Rover Tracker
Dov Cattan
CDA 4630-002
08/08/2021

The goal of the Rover Tracker project is to track a remote-control rover's rotation and acceleration using an inertial measurement unit (IMU) sensor and connected to a web server to display the accelerometer, gyroscope and even temperature readings. The way it works is that the Microcontroller Unit (MCU), collects the data from the IMU sensor, and then it sends them to the web server with the Wi-Fi Connection. This means all the current readings from the IMU that was collected by the MCU can be accessed by the IP address.

Table of Contents:

Contents

1 Introduction

1.1 Problem Description

The project tracks the Rover’s acceleration and rotation using a web server.

1.2 Significance of the Problem

It is important to track a device’s rotation and acceleration remotely in case someone not in a vehicle needs to see if car is flipped remotely, over if a coach wants to see if his player is running fast enough remotely

1.3 Goals and objectives

1. Build and wire the microcontroller and Inertial Measurement Unit
2. Code the microcontroller to initiate the Wi-Fi collect data from IMU
3. Code to build the Web Server for it to display the data collected by the microcontroller online
4. Adjust the code as necessary to display the readings in different Units such as from radians to degrees or from Celsius to Fahrenheit
5. Put in on the RC Rover and then record what you have tracked.

2 System Design

2.1 Project Requirements

2.1.1 The Rover will have a MSP 430 microcontroller on top of it with an Inertial Measurement Unit to collect the gyroscope and accelerometer readings
2.1.2 All the readings from the MSP 430 microcontroller will be displayed on a smartphone app connected to Bluetooth using an external Bluetooth sensor connected to the MSP 430.

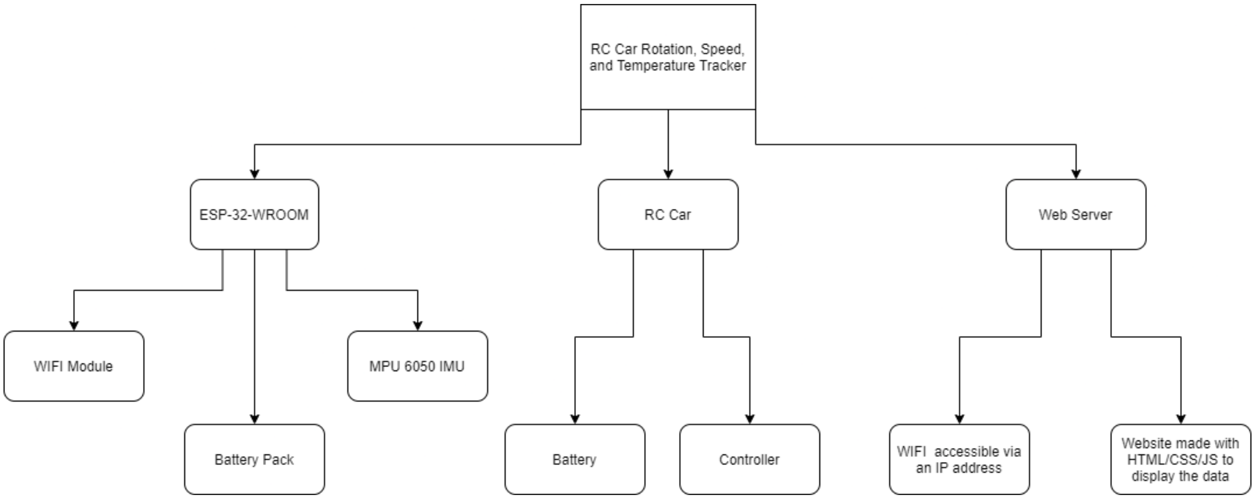
2.2 Project Requirements Not Met

Due to not having an external Bluetooth sensor, but rather got an ESP 32 built in Wi-Fi microcontroller instead, I abandoned meeting the requirements for having a Bluetooth smartphone app connected to the rover, and not having an MSP 430 to do so.

