Embedded System Design - Mini Project Final Report

Remote Object Targeting System

Jon Dromey[a], **Soroosh Noorzad**[b] **and Zixuan Wang**[c]

[a] *B.Sc. at the Mechanical Engineering Department*
[b] *Ph.D. at the Electrical and Computer Engineering Department*
[c] *M.Sc. at the Electrical and Computer Engineering Department*

Supervisor: Pierre-Emmanuel Gaillardon

1. Project Description and Final Goal

This project aims to develop a real-time remote object targeting system using a camera-based object detection mechanism and a laser pointer. The system identifies an object based on color, determines its coordinates, and adjusts a laser to point precisely toward it.

For example, it should recognize a blue ball in a space for which it is calibrated. It should be able to calculate the object's position in terms of direction and distance from the sensor. When the object's position has been calculated, it should be able to command actuators to enable the system to point a laser at the object.

Also, it should be fast enough to track the laser on the object as it moves through space. When no object is detected, it could go to a resting position and turn off the laser.

The system involves two microcontroller units and a central server:

- **Space Monitoring Camera**: A camera that takes pictures of space continuously and sends them wirelessly to a remote server for image processing.
- **Laser Targeting Unit**: Receives the object's coordinates and precisely adjusts a laser pointer to aim at the detected object.
- **Central Server**: Acts as an intermediary among units, processing images, managing data exchange, processing requests, ensuring accurate synchronization, and monitoring system status. It receives the pictures from the camera, performs the required processes, and sends the results to the remote actuators.

The project integrates real-time image processing, wireless communication, and embedded control to create a responsive and automated targeting system suitable for surveillance, tracking, or industrial automation applications.

2. Original Proposal Overview

As outlined in our initial proposal, the project consisted of eight milestones:

1. Motion detection with an indicator LED.
2. Servo-controlled laser pointer directed by detected motion.
3. UsART Wired communication between the camera and the actuator units.
4. Direct wireless communication between units.
5. Wireless communication via an online MQTT server.
6. Wireless communication via a local server.
7. Detecting a successful laser hit.
8. Implementing a local database and monitoring interface.

The block diagram of the whole system was drawn as Figure 1.

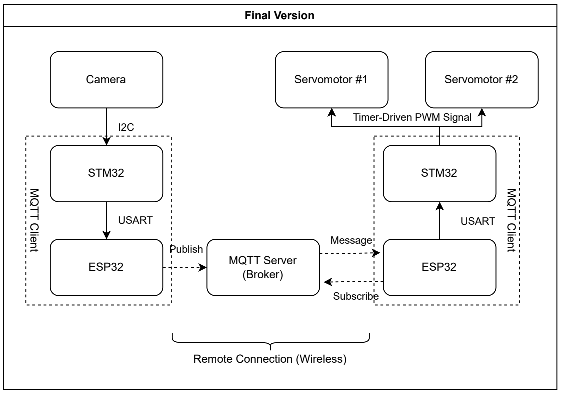

Figure 1. Block diagram of the proposed implementation

3. Development and Implementation

The initial phase of the project focused on establishing the actuator subsystem. We successfully implemented a servo control system driven by an STM32 microcontroller, allowing precise control of a laser pointer's orientation. After ensuring the reliability of this subsystem, attention turned to the object detection portion. Following a technical review, it was determined that processing image data directly on the STM32 would be impractical. Consequently, an ESP32 platform was selected to handle image acquisition and preliminary processing, leveraging its computational advantages and built-in wireless capabilities.

The next development phase involved configuring the ESP32-CAM to operate as an image server. We implemented a streaming system over Wi-Fi, allowing client devices to retrieve frames dynamically. Enhancements such as an AJAX-based refresh mechanism were introduced, improving user experience by enabling seamless image updates without requiring full page reloads. While object detection was not fully implemented at this stage, the foundational communication and streaming infrastructure was operational and reliable.

