REAL TIME SYSTEM AND INTERNET OF THINGS FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

CaptureMe

GROUP 14

STEFANUS SIMON RILANDO	**2206830422**
KEVIN NAUFAL ARYANTO	**2206062850**
GIOVAN CHRISTOFFEL SIHOMBING	**2206059566**
NICHOLAS SAMOSIR	**2206059396**

PREFACE

Security Cameras such as CCTV have been around for environment surveillance and provide a sense of security by capturing video in real-time. However, many surveillance systems are still dependent on manual monitoring or in need of human data analysis. These cameras are used to capture events, but it does not have the ability to analyze or detect a person's location automatically in precision timing.

CaptureMe introduces a breakthrough, by applying Artificial Intelligence and Internet of Things, CaptureMe is able to process data from the camera in real time to detect people within camera range. Initially, CaptureMe was originally developed to detect "RFS" before class lecture began. However, after time, we see the opportunity for our product in the security market. This system can send information about the number of people detected and display the image via application.

CaptureMe's goal is to create a smarter and more efficient detection system, enabling faster and more responsive surveillance. Using AI and IoT technology, CaptureMe provides a more automated and effective monitoring solution, reducing human error and making it easier to manage data generated by cameras

Depok, December 13, 2024

Group 14

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

In general Surveillance uses CCTV cameras only for capturing videos and requires manual monitoring or capture analysis for movement detection or important events. This can cause delay on real time events that potentially could risk security or reduce the effect of surveillance.+

In addition, traditional surveillance cameras are not equipped with the ability to automatically detect people's presence, which makes it inefficient in an environment that needs very secure supervision. Usually surveillance systems use operators for record monitoring, increasing the possibility of error and slowing down response if an unusual event happened.

Notification for surveillance systems is still manual, which means users cannot get real-time information about the movements of people that are detected by the camera. The lack of immediate access to information limits the user's ability to respond as soon as possible if facing potential threat or events that need intensive care.

1.2 PROPOSED SOLUTION

CaptureMe utilizes many hardware and software to implement an automated surveillance system. ESP32-CAM AI Thinker is used as the main camera equipped with a wifi module to capture videos and detect the presence of people using AI-based object recognition technology. Detection results will be sent to ESP32-S3 N18E8 which is used for data processing and server connection. Server is tasked to receive data from ESP32-S3 for further processing and sends real-time notifications to Blynk, allowing users to receive information of how many people that are detected and display the face of each person detected.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. Ability To Count Passerby

   The CaptureMe system must be able to automatically count the number of people passing through the monitored area. The system must be able to identify and count detected people without requiring manual intervention, even in varying lighting conditions.

2. Capture Passerby Face Images

   Every person detected must be automatically captured by the camera. The face images of people passing through the monitored area must be stored for further reference, with image quality sufficient to recognize the individual.

3. Real-Time Notification

   The system must send real-time notifications to the user every time a person is detected, along with the person's face image. This notification will be received by Blynk.

4. WiFi Connection and Remote Access

   The system must have a stable WiFi connectivity to ensure continuous data and image delivery. Users must be able to access and control the system remotely through the CaptureMe application or related platform.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person

Project Leader	Lead the project	Giovan Christoffel Sihombing
Staff	Help the project leader	Nicholas Samosir
Staff	Help the project leader	Stefanus Simon Rilando
Staff	Help the project leader	Kevin Naufal Aryanto

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

	November						Desember											
Task Title	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12
Hardware Design Completion																		
Software Development																		
Integration and Testing of Hardware and Software																		
Final Product Assembly and Testing																		

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

The hardware schematic of the CaptureMe system outlines the interaction between key components, ensuring efficient data flow and processing. The ESP32-CAM AI Thinker serves as the primary image capture and detection module. It captures images and identifies individuals, transmitting the data to the cloud. From there, the data is processed and sent to the ESP32-S3 N16R8, which acts as the central processing unit, handling the data and managing communication with other components.

Once the ESP32-S3 processes the data, it forwards relevant information to a database for storage, including detection results and captured images. This database serves as a repository for all data, ensuring it is accessible for further analysis or retrieval. Finally, the processed data is sent to the Blynk application, providing real-time notifications to users. This system architecture ensures a smooth and reliable operation, offering instant updates and facilitating easy interaction with the captured data.

