Instructions

This is a small project to assess your systems engineering, coding abilities, and problem solving skills. The project is open-ended and can be taken to whatever level you see fit and include any libraries, methods, or software you are proficient in or know of.

The test is due back in 1-week and must meet the following requirements:

Please use the google drive link provided to store all your files and code and documentation for your test.
Please use git to source control your code. You may not be able to store all large files on github, such as databases or models, so please ensure the google drive location has the git project and any files needed to run the project in full.
Please provide a README about your project. We are not looking for overkill, but we need to know how to test it and have an overview of the code itself.
For this test, please restrict yourself to Python or C++ for the programming languages.
After this take-home, we will have one more interview to discuss your approach and code.
The major steps are outlined below and some are harder than others, so get as far as you can, and don't worry if you don't finish them all!

The Problem

We would like you to design a simple differential drive robot platform and (optionally) implement as much as you can using hardware that you have on-hand. The purpose of this project is to evaluate your systems design, mechanical design, part sourcing, electronics design, and programming experience. We do not expect you to complete all aspects of the following and are more interested in the way you approach the problem and demonstrate engineering proficiencies in each of the relevant disciplines. Focus on the parts you are most comfortable with and have fun!

There are two separate projects we'd like you to attempt. One is focused on your mechanical design proficiency and the other is focused on firmware/software development.

Mechanical Design:

Design a simple adjustable camera mount in a CAD software of your choice.

- Design with the intent to 3D print using a low-cost FFF printer with PLA.
- The mount should fit the OAK-D stereo depth camera from Luxonis (LINK)
- The mount should allow for rotation of the camera about the 'Y' axis (pitch) and the 'Z' axis (yaw). The 'X' axis (roll) can be ignored. In other words we should be able to point the camera up and down or side to side.
- Include a locking/unlocking mechanism for position adjustments. This could be as simple as a screw to tighten the joint.
- The mount should magnetically attach to a vertical surface without slipping. Please pick the appropriate magnets to meet this requirement.

- Source required components from suppliers online and build a BOM with pricing and purchase links
    - Please think of all hardware involved including magnets, screws, 3D printed components/materials, etc. You do not need to select the printer or any tools.
- Use off-the-shelf components where possible. It's even allowable to select a complete camera mount and design custom brackets to mount the OAK-D or to design the mount from the ground up. I leave the method implementation of this mount entirely up to you.

Electronics/Software/Firmware Development:
Design a simple IMU data logging system using a microcontroller and host PC.

- Select a microcontroller and IMU.
    - Arduino or other MCU
    - IMU breakout board: OK
- Create a simple wiring diagram for connecting the IMU to the microcontroller
- Create simple firmware for a microcontroller
    - Collects data from the IMU
    - Passes accelerometer and gyroscope data to an external device over a USB serial connection or (optionally) a wifi connection.
    - Use whatever IDE and language you are comfortable with.
- Create a simple client program that runs on a PC
    - Connected via USB (or WiFi) to the MCU
    - Receives IMU data over USB/WiFi from the MCU and logs timestamped data to a CSV file.
    - Can be in an IDE and language of your choosing.
- Please note that we do not expect you to actually design and test this system. I'd like to see you lay everything out and push the code as far as you can.
    - Ideally I'd like to see you send and log timestamped "Hello World" or dummy IMU data to a csv file over serial/WiFi.
    - I'd like to see you import the required libraries to drive the IMU and compilable code to interface with it. I don't expect you to be able to test this without hardware.