



**Boston University
Electrical & Computer Engineering
EC464 Senior Design Project**

Second Prototype Testing Plan

Remote Vision Robotics (Eye-Robot)



by

Team 25
Remote Vision Robotics

Team Members:

Purvis (Naith) Amin, pmamin4@bu.edu

Nicholas Bart, nbart@bu.edu

Nikhil Gupta, nikhilgu@bu.edu

Jaden Tayag, jadtay@bu.edu

Ryan Verdile, rverdile@bu.edu

Required Materials

Hardware:

- Oculus Rift S
- Raspberry Pi 4 2GB
- Raspberry pi V2 Camera
- 1x Stepper Motor - NEMA-17 size - 200 steps/rev, 12V, 350mA
- 1x Qunqi L298N Motor Drive Controller Board Module Dual H Bridge
- 3x ESP32
- 3x SN65HVD230 CAN Board
- 1x Red LED
- 2x Green LED
- 3x 220 ohm resistors
- Breadboard
- Assorted Wires
- 12V Power Supply

Software:

- HTML
- Python 3
- Unity Version 2019.3 10f1
 - Oculus Integration Package
- Espressif IoT Development Framework
- Mjpg-Streamer

Setup

The setup for our 2nd prototype test includes an Oculus Rift S connected to a computer running Unity, an ESP32 CANbus system, three stepper motors and their respective stepper motor controllers, and a Raspberry Pi 4 connected to a Raspberry Pi camera and running a web server. The Unity program allows the user to wear the headset and record their movements as they turn their head. The CAN Bus system will be demonstrated by toggling LEDs to indicate successful CAN Bus message transmissions. The accuracy of turning the motor will be demonstrated by Finally, the camera system is able to record footage in real time and stream it to an Apache 2 web server.

Pre-testing Setup Procedure

1. Start the recording stream on the camera.
2. Flash each receiving CAN node with `can_receiver.c`
3. Flash transmitting CAN node with `can_transmitter.c` then unplug it.

Testing Procedure

1. Run Unity program on computer and use headset to record movements.
 - a. After ending the program, the movements should be saved into a file named *movementdata.txt*.
2. Plug in transmitting CAN node
 - a. Red LED toggles for transmitting node and green for receiving nodes. Toggling at the same time (brightness can be inversed) indicates successful CAN transmission/reception.
3. Demonstrate the motors being able to move a set amount of degrees
 - a. Show rotations to and from a home position, of iterations of 90, 180, and 360 degrees
4. Test the camera stream linked to the IP address of the Raspberry Pi.

Measurable Criteria

The criteria for successful test and output is as follows:

- I. The Unity program should successfully record headset movements from the user and save them to *movementdata.txt*.
- II. The LEDs should toggle at the same time to.
- III. The live stream from the camera could be viewed from the web browser.

Hardware Pinout

Each ESP32	Usage/Description
GPIO 14	LED Power
3V	CAN Power
GND	CAN and LED GND
GPIO 25	CAN RX
GPIO 26	CAN 26