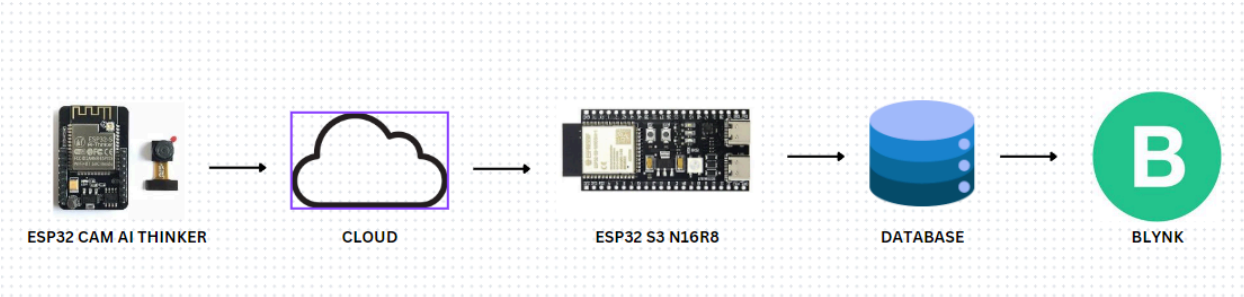


Fig 2.1.1. Schematic.

2.2 SOFTWARE DEVELOPMENT

This software development project focuses on building a system that integrates the ESP32 microcontroller with the AI Thinker ESP32-CAM camera module to capture and transmit

images over a Wi-Fi network. The system enables remote viewing of images through a web interface, offering functionalities such as image capture, streaming, and camera control, including the ability to turn the camera's flash on or off. The development was carried out using the Arduino framework for the ESP32, which allowed for seamless integration with both the camera module and the Wi-Fi features.

The primary objective of this project was to create a camera system that can connect to a Wi-Fi network and provide real-time image viewing via a web server. This system was designed to be both simple and adaptable, serving as a platform for additional features such as video streaming or motion detection. The ESP32 establishes a connection with the Wi-Fi network using specific credentials, and once the connection is successful, the camera is initialized, and the web server is activated. The server then serves images from the camera, and the camera's settings, such as brightness or image flip, can be adjusted.

In terms of hardware, the ESP32 microcontroller handles both the camera module and the Wi-Fi communication. The camera uses the OV2640 sensor to capture images, which are processed and served by the ESP32. The ESP32 communicates with the camera through multiple GPIO pins, including those for data, clock, and control signals, as defined in the camera_pins.h file. These settings ensure proper interaction between the microcontroller and the camera, allowing it to capture images at the configured resolution and frame rate.

The software is organized into various sections, starting with the Wi-Fi connection setup using the predefined SSID and password. After a successful connection, the camera is initialized using the camera_config_t structure, which specifies the pixel format (JPEG) and frame resolution. The camera sensor is then adjusted for settings such as brightness, saturation, and vertical image flip. A basic HTTP server is launched, and the ESP32’s IP address is displayed through the Serial Monitor, allowing users to connect to the camera's web interface via a browser.

Flash control is an additional feature in this system, with the flash being managed through GPIO pin 4. The flash can be toggled using the controlFlash() function, which ensures that the flash is off initially when the system powers up. The loop() function continuously

operates and can be modified in future versions to include tasks such as scheduled image captures or handling additional requests from the web interface.

The development and debugging process was supported by outputting key data to the Serial Monitor, such as information on camera initialization, Wi-Fi connection status, and flash control. The web interface allows users to view the captured images by entering the camera's IP address into a web browser. The flash control function has been verified by toggling the flash state and observing changes in the lighting conditions captured by the camera.

There is room for further development in the system, such as adding motion detection, live video streaming, or user authentication for the web interface. Additionally, error handling can be strengthened to address potential issues like camera initialization failures or Wi-Fi connectivity problems. The web interface could also be enhanced to offer users more control over camera settings, such as resolution, frame rate, and image quality.

