#MComp Mower Project

ESP32 Programming

Open `Lawnmower-system/Microcontroller/Robot Sensor Motor Controller - DMP/` in VSCode with the PlatformIO extension and upload the PIO project to the ESP32

Raspberry PI

- Install Ubuntu 22.04 and ROS2 (ROS2 install instruction: https://docs.ros.org/en/humble/Installation.html)
- Clone the `Lawnmower-system` repo (https://github.com/UEA-MComp/Lawnmower-system)
- Clone the `BNO085-ROS2-Node` repo (https://github.com/UEA-MComp/BNO085-ROS2-Node)
- Clone the `telemetry` repo (https://github.com/UEA-MComp/telemetry)
- Using a Terminal, navigate into `Lawnmower-system/ros_mower_robot_ws/`
    - Source ROS and run `colcon build` in this directory
    - Source (in the same repo) `install/setup.bash`
    - Need to start the bridge node for communiction with the ESP32 over USB
        - run `ros2 run robot_motor_control_py robot_bridge`
- In a new Terminal session , navigate into `telemetry`
    - check all sub module are present in `src`
    - `11feb-linux.conf` is the RTKLIB config file. You will need to update this with you RTK Setup
    - Follow the rest of the instruction detailed in `README.md`
    - There is a launch file in there that makes it nice and easy to run
    - By default, the robot doesn't send telemetry information to the server (it's commented out in the launch file), if you want to send information to the server you need to define a ROS discovery IP and a ROS session ID in a new ROS session. Again more notes on how to do this is in the repository
- Using a new Terminal session, navigate into `Lawnmower-system/ros_mower_robot_ws/`
    - source ROS and `install/setup.bash`
    - In `lidar_sim.py` there are vars called `test_shape`, `nogos`, and `path` for the perimeter, nogo areas, and generated path respectively.
    - After updating perimeter, nogos, and path, run `colcon build`
    - run `ros2 run path_nav robot_nav`

Server

- The server acts as a bridge between the robot and phone app
- In practice, we didn't do too much with it, its more a proof of concept
- You need to have access to a server. You need to set up a wireguard server on that server
- All of the server stuff is in `serverfiles-dockerfiles` repo
- There is detailed notes in there.
- It is all docker stuff (learn docker 😛) there is a nice `docker-compose.yml` that deploys everything. First you will need to update the `db.env` file and the wireguard config file first
- The docker build clones other repos so you might need to update the repository URL in the Dockerfile.

Misc Note about running robot

- You will want to calibrate the IMU of the Robot.You can do that with the instruction in `Lawnmower-system/Calibration.md`

Our email addresses

- Eden Attenborough `eden.attenborough@outlook.com` will hopefully be back at UEA in 2025 😛
- Tye Buckingham `buckingham.t.g@gmail.com` for pathing and obstacle avoidance - further details can be found at repos under https://github.com/Tye-Buckingham
- Robert Stevenson `RobertLStevenson01@gmail.com` for robot hardware