must be able to take a picture with reasonable clarity (5 MP, 2560 x 1920 pixels) while still being within the processing capabilities of an ESP32.

4. Cloud / Post Processing



Requirements:

- The camera must be able to connect to WiFi.
- The camera must be able to send a captured image and associated metadata (location, time, and date) over WiFi to a cloud-based website.
- The pictures must accurately be received over the WiFi (correct color, field of view, image type).
- Users must be able to access the website and the interface should be user-friendly.
- The website must recognize that a given image and its metadata are connected and save that information accordingly
- The website must be able to use an AI facial recognition algorithm to group together images of the same person and must display separate albums for the people who appear in multiple photos.
- The AI facial recognition algorithm must be adequately trained so that it avoids as much bias in identification as possible.
- The website should also have an option that allows the specified users to view all of the images shared with them, even if they are not in that exact photo or the photo is not of people.

<u>Written Plan for Design Review 1</u>

<u>To achieve the requirements for design review 1, the following plan has been developed:</u>

1. **Finalize/select main components to use:**
   a. Camera module
   b. GPS module
   c. Dual LED flash driver (or equivalent flash functionality)
   d. Laser line levels
   e. ESP32 SD card module
   f. Battery circuit
2. **Analyze datasheets for each component**
   a. Review datasheets for all finalized components
   b. Identify required connections
      i. Power supply requirements (voltage, current)
      ii. Signal pins (e.g, I/O pins, communication protocols like I2C, SPI, UART)
      iii. Decoupling capacitors and resistors for stable operation
      iv. programming/debugging connections (e.g, LED for battery indicator)
3. **From those connections, look at what additional components are necessary for each subsystem (ie, resistors, capacitors, ground connections, etc.)**
   a. Verify that all necessary components are in stock in the EIH, otherwise make note of which components are not in stock and add it to the list of parts still needed to be ordered
4. **Organize Documentation and files**
   a. Create a folder for all components for PIXEL project
   b. Download and import parts libraries for main components and any additional components not included in StockLib → verify size and footprints of parts libraries which could pose potential problem for layout

5. **Define Subsystem Design - Make the Kicad schematic (does not need to be final, but helpful to see the connections) for each subsystem group:**
   a. Camera control module: include button interface, flash driver, and camera connections to ESP32
   b. Power Subsystem: include battery connections, voltage regulation, and indicators (e.g, low battery LED)
   c. User Interface: include button wiring, OLED display connections, and Bluetooth module connections
      i. Make sure the OLED can be connected to the camera module and ESP32 through I2C so that a picture taken on the camera can be processed/made compatible for display

6. **ESP32 Pin Mapping**
    a. Look at ESP32 datasheet closely to determine which I/O pins should be used for each subsystem with the goal of optimizing layout when switching to PCB design
7. **After schematic is completed, describe function of each major component and list any remaining uncertainties or subsystem problems with possible corresponding action plan to address problems:**
    a. Potential problems and mitigation solutions:
        i. Flash driver interface could be over complicated → might need to pivot to pre-designed flash specifically for a camera
        ii. Limited I/O pins on the ESP32 for all subsystems → use multiplexer or reduce number of buttons/features if necessary
        iii. Insufficient decoupling capacitors which could potentially lead to noise and instability → doublecheck datasheet recommendations for decoupling
        iv. Mismatched voltage levels between components (3.3 V and 5V) → add level-shifting components if required
        v. High-power components (LEDs, lasers) → check thermal dissipation requirements on datasheets
        vi. Components in the schematic are not in stock or have long lead times → verify component availability before including in schematic and identify alternative/backup components where possible
        vii. Multiple devices on the same bus with conflicting addresses or unsupported configurations → ensure unique addresses for I2C devices and validate protocol configurations
        viii. Overloading power rails due to underestimation of total current draw
        ix. Incorrect pin assignment for communication protocols

8. **Map out design and implementation of website**
    a. Create markup of website that includes tags to structure web pages
    b. Determine optimal way to organize photos
        i. Page for all photos
        ii. Pages for photos separated by day
        iii. Pages for photos separated by person identified
9. **Create an action plan to reduce remaining problems**
    a. Review iteratively - each team member goes through schematic for errors, each makes note of inconsistencies

<u>Written Plan for Design Review 2</u>

To achieve the requirements for design review 2, the following plan has been developed:

Steps to Demonstrate Each Subsystem Working:

<u>Camera Control Module and Interface:</u>

Initial Steps (to be completed right after Design Review 0):

1. Order camera module.
2. Order GPS module.
3. Order dual LED flash driver.
4. Order laser line levels.
5. Order ESP32 SD card module.