2.3 HARDWARE AND SOFTWARE INTEGRATION

The ESP32-based camera system integrates both hardware and software components to deliver a functional solution for capturing and streaming images over a Wi-Fi network. The hardware setup involves the ESP32 microcontroller, which serves as the central processing unit, and a compatible camera module. The camera's functionality is governed by the correct configuration of GPIO pins, which vary depending on the model of the camera being used. The software handles tasks such as camera initialization, Wi-Fi connection management, image capturing, and streaming. The integration of these components is key to enabling a smooth operation where the ESP32 controls the camera to capture images and transmits them over Wi-Fi.

The camera pin configuration is one of the most critical aspects of hardware integration, as it ensures proper communication between the ESP32 and the camera module. The preprocessor directives in the code define GPIO pins for several camera models, including the WROVER KIT, ESP_EYE, M5STACK_PSRAM, M5STACK_V2_PSRAM, M5STACK_WIDE, AI_THINKER, and TTGO_T_JOURNAL. These models have distinct pin assignments for

functions like power control (PWDN_GPIO_NUM), reset (RESET_GPIO_NUM), external clock synchronization (XCLK_GPIO_NUM), and data transfer (SIOD_GPIO_NUM, SIOC_GPIO_NUM). The camera's image data output pins (Y2_GPIO_NUM to Y9_GPIO_NUM), along with synchronization pins for vertical and horizontal sync (VSYNC_GPIO_NUM, HREF_GPIO_NUM), and the pixel clock (PCLK_GPIO_NUM), are also defined. This configuration ensures that the camera's various components are correctly wired to the ESP32 for seamless communication and image capture.

On the software side, the system initializes the camera with key settings such as pixel format (JPEG), resolution, and frame rate. The configuration dynamically adapts based on the availability of PSRAM, selecting higher resolutions and lower compression for models that support it. For instance, when PSRAM is detected, the resolution is set to UXGA (1600x1200), while models without PSRAM default to SVGA (800x600) to optimize performance. The sensor settings, including brightness and saturation, are adjusted to ensure optimal image quality, especially when using sensors like the OV3660, which requires specific tuning for flip and saturation control.

Wi-Fi connectivity is another integral part of the system, allowing for remote access to the camera stream. The software connects the ESP32 to a Wi-Fi network using the SSID and password provided, and once the connection is established, it prints the device's local IP address, which can be used to view the live camera feed in a web browser. Additionally, a server is started to stream the camera feed over the network. Flash control is another key feature in this setup, enabling the flash to be turned on or off depending on the conditions, such as low-light environments.

The main program loop introduces a delay to manage system load and power consumption, ensuring that the camera remains active without overloading the server. The main code doesn't handle frequent data processing but instead focuses on maintaining a stable environment for image capture and transmission. The server continuously streams the video feed to connected devices, which can view the images in real-time.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

To ensure the functionality and reliability of the CaptureMe system, testing was conducted on its core components. The testing focused on verifying the integration of hardware and software, specifically the ESP32-S3 and ESP32-CAM modules, and their respective functionalities.

The ESP32-S3 was tested to ensure its functionality as the central processor for data handling and communication. The first test involved verifying its ability to receive data from the ESP32-CAM module, ensuring smooth data transfer over both GPIO connections and the Wi-Fi network. The second test aimed to assess the ESP32-S3's capability to process incoming data and forward it to the server without delays, ensuring smooth and consistent operation.

