# Machine Learning for Beacon Tracking and Autonomous Navigation Using UGV and ESP32

Gautam Singh

April 12, 2023

# Outline

# Aim

Implement a machine learning based algorithm on a WiFi-enabled microcontroller such as the ESP32 to navigate the unmanned ground vehicle (UGV) towards a beacon (here, a WiFi access point).

# Aim

Implement a machine learning based algorithm on a WiFi-enabled microcontroller such as the ESP32 to navigate the unmanned ground vehicle (UGV) towards a beacon (here, a WiFi access point).

## Assumptions

1. No obstacles surrounding UGV and beacon.
2. No undulating terrain for beacon to navigate.

# Hardware

1. UGV chassis with DC motors
2. ESP32 microcontroller with Type-B USB cable
3. L293D Motor Driver IC
4. Breadboard and Jumper Wires
5. Android phone
6. (Optional) USB 2.0/3.0 Hub

# Software

Relevant platformio codes can be found here.

1. In this directory, type `pio run` to generate the firmware to flash to the ESP32.
2. Using ArduinoDroid, flash it to the ESP32 from your Android phone.

A more detailed manual is present here.

# Circuit Diagram

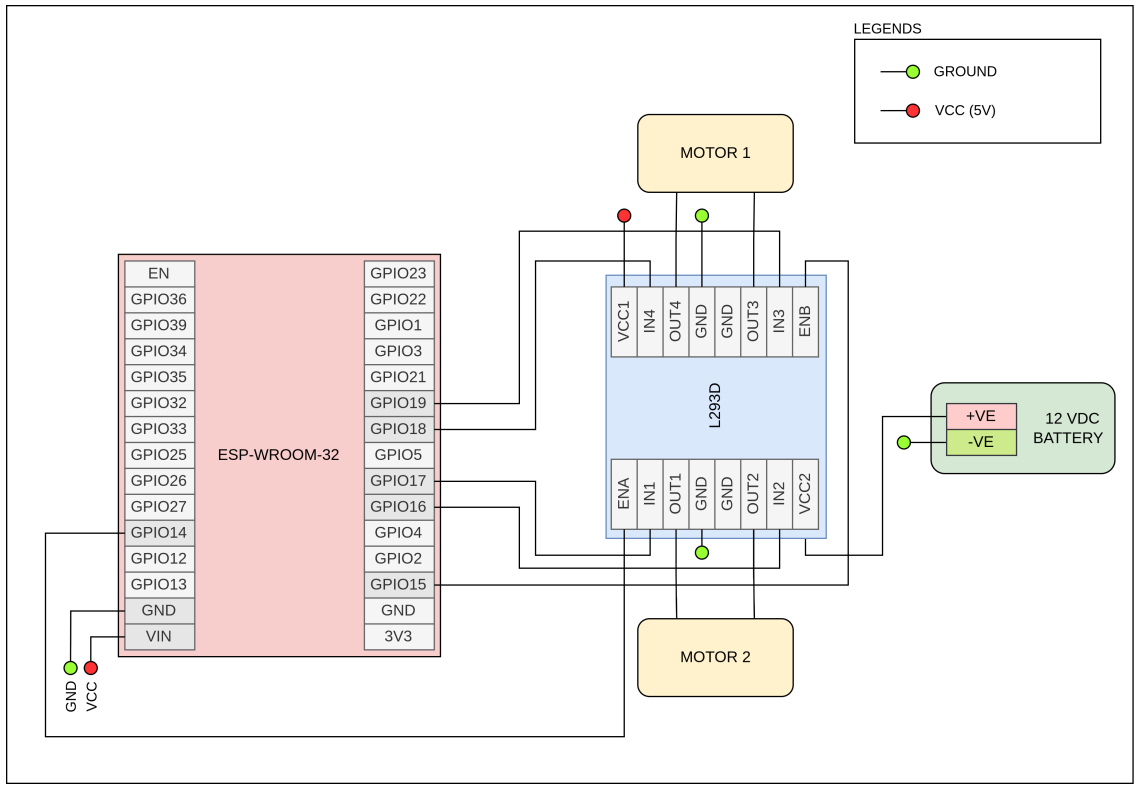


Figure: Circuit Diagram for Beacon Tracking.

# Underlying Principles

1. To estimate (radial) distance to beacon, we use its signal strength. For WiFi, this is the **Received Signal Strength Indicator** (RSSI).

# Underlying Principles

1. To estimate (radial) distance to beacon, we use its signal strength. For WiFi, this is the **Received Signal Strength Indicator** (RSSI).
2. The RSSI ($R$ dBm) at distance $d$ metres is given by

$$R(d) = R(1) - 10 \log_{10}(d) \tag{1}$$

# Underlying Principles

1. To estimate (radial) distance to beacon, we use its signal strength. For WiFi, this is the **Received Signal Strength Indicator** (RSSI).
2. The RSSI ($R$ dBm) at distance $d$ metres is given by

$$R(d) = R(1) - 10 \log_{10}(d) \tag{1}$$

3. Clearly, $R(d)$ is a convex function. Hence, we can use gradient descent.

# Underlying Principles

1. To estimate (radial) distance to beacon, we use its signal strength. For WiFi, this is the **Received Signal Strength Indicator** (RSSI).
2. The RSSI ($R$ dBm) at distance $d$ metres is given by

$$R(d) = R(1) - 10 \log_{10}(d) \tag{1}$$

3. Clearly, $R(d)$ is a convex function. Hence, we can use gradient descent.
4. *But how do we implement it?*

# Algorithm Description

Note that the following is a generic recursive description. For concrete examples, please see slide number 5.

# Algorithm Description

Note that the following is a generic recursive description. For concrete examples, please see slide number 5.

1. If the UGV is close enough to the beacon, terminate.

# Algorithm Description

Note that the following is a generic recursive description. For concrete examples, please see slide number 5.

1. If the UGV is close enough to the beacon, terminate.
2. Take measurements at various points on a straight line.

# Algorithm Description

Note that the following is a generic recursive description. For concrete examples, please see slide number 5.

1. If the UGV is close enough to the beacon, terminate.
2. Take measurements at various points on a straight line.
3. Based on these measurements, decide the next move of the UGV, and recurse till the UGV is close enough to the beacon.

# In-Class Demonstration

# Conclusions

1. The UGV eventually converges close to the beacon.

# Conclusions

1. The UGV eventually converges close to the beacon.
2. However, if there are a lot of nearby obstacles, the UGV may not converge close to the location of the beacon. It may either get physically blocked by the beacon or the signal interference may be too high.

# Thank You!