**4180 Final Project**

**Balancing Robot**

**By**

**Daniel Niemann, Fan Han Hoon, Sid Gulati, Muath F Alsubhi**

**For**

**Dr. James Hamblen**

**ECE-4180-A: Embedded Systems**

**December 8th, 2023**

**Introduction**

The field of robotics and embedded systems is rapidly advancing, bringing forward innovations that blend mechanical design with sophisticated software control. This report presents a comprehensive overview of a project undertaken in an Embedded Systems class – the design and implementation of a balancing robot. This project focuses on utilizing a combination of hardware components and embedded software to achieve the challenging task of maintaining dynamic balance in a robotic system.

**Project Goal**

The primary objective of this project is to construct a robot that can autonomously maintain its balance in real-time. This involves a careful integration of motion sensors, motors, and control algorithms to create a system that can sense its orientation and make rapid adjustments to avoid falling over. The project aims not only to demonstrate technical skills in embedded systems design but also to explore practical applications of control theory in robotics.

**Components Overview**

The balancing robot is built using key components that work in unison to achieve the desired functionality:

1. **Motors:** Two motors are employed to drive the wheels of the robot. These motors are critical for initiating movement and making adjustments to maintain balance. The selection of the motors was based on factors like torque, speed, and response time to ensure they are capable of reacting swiftly to balance corrections.
2. **Chassis:** The chassis forms the structural foundation of the robot. It is designed to be robust yet lightweight, offering the necessary support for all components while minimizing inertia. The design of the chassis significantly impacts the robot's ability to balance by affecting its center of gravity and stability.
3. **LMS9DS0 IMU (Inertial Measurement Unit):** At the heart of the robot's sensing capabilities is the LMS9DS0 IMU. This compact module includes a 3D accelerometer, 3D gyroscope, and 3D magnetometer. For the purpose of this project, the accelerometer and gyroscope are of primary importance. They provide real-time data on the robot's tilt and rotational motion, which are crucial for the balancing algorithm.
4. **HC-SR04 Sonar Sensor:** The sonar sensor offers a safety protocol where it stop all motor movements when it detects any obstruction to prevent any additional fall damage exerted on the circuit. The selection of sonar sensor is based on the accuracy of the reading and the data reading speed.
5. **Adafruit Bluefruit LE UART Friend (BLE):** The Bluetooth module enables the user to have additional user inputs that controls the robot. It could also be used as a serial port for debugging purposes during the project development process.
6. **H-bridge:** H-bridge is a crucial component that was used to send outputs for motor movements such as forward and backward kinematics. It allows the user to output multiple motors under the same PWM settings.

**Significance in Embedded Systems**

This balancing robot project exemplifies the application of embedded systems in creating intelligent, autonomous machines. By integrating sensors, actuators, and control systems within a single design, the project highlights how embedded systems can be used to solve complex, real-world challenges. It serves as a practical demonstration of concepts like sensor integration, real-time data processing, and closed-loop control, which are fundamental to the field of robotics and automation.

#### System Architecture

The balancing robot is designed with a focus on achieving and maintaining dynamic equilibrium through a harmonious interplay of its hardware and software components. The system architecture is bifurcated into two main sections: hardware design and software control.

##### Hardware Design

The hardware components of the robot are strategically chosen and assembled to facilitate balance and movement.

1. **Chassis:** The chassis is engineered to be as symmetrical as possible to ensure uniform weight distribution, which is vital for maintaining balance. Its low center of gravity aids in stabilizing the robot during dynamic movements.
2. **Motors:** Two high-precision DC motors are used. Their responsiveness and torque are key for making quick and accurate adjustments to keep the robot upright. The motors are connected to the wheels, and their speed and direction are controlled to adjust the robot's tilt and balance.
3. **LMS9DS0 IMU Integration:** Positioned centrally on the chassis, the LMS9DS0 IMU is crucial for sensing the orientation of the robot. Its accelerometer provides data on linear acceleration, while the gyroscope offers information on angular velocity – both essential for determining the robot's instantaneous position and motion.
4. **HC-SR04 Sonar Sensor:** The sonar sensor is positioned at the tip of the robot as it is more prone to any sensitive distance reading as it will be the first point of contact upon any spinning or falling momentum. The location of the sensor also provides an easier debugging process on the calibrations.
5. **Adafruit Bluefruit LE UART Friend (BLE):** The Bluetooth module was placed above the battery pack to ensure that it provides stability on maintaining the low center of mass of the robot. It will be utilized for scaling the motor speed on spinning movements, forward and backward movements and PID balancing mode.
6. **H-bridge:** Due to the limited space of the circuit, the H-bridge was placed at the nearest spot above the mbed. It was one of the most important part of the robot as the balancing movement is highly sensitive to any changes on the motor output.