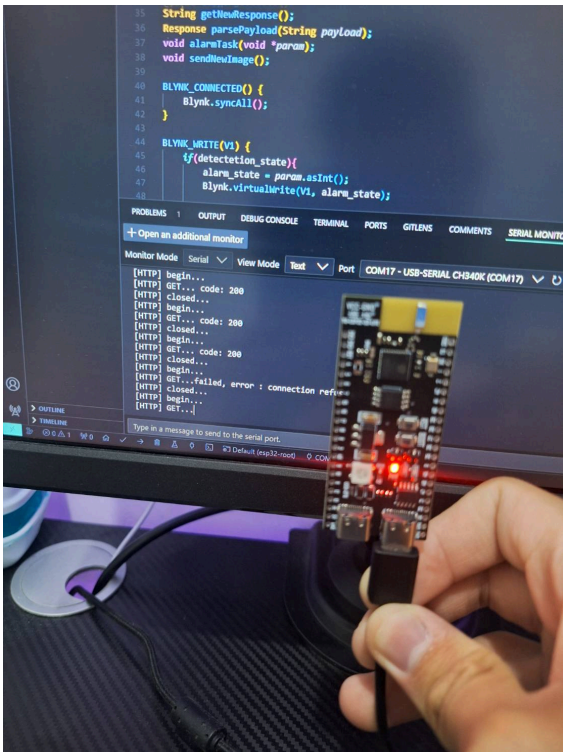


Fig 3.1.1 ESP32 S3 Testing Result

Lastly, the module was tested for its real-time notification capability. This test was to confirm that notifications containing detection data and captured images are sent promptly to the Blynk application, demonstrating its ability to support immediate responses under varying Wi-Fi conditions.

The ESP32-CAM module underwent testing to evaluate its performance in image capture and detection. The first test focused on image quality under different lighting conditions to ensure clarity and sharpness, including its flash functionality for low-light environments. The module's face detection capabilities were tested using AI-based object recognition technology, ensuring it accurately identified and captured individuals while minimizing false detections.

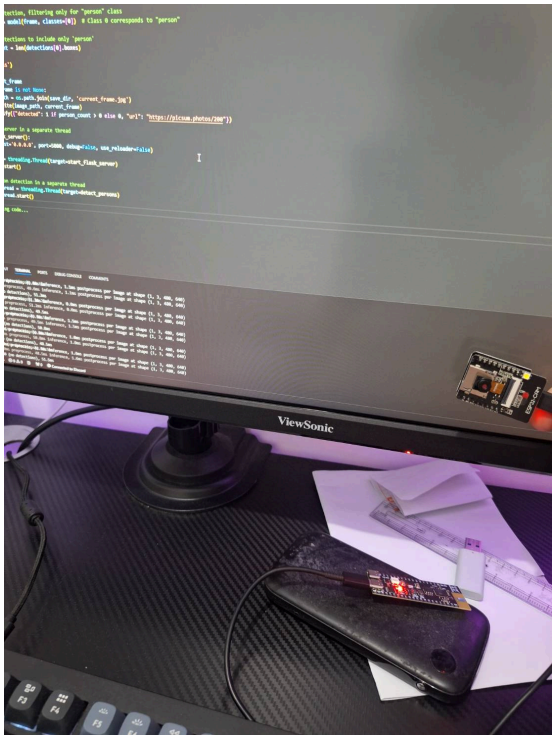


Fig 3.1.2 ESP32 Cam Sensor Test

Additional tests involved assessing the streaming functionality of the camera, ensuring that a stable video feed could be accessed via a web interface. The ability for users to adjust settings such as brightness and resolution was also tested to optimize performance. Finally, flash control tests were conducted to verify the camera's response to flash commands for use in dark environments.

3.2 RESULT

The results of the project demonstrate the successful integration of the ESP32-S3 and ESP32-CAM modules with the Blynk platform, providing real-time monitoring and notification capabilities. The Blynk Console effectively displays a counter for person detection, incrementing each time the ESP32-CAM identifies a person. This feature ensures accurate tracking of detected events for analysis and optimization. Additionally, the console showcases a list of captured images from the ESP32-CAM, allowing users to visually verify detection accuracy. Alongside these features, an alarm status switch is available, enabling users to remotely toggle the alarm on or off, enhancing system flexibility and user convenience. The system's online status is also clearly displayed, confirming the active connection of the ESP32-S3 module. This seamless integration highlights the effective functionality of the hardware and software components, ensuring reliable performance and a user-friendly monitoring experience.