In subsequent development, efforts shifted towards implementing image processing capabilities on the ESP32-CAM. Initial experiments with background subtraction techniques encountered severe memory limitations, even at reduced image resolutions and grayscale formats. To address these challenges, processing tasks were offloaded to an external server. Although this server-based background subtraction successfully isolated objects, the solution introduced unacceptable latency for real-time control. Recognizing this, we pivoted to a brightness-thresholding method that required only single-frame analysis. This approach dramatically reduced computational overhead, enabling the ESP32-CAM to locally detect objects based on pixel brightness deviations from an assumed white background. Detected object centers were transmitted over UART and later via MQTT protocols, establishing a robust communication pipeline from the vision system to the actuator controller.

To further enhance system performance, we transitioned processing entirely to the laptop side by streaming JPEG images to a Flask server. Using OpenCV, we performed real-time color-based object detection, targeting a specific blue object. Socket-based communication replaced HTTP methods to reduce transmission delays, improving responsiveness. The Flask server simultaneously visualized results via a live web interface and published detected object coordinates through MQTT topics. These coordinates were subscribed to by an ESP32 device and relayed to the STM32 over UART, enabling real-time actuator adjustments.

Throughout the project, practical calibration challenges emerged. Due to physical misalignments between the camera perspective and laser pointer positioning, manual calibration steps were necessary each time the system was deployed in a new environment. Although we considered implementing a planar homography-based transformation for automatic calibration, time constraints prevented the full realization of this feature.

Overall, the project successfully completed milestones involving actuator setup, wireless image streaming, server-based processing, and servo control integration. We achieved the majority of the originally proposed functionalities, with only the final laser-hit detection module left as a future enhancement.

4. Final Achievements

Out of the eight originally proposed milestones:

- We completed milestones 1 through 6 and partially completed milestone 8 (implementing a local database).
- Milestone 7 (detecting successful laser hits) was not implemented, but could have been completed with minor additional work.

The final block diagram of the whole project can be seen in the figure 2.

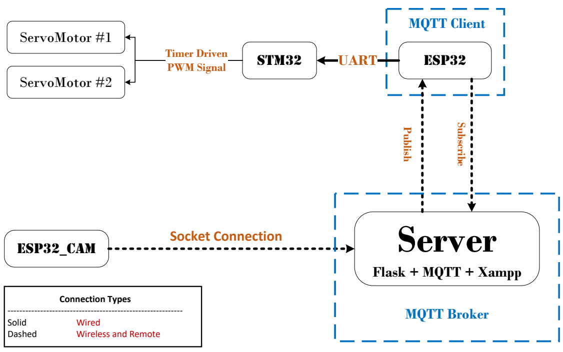

Figure 2. Block diagram of the final implementation

Technologies covered include:

- Real-time embedded control (STM32 timers, UART, interrupts)
- Wireless networking (ESP32 Wi-Fi, MQTT)
- Computer vision (OpenCV-based color tracking)
- Web technologies (Flask, XAMPP, AJAX, MySQL)

5. Conclusion

This project successfully demonstrated the integration of embedded systems, wireless networking, and real-time computer vision. Despite facing hardware limitations and calibration challenges, we created a functional laser-guided object tracking system.

The experience provided deep insight into hardware/software co-design and advanced our practical skills in embedded system development beyond standard laboratory exercises.

The final project implementation has been drawn in Figure 3.

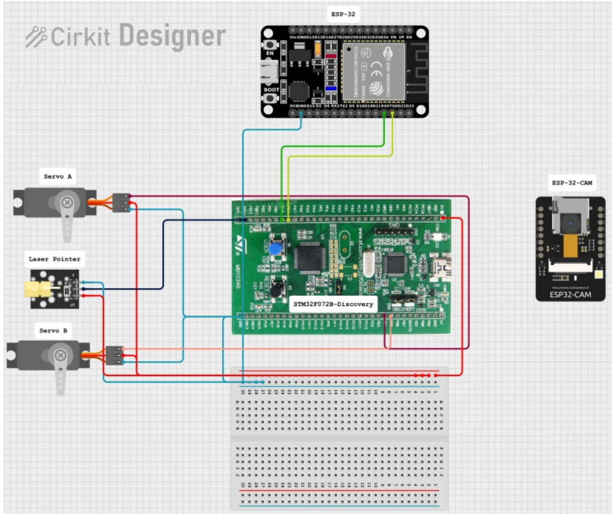

Figure 3. The final schematic of the whole project implementation