Inverted Pendulum BalanceBot

Electrical Engineering Senior Projects I – ECE395

Fall 2023

Prof. Carl Sable

Ahmad Malik & Nicholas Singh

Abstract

This project centers on the creation of a sophisticated self-balancing robot employing control systems driven by a PIC microcontroller. The primary objective is to design a dynamic and stable robot capable of maintaining equilibrium in real time, exemplifying the combination of robotics, control theory, and engineering. The robot's core functionality involves the system responsible for balancing the robot in the upright position, along with an independent balancing system for a tray designed to carry a load to maintain its own balance. The control is accomplished through precise sensor feedback, precise and highly reactive stepper motors, control algorithms, and the integration of a 48 MHz Microcontroller for enhanced processing capabilities. By leveraging a comprehensive set of sensors, including accelerometers and gyroscopes, the robot continuously monitors its tilt angle and rotational speed, similar to many hoverboards and Segways. Beyond its captivating demonstration of robotics and control theory principles, this project serves as a dynamic educational tool, offering hands-on insights into STEM concepts. Moreover, the robot's adaptability and potential for real-world applications, such as in autonomous vehicles or assistive technologies, underscore its broader relevance and innovative potential

Table of Contents

Introduction

The historical trajectory of self-balancing robots can be traced back to the early 20th century, rooted in the foundational principles of control theory within systems engineering and mathematics. Control theory, a pivotal discipline dealing with the behavior of dynamical systems influenced by inputs, has significantly shaped the landscape of self-balancing robotics. The PID controller, a ubiquitous control loop feedback mechanism widely employed in industrial control systems and diverse applications requiring continuous, precise control, stands as a testament to this evolution.

Advancements in sensor technology and high-speed microcontrollers have propelled self-balancing robots beyond rudimentary mechanical models to sophisticated systems capable of maintaining equilibrium under diverse conditions. Existing technologies in the realm of self-balancing systems have witnessed remarkable progress, laying the foundation for the innovative developments outlined in our project. One notable example is the Segway Personal Transporter, introduced in the early 2000s. The Segway employs a sophisticated system of gyroscopic sensors and accelerometers to maintain balance, enabling users to navigate effortlessly by leaning forward or backward [1]. This technology has found applications in various industries, from personal transportation to security and logistics.

Similarly, in the field of robotics, Boston Dynamics has pioneered self-balancing robots such as the "Handle." This two-wheeled robot combines advanced control algorithms with efficient wheel-based locomotion, showcasing dynamic stability and agility in comp environments [2]. The incorporation of balancing technologies in such robots has expanded their potential applications in logistics, warehouse automation, and even entertainment.

These existing technologies underscore the practicality and versatility of self-balancing systems. Our project builds upon this foundation, leveraging advancements in sensor technology and microcontrollers to contribute to the evolving landscape of dynamic robotic control.Our project builds upon these technological advancements, leveraging sensors such as accelerometers and gyroscopes, providing real-time precise data on the robot's tilt angle and rotational speed.

This project transcends the realm of academic exploration and harbors the potential for real-world applications. The principles underlying this self-balancing robot can be adapted for autonomous vehicles, where maintaining balance and stability is crucial. They could also be used in assistive technologies for individuals with mobility impairments, thus enhancing their independence and quality of life. This project also holds substantial educational value, serving as a tangible, hands-on tool to impart insights across the STEM spectrum (Science, Technology, Engineering, and Mathematics). Positioned as an engaging teaching aid, it provides students and educators with a practical understanding of complex concepts.

In essence, our project embodies a harmonious synthesis of technology, education, and innovation. It unveils exciting possibilities at the intersection of robotics and control systems, presenting a compelling journey for STEM enthusiasts, researchers, and technology aficionados. Subsequent sections will delve into the intricacies of the project, exploring its design, development, operation, and potential applications

Background

Hardware Communication Protocols:

This project involves several communication protocols implemented between the hardware components of the robot. These components include the inertial measurement unit (IMU), Load Cell, PIC18 microcontroller, 2.4 GHz receiver, and chip programmer. The communication structure revolves around the microcontroller, serving as the central hub, receiving sensory data from the load cell, receiving, and accelerometer and gyroscope readings from the IMU. The microcontroller undertakes the crucial task of processing the received data and subsequently generating pulses sent to the stepper motor drivers. These drivers, in turn, power the coils or phases of the corresponding stepper motor, resulting in motor movement. The communication protocols governing the interaction between the microcontroller and peripheral devices are I2C, PWM, SPI, and STEP/DIR