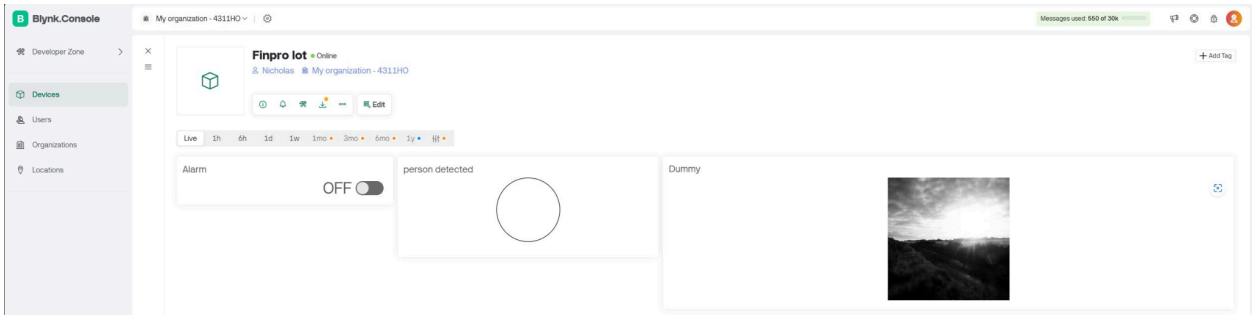


Fig 3.2.1 Blynx Result

3.3 EVALUATION

The evaluation of the CaptureMe project centers on the effectiveness and reliability of its components in delivering the expected results. The integration of the ESP32-S3 and ESP32-CAM modules serves as the foundation for the system's core functionality, namely person detection and image capture. The ESP32-CAM's ability to capture high-quality images is a key advantage, as it ensures that the images of detected individuals are clear and identifiable. This is crucial for real-time monitoring through the Blynk platform, where users can track detections and view captured images. The person detection counter accurately registers each detection event, providing real-time feedback on the system's activity. The integration with the Blynk Console allows users to monitor the status of the system, including the alarm function, which provides immediate notifications whenever a person is detected. This feature enhances the system's responsiveness and usefulness in practical applications, such as security monitoring.

Furthermore, the system's overall reliability is bolstered by the seamless interaction between the hardware components and the Blynk platform. The ESP32-S3's performance in handling the processing of detection data and image transmission to the Blynk server is commendable, as it operates efficiently without significant delays. The Blynk Console's user interface offers a simple yet effective way to manage and interact with the system, displaying real-time data and images of detected individuals. The captured images, displayed as part of the system's list, allow users to easily review the history of detections, making it easier to verify events and monitor activity. The evaluation confirms that the system's design and the components used are not only functional but also practical for a wide range of real-time monitoring applications, particularly in areas requiring surveillance or security. The system performs as intended, with high accuracy and reliability, making it a valuable solution for monitoring and security purposes.

CHAPTER 4

CONCLUSION

The CaptureMe project represents a cutting-edge solution for modern surveillance systems, combining the power of AI and IoT to enhance efficiency and functionality. Using the ESP32-CAM AI Thinker and ESP32-S3, the system is capable of real-time detection, counting passerby, capturing face images, and delivering instant notifications, providing users with a comprehensive monitoring tool.

The integration with the Blynk platform allows for remote access and control, ensuring users can monitor activities and receive notifications in real-time through a user-friendly application. The system's robust WiFi connectivity ensures uninterrupted data transmission and reliable performance.

The modular software design ensures efficient operation, scalability, and adaptability, enabling smooth integration of hardware and software components. Features like face capture, real-time notifications, and automated counting provide a practical, automated, and error-reducing solution for surveillance needs.

CaptureMe successfully fulfills its acceptance criteria, including the ability to detect and count individuals, capture face images, and provide instant notifications. This project highlights the potential of integrating advanced technologies to create an automated, intelligent, and user-friendly surveillance system, setting a new standard in the realm of security and monitoring.

REFERENCES

[1] A. Germanov, “How to Detect Objects in Images Using the YOLOv8 Neural Network,” freeCodeCamp.org, May 04, 2023. https://www.freecodecamp.org/news/how-to-detect-objects-in-images-using-yolov8/ (accessed Dec. 10, 2024).