##### Software Control

The software aspect of the robot is developed using the mbed platform, known for its real-time operating capabilities which are essential for this project.

1. **Sensor Data Acquisition:** The LMS9DS0 IMU's data is continuously polled to monitor the robot's orientation. This data is used to determine if the robot is tilting in any direction.
2. **Data Processing:** The raw data from the IMU is processed using filtering techniques to remove noise and improve accuracy. This processed data provides a clear and reliable understanding of the robot's current state.
3. **Control Algorithm:** A Proportional-Integral-Derivative (PID) control algorithm was implemented. This algorithm continually calculates an error value as the difference between a desired setpoint (upright position) and the current tilt angle and applies a correction based on proportional, integral, and derivative terms. The PID controller's output modulates the speed and direction of the motors to counteract any detected tilt, thus maintaining balance.
4. **Real Time OS: A real time operating system (RTOS) was implemented for multithreading. The RTOS was designed to speed up the process of reading inputs from Bluetooth, getting real-time data from sonar sensor and running IMU and motor controls at the same time. If RTOS was not implemented, the delays between sensor readings will cause a huge output delays for the motor output on balancing.**

#### Implementation Challenges

Throughout the design and implementation phase, several challenges were encountered:

* **Sensor Calibration:** Ensuring that the LMS9DS0 IMU provided accurate and consistent data was critical. Proper calibration was necessary to account for any sensor biases or drift.
* **Algorithm Tuning:** Fine-tuning the PID controller's parameters (Kp, Ki, Kd) was essential for achieving a balance between responsiveness and stability. This required extensive testing and iterative adjustments.
* **Mechanical Design:** The mechanical construction of the robot, particularly the chassis and wheel alignment, had to be precise to ensure that the robot could balance effectively.
* **Circuit Design:** Due to the limited space on the robot, the team had to pack all the critical sensors as close to the mbed as the team noticed that a longer cable connection will often result in a significant delay on the data transmission.

#### Testing and Observations

To validate the functionality and performance of the balancing robot, a series of tests were conducted under various conditions. These tests were designed to assess the robot's ability to maintain balance, respond to disturbances, and recover from different tilt angles.

**Testing Procedures:**

1. **Static Balance Test:** The robot was placed in a stationary upright position to observe its ability to maintain balance without external disturbances.
2. **Dynamic Balance Test:** The robot was subjected to gentle pushes from different directions to test its ability to recover and return to a balanced state.
3. **Inclined Surface Test:** Trials were conducted on surfaces with slight inclines to evaluate the robot's performance on uneven terrains.
4. **Endurance Test:** The robot's long-term stability and battery life were tested over extended periods to assess its practical usability.

#### Results

The results of the testing phase were promising, highlighting the robot's capability in maintaining dynamic balance:

* In the **Static Balance Test**, the robot successfully remained upright for extended periods without toppling over.
* During the **Dynamic Balance Test**, the robot demonstrated effective real-time adjustments to external pushes, showcasing the responsiveness of the control system.
* The **Inclined Surface Test** revealed some limitations in extreme conditions but overall satisfactory performance on mild slopes.
* The **Endurance Test** indicated adequate battery life and consistent performance over time, confirming the robot's practicality for continuous operation.

#### Performance Analysis

The effective balance maintained by the robot can be attributed to the precise data provided by the LMS9DS0 IMU and the efficient processing and response of the PID control algorithm. The tests confirmed that the integration of the IMU data with the motor control system via the PID algorithm was successful in achieving the project's primary goal of maintaining balance. The Bluetooth and sonar threading also proved that the balancing robot would also minimize the potential accidental fall damage and allowed users to interrupt the loop for different motor speed controls.

#### Improvements and Future Work

While the project met its fundamental objectives, there are areas for improvement:

* **Enhanced Sensor Fusion:** Incorporating additional sensors, like ultrasonic distance sensors, could enhance the robot's environmental awareness and stability.
* **Algorithm Optimization:** Further refinement of the PID controller's parameters could improve the robot's ability to handle more challenging scenarios.
* **Mechanical Design:** Adjustments to the chassis for better weight distribution and stability could enhance performance, especially on uneven surfaces.
* **Circuit Design:** A custom PCB design would allow the design to be more compact and potentially lower latency time on the IMU readings and motor outputs.

#### Conclusion

This project successfully demonstrates the application of embedded systems in robotics, particularly in creating a self-balancing robot. The integration of the LMS9DS0 IMU with a well-tuned PID control algorithm proved effective in maintaining dynamic equilibrium. This endeavor not only serves as a valuable learning experience in embedded system design but also contributes to the broader field of robotics by showcasing practical applications of control systems and sensor integration.