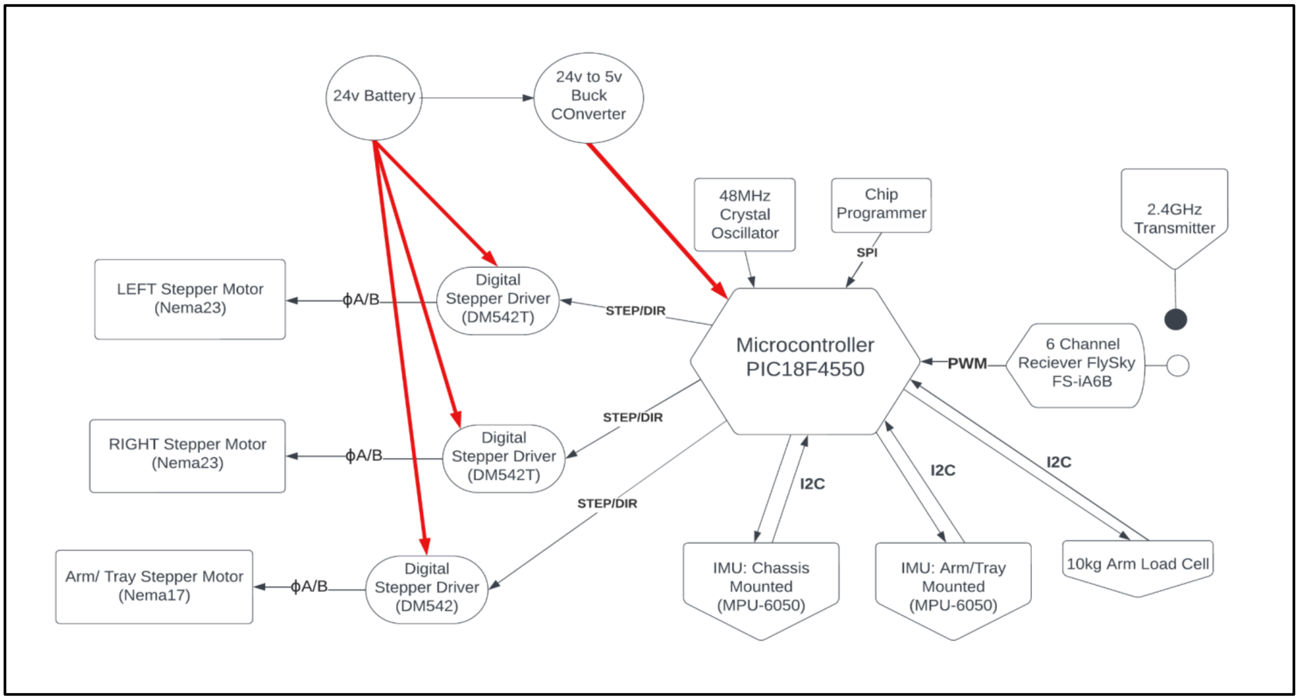


Figure 1: hardware components and their respective communication protocols

The diagram shows the various components and their respective communication protocols. I2C (Inter-Integrated Circuit) facilitates the exchange of data between the microcontroller and devices like the IMU. PWM (Pulse Width Modulation) is employed to regulate the power supplied to the stepper motor, enabling precise control over its movements. SPI (Serial Peripheral Interface) is utilized for high-speed, full-duplex communication between the microcontroller and other components, such as the 2.4 GHz receiver. STEP/DIR (Step/Direction) protocols dictate the movement of the stepper motor by conveying step pulses and direction signals. This communication framework forms the backbone of the self-balancing robot's functionality, ensuring efficient, fast data exchange and coordination among its hardware elements.

Pulse Width Modulation (PWM) is a digital technique used for precise control of analog circuits using microprocessor outputs. In this project, it will facilitate communication between a channel on the receiver to one of the general purpose input pins of the PIC18. This protocol involves modulating the duty cycle of a square wave, where the duty cycle represents the proportion of time the signal is "on" compared to the total cycle time. By adjusting the duty cycle of the square wave, data can be transmitted without delay. Typically, PWM finds diverse applications, from controlling LED brightness to motor speeds and power levels in various devices. Servos are commonly controlled using PWM signals.. A PWM signal sent to a servo determines the position or angle of the servo's shaft, typically 0° to 180°. The duty cycle of the PWM signal sets the desired position, where a 5% duty cycle represents 0 degrees, and 10% represents 180 degrees[4]. When it comes to motor control, PWM can also be used to control the speed simply by coupling the motor to the duty cycle of the signal, where the motor is powered only during the presence of the square wave. PWM signals operate under a modulating frequency which changes based on application. To illustrate this, consider a lamp turned on for five seconds, then off for five seconds, representing a 50% duty cycle. The lamp would appear bright during the first five seconds and completely dim during the next five, not ideal for continuous use. To ensure the lamp receives enough power to be perceived to be in the “on” state, the cycle period must be shortened to achieve the desired effect of a dimmer yet continuously lit lamp. Here, it becomes imperative to raise the modulating frequency and use the duty cycle to control dimming. This principle holds true in various other applications of PWM, where typical modulating frequencies can range from 1kHz to 200kHz and are application-specific [4]. In our project, the receiver is configured to emit a PWM signal through one of its channels. Channels 1 through 6 on the receiver can be configured with the transmitter to convey directional commands such as forward, reverse, left turn, and right turn. Tying one of these channels to a capacitor, while simultaneously discharging through a resistor in parallel, results in a voltage that accurately reflects the duty cycle of the PWM signal, considering a constant modulating frequency. This voltage is directed to the input pin of the PIC18, where an analog-to-digital conversion (ADC) process is initiated. Consequently, the PWM signal is effectively communicated as it gets translated into an 8-bit value comprehensible to the microcontroller.

