ECEN 361 Project Definition
Self Balencing Robot

GROUP: THE WEEBLE WOBBLES

JAMIN POTTLE, HIRAM BENROTH, KYLE MURDOCK,
NICHOLAS PETERSEN, WYATT CORRIGAN.

Project Scope

The project's main function is to make a Remote-Controlled Self Balancing Bi-Wheeled Robot. The robot will use two motors and an inertial measurement unit to balance itself out. The vehicle will also connect to an Xbox controller wirelessly via Bluetooth, this will control the movement of the robot, so that we can drive it around for demonstration purposes. As a stretch we want overrides from sonar sensors that prevent the robot from crashing. This robot serves recreational purposes, and is useful because it could be kid proof, as you can drive it, however you cannot destroy anything.

Lessons to Learn

Main team learning objectives include but are not limited to the following. First, we need to learn how to interface with motors and the inertial measurement unit. We also are going to research and develop a Proportional-Integral-Derivative, so that we can smooth out movement and more dynamically control the balancing. As well we need to learn physics of Pendulums balance, and how to apply that in hardware criteria and specifications, as well as in code. Finally, we need to learn about overrides for preventing crashes, as well as Bluetooth protocols to connect an Xbox controller.

Roles and Responsibilities

Role	Name	Responsibilities
Team Leader	Hiram Benroth	● Set up project in Monday.com and invite team ● Manage project tasks in Monday.com ● Submits group assignments ● Establishes meeting schedule (1-2 times per week)
Hardware Lead	Jamin Pottle	● Final decision maker on hardware selection ● Hardware block diagram owner ● Gathers necessary hardware (purchase or loan)

Software Lead	Kyle Murdock	• Final decision maker on software architecture • Software block diagram owner • Set up Git repository and share with team
Hardware Liaison	Nicholas Petersen	• Responsible for configuring IMU for gyroscope config (pitch, yaw, roll) • Working with the PID Controller to correct errors • Responsible for possible dead reckoning configuration using wheel encoders
CAD Lead	Wyatt Corrigan	• Responsible For Being owner of Overall 3d Files • Works With Hardware Lead • Brings different iterations of design to group meetings.

Schedule

Create initial set of tasks in Monday.com and assign a person and dates and durations. Export main table to Excel and Gantt chart to PDF and turn in with this sheet.

Derived Requirements

List the requirements for your project. Derived requirements are typically broken down into three categories: general, interface, and functional.

General Requirements

General requirements detail the general aspects of the system and cover items such as size, weight, coding standards followed, licensing, etc. Usually this list is short, but not empty. Use these to help guide hardware selection.

-
- Platform.io
- Coding Standards
    - Naming Convention
        - Variables: camelCase, if there is a number put it at the end, ex. var2, endPoint
        - Functions: CapitalCase, action first, ActionType() i.g. GetVar();
            - FILENAME_GetVar();
        - Enums: ENUMTITLE_NAME
        - DefType(Structs): One word, all lowercase; i.g. string
        - Defines: ALL_CAPS
        - Includes: lowercase, <stdlib.h> for outside libraries, “mylib.h” homemade ones
    - Documentation
        - Description after first curly bracket.
        - Normal number of comments for easy readability.
- Hardware Requirements

- 1-2kg PLA Chassis
- (2) DC motors w/wheels
- ESP32 Microcontroller
- IMU
- 12V Power supply
- DC Power converter unit
- Ultrasonic sensor (stretch)

Interface Requirements

This project will include different protocols to influence communication with the different pieces. First, we are going to use I2C to communicate with the IMU. WE are also going to have a Bluetooth frequency at a baud rate of 9600 or 115200, depending on what works. These are going to be the main communication standards that we use.

Functional Requirements

The Balancing Robot will be able to functionally maneuver around any space with the capability of autonomously detecting obstructions in its path. It will also contain balancing and self-correctional functions so that it can carry any object while moving back to an erect position, keeping the object vertically balanced.