Component Testing:

1. Install the Arducam library. Connect the camera module to the ESP32 development board. Run the example test code from the library. Install the visualization software to a laptop to test the camera autofocus. Once the camera is seen to be working, add a push button between an ESP32 GPIO pin (with a pull-up resistor to 3.3V) and ground. Write code to read when that specific GPIO signal has been changed to LOW that will then generate a call to the camera module and its relevant pins to capture an image. Store this image in a separate file or send it serially to view and ensure its quality and accuracy. (https://how2electronics.com/interfacing-5mp-spi-camera-with-esp32-wifi-module/)
2. Wire the selected GPS module to the ESP32 with the necessary connections and stock parts. Download the related library for the decided GPS module. Run example code on the GPS module to ensure that location, time, and date information can accurately be acquired and displayed in the serial monitor.
3. Connect the flash driver to the necessary ESP32 pins (VCC, GND, SDA, SCL, etc) and to the flash LED. Properly wire a potentiometer to an analog input pin on the ESP32. Develop code that reads the brightness of the analog pin and writes that brightness to the flash driver using I2C and then lights up the LEDs (https://www.digikey.com/en/products/detail/lumileds/LHC1-3090-1205CRSP/4879077). Ensure that when the potentiometer is turned, the brightness of the flash changes. Check that when the potentiometer is turned all the one to one direction, the flash does not go off at all.
4. Test the laser line level outline creation by experimenting with different angles and setups to create a frame. Ensure that this frame is bright enough throughout the box and reasonably sized. Wire the line levels to a 3.7V battery with a breadboard for testing.
5. Connect the SD card module to the ESP32. Using the relevant directories and example code, ensure that data can be written and saved to a file. Code and Wiring Resource SD Card Module

Subsystem Testing:

1. 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.
2. Test that an image can be saved to a file on the SD card module, not just numerical data, by editing the code to not just write characters to a file.
3. Connect and write code for the laser line levels so that a push button and a development board can be used to turn on and off the lasers through user interaction. Test that the lasers all turn on and off.

Note: The plan for the cloud platform and AI facial recognition from this subsystem are discussed in the cloud/post processing section below.

Power:

Initial Steps (to be completed right after Design Review 0):

1. Order 3.7-volt lithium ion rechargeable batteries and a charger (examples linked here and here). Both products should be readily available and easily accessible to users.
2. Order buck converter (and voltage regulator if necessary).

Component Testing:

1. Design a circuit to effectively use the buck converter to power the microcontroller. The converter should get its input from the batteries and step down the voltage to the operating level of the ESP32. Depending on the buck converter that is chosen, a voltage regulator may also be needed to ensure a stable output. It is important to verify that the output from the converter is a suitable level so it does not overpower any of the components once integrated into the entire system.
2. Design a battery-monitoring system with the ESP32. Two of the ADC pins should be used to continuously check the voltages of each battery individually. To avoid overpowering the ESP32, a voltage divider must be used to scale down the voltages.
3. Develop code that turns on the red power indicator LED when the battery voltages go below 3.5 V and powers off the device once they go below 3.4 V (to prevent damage to the battery). When both are above 3.5 V, the camera should operate normally with no LED. When either is at 3.5 V, the red LED should turn on to alert the user that the batteries are close to dying. Once either of the batteries reaches 3.4 V, power should be cut off, making the device effectively “dead” so that the battery voltage does not go below the minimum safe voltage. The values used in the code will not be 3.5 and 3.4, rather they will be based on how the voltage divider scales the voltages.

Subsystem testing:

1. Integrate the batteries and buck converter circuit with the ESP32. Test that the buck converter effectively steps down the voltage to the level required by the ESP32. Also check to make sure the ESP32 is monitoring the voltage levels from both batteries individually. This can be done by writing code that continuously prints the output from each battery to the serial monitor.

2. Test what happens as the batteries discharge. When one of the batteries hits 3.5 V, ensure that the red LED turns on. As the batteries continue to discharge, make sure that power is turned off when 3.4 V is reached by either battery.

User Interface

Initial Steps (to be completed right after Design Review 0):

1. Order the OLED display.
2. Order bluetooth remote control.

Component Testing:

1. Develop code for OLED display testing. Download the necessary libraries and define an accurate pixel size and outline. Use example code test functions to ensure that the entire OLED screen is functioning as expected.
2. Test the delete last photo button by connecting the button to a GPIO pin of the ESP32 so that a HIGH or LOW signal can be sent to the ESP32. Write code that receives this signal and erases the previous data set when the button is pressed.
3. Develop code that allows the remote control to connect to the ESP32 using Bluetooth. Use the Bluetooth library and pair the device. Test that a signal can be received from the remote control by the ESP32.
   Note: Flash intensity dial, photo capture button, and laser outline power button are described in the camera control section.