2.3 Project Requirements Added

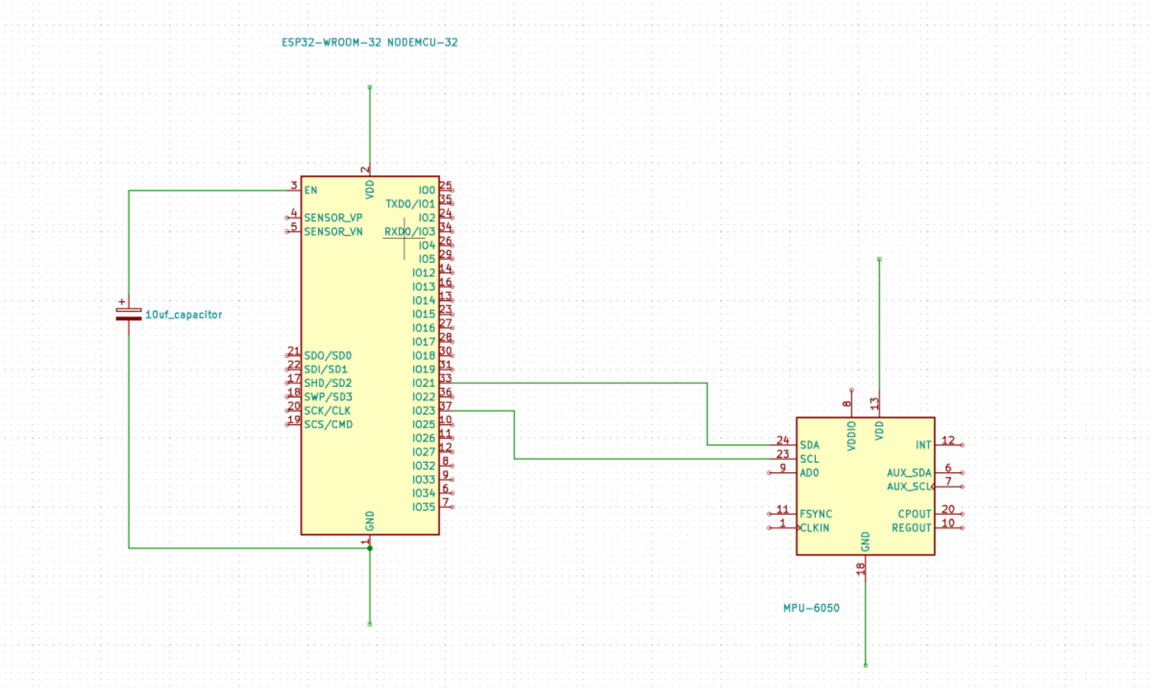
Since I couldn’t do a smartphone app for what I wanted to do, I decided to build a web server instead of having a smartphone app, therefore making it accessible both by a desktop and mobile browser via an IP address. This uses the built in Wi-Fi sensor of the ESP 32 microcontroller

