# 

# **Laser Car Weed Control System**

## **Table of Contents**

1. Overview
2. Features
3. Architecture
4. Hardware Requirements
5. Software Requirements
6. Directory Structure
7. Configuration
8. Installation & Setup
9. Usage
10. Command Protocol (ESP32)
11. Troubleshooting
12. Contributing & Support
13. License
14. Contact

## **1. Overview**

A collaborative robotics project combining four platforms into an autonomous weed-control vehicle:

* Raspberry Pi: streams IMU & GPS data via MQTT.
* Jetson Nano: runs YOLOv8 on camera frames, publishes detections & frames.
* ESP32: receives driving commands over TCP to control RoboClaw motors; falls back to FlySky RC.
* Host PC: Flask web dashboard for live map, video, PID tuning, and laser treatment.

Communication uses MQTT (port 1883) and TCP (port 10001) on a shared Wi-Fi LAN.

## **2. Features**

• Real-time IMU + GPS sensor fusion for robust navigation  
 • YOLOv8 weed detection with normalized bounding boxes  
 • Live video streaming with overlaid weed markers  
 • On-screen & RC transmitter driving (WASD/arrow keys, Space = stop)  
 • Real-time PID tuning (Kp, Ki, Kd, speed)  
 • Manual & automatic laser treatment with safety delay, aim power, before/after captures  
 • Dataset collection interface

## **3. Architecture**

RC Transmitter → ESP32 ↔ TCP(10001) ↔ Jetson Nano → MQTT(1883) ← Raspberry Pi  
 ESP32 → RoboClaw → DC Motors; Host PC subscribes to MQTT, serves Flask UI, controls laser via USB serial

## **4. Hardware Requirements**

• Raspberry Pi 3/4 + BerryIMU (v1/v2/v3) + GPS module  
 • Jetson Nano + USB webcam  
 • ESP32 Dev Kit + FlySky iBus receiver + RoboClaw motor controller + 2× DC motors + wheels  
 • Laser diode module (TTL via USB/serial)  
 • Wi-Fi network connecting all devices on the same subnet

## **5. Software Requirements**

Raspberry Pi  
 • OS: Raspbian Buster or later  
 • Python 3, libraries: paho-mqtt, gpsd, BerryIMU

Jetson Nano  
 • JetPack (CUDA, cuDNN, TensorRT)  
 • Docker, Python 3, ultralytics, opencv-python, paho-mqtt

ESP32  
 • Arduino IDE ≥1.8.13 or PlatformIO  
 • ESP32 board package, libraries: IBusBM, RoboClaw

Host PC  
 • Python 3, libraries: Flask, paho-mqtt, opencv-python, numpy  
 • Web browser (Chrome/Firefox)

## **6. Directory Structure**

esp32/  
  RobotController.ino  
 pi/  
  berryIMU\_mqtt.py  
 jetson/  
  weed4.py  
 host/  
  app.py  
  requirements.txt  
  templates/index.html  
  static/css/  
  static/js/  
  static/captures/  
 README.md

## **7. Configuration**

Network & IPs (example):  
 • Raspberry Pi: 192.168.1.103  
 • Jetson Nano: 192.168.1.102  
 • Host PC: 192.168.1.104  
 • ESP32: DHCP or reserved static

MQTT Broker: run Mosquitto on Pi or point clients to 192.168.1.103:1883

ESP32 Wi-Fi credentials & motor addresses inside RobotController.ino:  
 ssid, password, TCP\_PORT 10001, Mot1 0x80, Mot2 0x81

Laser port (Host PC): /dev/ttyUSB0

## **8. Installation & Setup**

### **a. Raspberry Pi (IMU & GPS)**

1. SSH into Pi
2. Install dependencies: sudo apt install python3-pip gpsd-clients
3. pip3 install paho-mqtt gpsd BerryIMU
4. cd pi/, edit broker IP if needed, run python3 berryIMU\_mqtt.py

### **b. Jetson Nano (YOLO Detector)**

1. SSH into Nano
2. sudo nvpmodel -m 0 && sudo jetson\_clocks
3. Build or pull Docker container preconfigured for YOLOv8
4. docker run --gpus all --network host …
5. Inside container: pip3 install ultralytics opencv-python paho-mqtt, run python3 weed4.py

### **c. ESP32 (RoboClaw Controller)**

1. Open RobotController.ino in Arduino IDE
2. Update Wi-Fi & motor addresses
3. Install IBusBM & RoboClaw libraries
4. Select ESP32 board, upload
5. Monitor serial (115200 baud) for IP and TCP connections

### **d. Host PC (Flask Web App)**

1. Clone host/ folder, create and activate Python venv
2. pip install -r requirements.txt
3. Edit app.py: ESP32 IP for RobotSocket, broker IP for MQTT
4. Run python app.py, open http://192.168.1.104:5049

## **9. Usage**

1. Click “Connect Robot” and “Connect Laser” in the UI
2. Drive via on-screen arrows or WASD/Space
3. Monitor map (GPS/IMU fusion) and video (weed overlays)
4. Adjust PID sliders for better motor response
5. Operate laser manually (On/Off) or run AUTO experiment
6. Use “Dataset” tab to upload training images

## **10. Command Protocol (ESP32)**

Send text commands over TCP port 10001 (each ending with CRLF):  
 SYS CAL → calibrate  
 MMW !MG → enable motor guard  
 MMW !M <L> <R> → set left/right speed (–250…250 mapped to 0–127)

# 

## **11.Modifying Auto-Start Configurations**

If you need to modify the auto-start services that have been configured, follow these instructions:

### **Raspberry Pi Service**

To edit the Raspberry Pi IMU/GPS service:

bash

*# Edit the service file*

sudo nano /etc/systemd/system/berryimu.service

*# After making changes, reload and restart*

sudo systemctl daemon-reload

sudo systemctl restart berryimu.service

*# Check status to verify changes*

sudo systemctl status berryimu.service

### **Jetson Nano Service**

To edit the Jetson Nano weed detection service:

bash

*# Edit the service file*

sudo nano /etc/systemd/system/weed-detection.service

*# Edit the startup script*

sudo nano /home/nano241/start\_weed\_detection.sh

*# After making changes, reload and restart*

sudo systemctl daemon-reload

sudo systemctl restart weed-detection.service

*# Check status and logs*

sudo systemctl status weed-detection.service

cat /home/nano241/weed\_service.log

### **Common Modifications**

#### **Change MQTT Broker Address**

If the MQTT broker IP changes, update in:

* Raspberry Pi: Edit berryIMU\_mqtt.py
* Jetson Nano: Edit Docker container's weed4.py
* Control Computer: Edit app.py

#### **Change Camera Settings**

To modify camera resolution or frame rate:

bash

*# Connect to Docker container*

sudo docker exec -it yolov11n /bin/bash

*# Edit the detection script*

nano /ultralytics/weed4.py

#### **Adjust Startup Dependencies**

If you need to change service dependencies or startup order:

* Modify the After= and Wants= parameters in the service files
* Change network waiting time in start\_weed\_detection.sh

#### **Disable Auto-Start**

To temporarily disable auto-start:

bash

sudo systemctl disable weed-detection.service *# Jetson Nano*

sudo systemctl disable berryimu.service *# Raspberry Pi*

Always check log files after making changes to verify everything works correctly.