Subsystem Testing:

1. Test that the ESP32 can display live information on the OLED, similar to how it will need to display the live camera feed as the viewfinder when the subsystems are integrated.
2. Integrate the laser line level button code and the remote control button code to ensure that when the remote control is connected, the frame will automatically display to be used in self-timing instances.
3. Test the OLED's functionality with the ESP32 (resolution, pixel locations, shape) by reading in an image file to the ESP32 and displaying it to the OLED screen. Ensure that the image is accurate and not cropped.

Cloud/Post Processing:

Initial Steps (to be completed right after Design Review 0):

1. Research website creation.
2. Research AI facial recognition algorithms.

Component Testing:

1. Build a test website. Connect the ESP32 to this website using correct code to initialize and utilize a WebSocket server. Send information, likely the GPS data, over the WebSocket and display it on the website page. Ensure that the data transfer is fast enough and accurate. Once numerical data can be sent successfully, explore the sending of an

image or sending pixel values as numerical data and converting back into an image at the website.

2. If a website is not working, write MATLAB code to use the MATLAB WiFi interface to establish a connection to the ESP32. Read data from the ESP32 into the MATLAB code, starting with numerical data. Then, test sending image data or pixel values and converting to a workable image in MATLAB.
3. If the initial testing of the website was successful, develop AI facial recognition code that can be tested on an existing data set. OpenCV projects or Edge Impulse with the Arducam module Edge Impulse with an Arducam project code can be referenced and tested to determine the accuracy of facial recognition results.
4. If the MATLAB route was taken, utilize MATLAB AI and facial recognition libraries to test facial recognition algorithms with existing databases.

Subsystem Testing:

1. If using a website, test that an image can be clearly and accurately displayed on the website screen. Create different tabs or links for different people that can be accessed by the facial recognition algorithm.
2. If MATLAB is being used, develop a GUI that can display the image that is read in on its screen. Create tabs or pages that can display different images that can correspond to different people.

After working with each component and subsystem, identify any remaining uncertainties, difficulties, and failures. If an original plan is not functioning as expected, a different communication protocol or user interaction type can be explored and tested. Then, if a certain component still will not work and is realized to be an inadequate choice, the functionality will be researched again and an alternate component or solution will be proposed. A main unknown at the moment is the details for the website and AI facial recognition algorithm, so multiple different options will likely need to be explored. Similarly, the specific GPIO pins that will be available for each component are not yet known. The interactions between the subsystems still still need to be tested to ensure that each button press and function will provide all of the expected results.

Steps to provide a preliminary board design:

1. Make updates to schematic from Design Review 1.
2. Design an approximate layout for the board based on the schematic. Keep in mind the overall camera size and user experience.
    a. The push buttons should be neatly aligned along the side of the OLED, one push button should be externally wired to the top of the camera case, the header pins should not get in the way of the case, etc.
    b. The ESP32 S3 antenna port location should be accessible.
    c. The space within the board outline should be effectively utilized.

3. Route the connections for each component. Ensure efficient trace lengths and widths are used.
4. Label all relevant signals and pins.
5. Add mounting holes and fiducials.

The specific number of LEDs and their locations is still not known. While it is known that a power LED, a low battery LED, and a WiFi connection LED are desired, additional LEDs may be added to indicate instances such as a Bluetooth connection or that an image was successfully sent.

Availability Verification

- Microcontroller
  - ESP32-S3-WROOM-1U-N8R8 from DigiKey; can ship immediately
- Camera LED Flash Driver
  - MP3336AGC-Z from DigiKey; can ship immediately
  - AL3644CH12-7 from DigiKey; can ship immediately
- Battery
  - 18650 Rechargeable 9900 mAh 3.7V Li-ion Battery from Amazon; shipping time is ~1.5 weeks.
  - Charger for the above battery from Amazon; two-day shipping
- Bluetooth Remote
  - Remote from Amazon; two-day shipping
- GPS Module
  - GPS NEO-6m from Amazon; two-day shipping
- Camera Module
  - Mega 5MP SPI Camera Module from ArduCam; standard shipping is 2-4 weeks and express shipping is 4-7 business days
- Laser Lines
  - Red 5mW laser line (2 pack) from Amazon; prime shipping
  - Green 5mW laser line from Amazon; prime shipping
- OLED Display