2.4 System Diagram

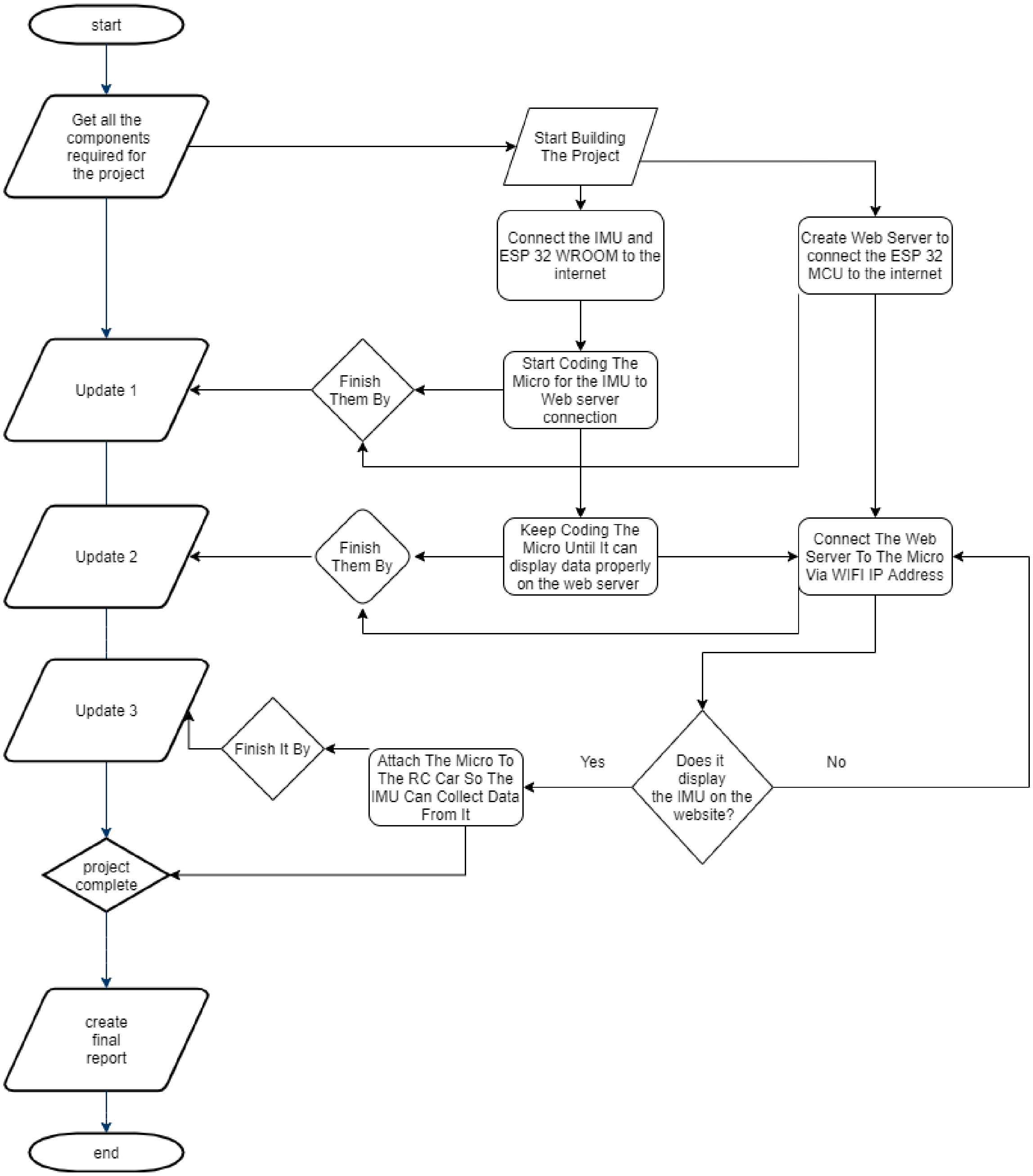

RC Car Rotation, Speed, and Temperature Tracker
ESP-32-WROOM
RC Car
Web Server
WIFI Module
MPU 6050 IMU
Battery Pack
Battery
Controller
WIFI accessible via an IP address
Website made with HTML/CSS/JS to display the data

It shows how the sub-systems work together to properly create the Rover Tracker.
The breadboard of the rover tracker with of all the components together (MCU, IMU) was stuck onto the rover with a rubber band, along with its battery pack

Breadboard Schematic On KiCad



2.5 Flowchart Diagram


start
Get all the components required for the project
Start Building The Project
Connect the IMU and ESP 32 WROOM to the internet
Create Web Server to connect the ESP 32 MCU to the internet
Update 1
Finish Them By
Start Coding The Micro for the IMU to Web server connection
Update 2
Finish Them By
Keep Coding The Micro Until It can display data properly on the web server
Connect The Web Server To The Micro Via WIFI IP Address
Update 3
Finish It By
Attach The Micro To The RC Car So The IMU Can Collect Data From It
Yes
Does it display the IMU on the website?
No
project complete
create final report
end

3 System Implementation

3.1 Hardware

- Remote Control Rover with Rechargeable batteries
- Controller with Alkaline Batteries
- ESP 32 NodeMCU 32S Microcontroller

- MPU 6050 Inertial Measurement Unit
- 10uf Capacitor
- 1000ma Battery Pack

3.2 Software

- Arduino IDE for the Microcontroller Code
- VS Code for the Web Server Code
- Web Server to test the Web Server

3.3 User Interface

The web server displays the gyroscope readings in degrees, the accelerometer readings in meters per second squared, and the temperature in Fahrenheit. It also enables you to reset the readings when needed.

3.4 Data Communications

The data is received by the Wi-Fi the ESP 32 is connected to by the IP address of the router. It can be accessed by any device that has a browser by typing the correct IP address to the browser search bar.

4 Testing and Performance Evaluation

This project needed to constantly be tested in order to work, but the first thing was to test it by the computer and make sure it was even able to be connected to the Wi-Fi.

Then I had to then make sure the web server was properly reading the data collected by the microcontroller.

Next, I had to then test it on the battery pack to make sure it's not only working just because it's connected to the computer.

Lastly, I had to make sure it was able to work in another Wi-Fi location to confirm it can work anywhere that has a Wi-Fi connection

4.1 Performance Assessment

It had times where it was flawed, but the more improvements I put into it, the better it was able to work.

5 Budget

Most of the components were given by the college, but the Rover RC Controller was 12.69$ and the IMU was 7.99$ all bought on Amazon