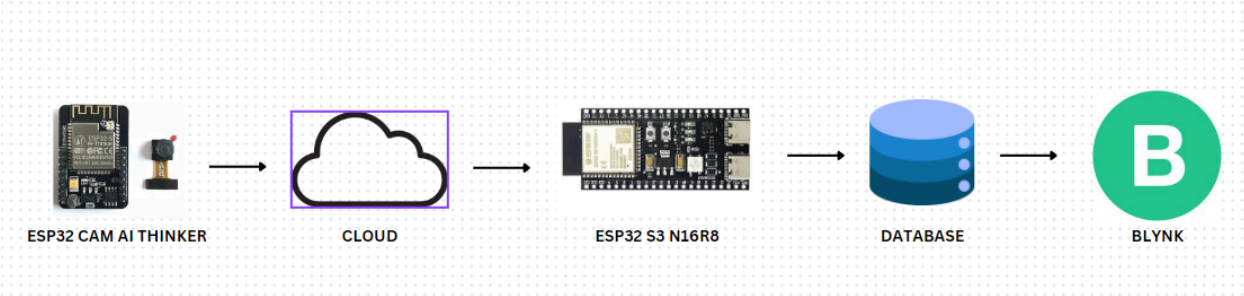
[2] Arduino Forum, “How to connect ESP32 to flask python via SOCKETIO,” Arduino Forum, Oct. 25, 2023. https://forum.arduino.cc/t/how-to-connect-esp32-to-flask-python-via-socketio/1182093 (accessed Dec. 10, 2024).

[3] Nannigalaxy, “GitHub - Nannigalaxy/esp32-cam_flask: A simple flask server for esp32-cam to upload captured image.,” GitHub, 2020. https://github.com/Nannigalaxy/esp32-cam_flask (accessed Dec. 10, 2024).

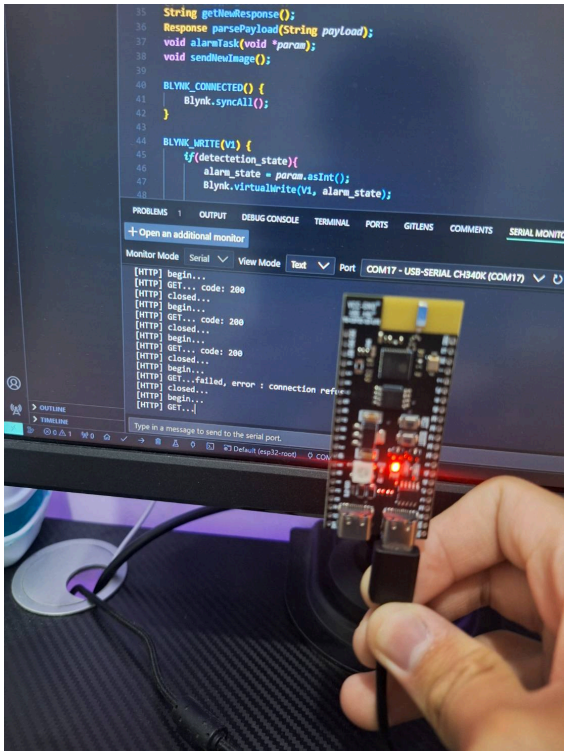
[4] Blynk, “Connecting Blynk and ESP32,” Blynk Community, Mar. 27, 2024. https://community.blynk.cc/t/connecting-blynk-and-esp32/69891 (accessed Dec. 10, 2024).

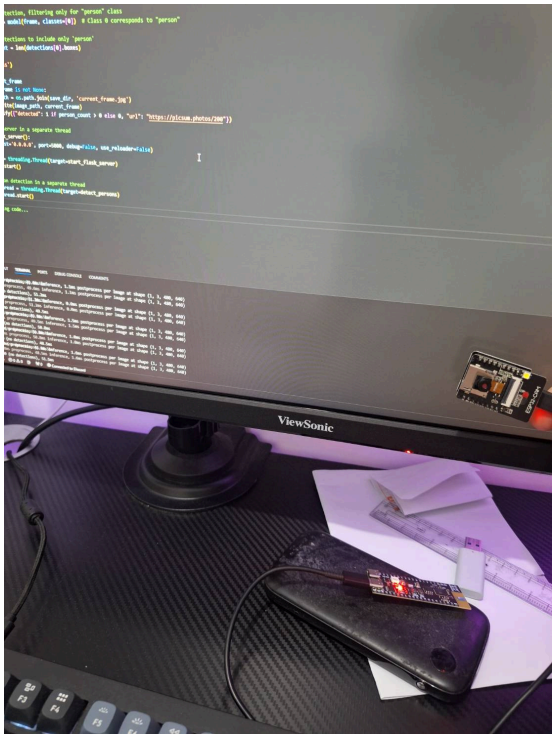
APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation




ViewSonic