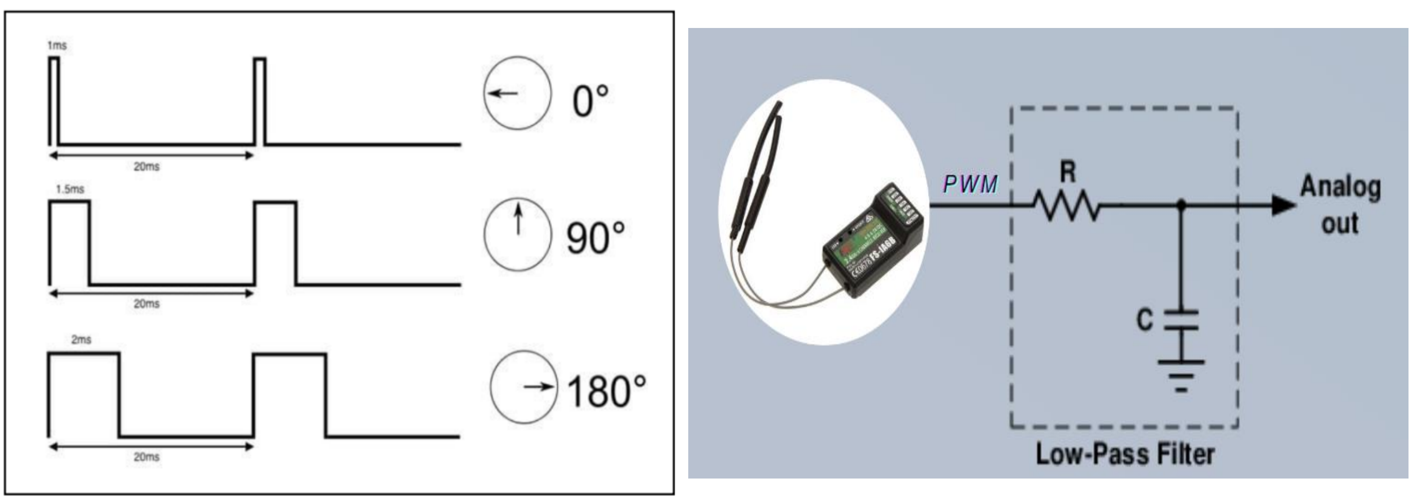


Figure 2: [left]PWM signal duty cycle to servo rotation angle. [right] PWM signal from receiver converted to analog signal for PIC18 microcontroller

Inter-Integrated Circuit (I2C) is another protocol used in our robot. We will be using it to communicate between the microcontroller and the IMU chip along with the load cell chip. This protocol allows numerous master and slave devices to connect to each other using only two wires. In our case, we will have a single master, the PIC18 microcontroller, and several slaves, IMU, and Load Cell Sensor. I2C, the serial data line (SDA) is used for sending and receiving data between the master and slave devices, while the serial clock line (SCL) carries the clock signal generated by the master. I2C is also synchronous, which means the output of bits is synchronized using the SCL clock signal generated by the master. Furthermore, the protocol is half-duplex: only the master or a slave device is sending data on the bus at a time. An I2C master device starts and stops communication, which removes the potential problem of bus contention. Also, communication with a slave device is sent starting with a unique slave address on the bus in the case where multiple slave devices are present. The communication begins with a start condition, denoted by the transition of the serial data (SDA) line from a high to a low state, while the serial clock (SCL) line shifts from high to low, signaling the initiation of communication. Then the address frame, composed of a 7-10 bits sequence, uniquely identifies the intended slave device to which the master is directing the message. This address must remain unique for each slave on the bus to avoid conflicts. Next, a read/write bit is sent, indicating whether the master is transmitting data to the slave (line low) or requesting data from it (line high). Each frame is accompanied by an ACK/NACK Bit, serving as an acknowledgment or non-acknowledgment bit. This bit is a feedback mechanism, providing confirmation of successful frame reception. The communication concludes with a stop condition, where SDA and SCL mark the end of communication [6]. These fields of the I2C frame ensure fast, synchronized, and reliable data transfer, addressing, and feedback within the protocol. When multiple devices are interconnected on a single I2C bus, the key to preventing data collisions and managing the clock line lies in specific strategies. Each master must first assess the state of the SDA line, whether it is in a low or high state, before initiating transmission. If the SDA line is low, it signifies that another master is actively using the line, requiring the waiting master to defer its operation. Furthermore, the protocol supports both 7 and 10-bit slave addresses, enabling communication with 128 slaves or up to 1023 slaves on a 10-bit bus [6]. These multi-master capabilities expand the versatility of I2C communication, permitting multiple masters to operate alongside single or multiple slave devices.

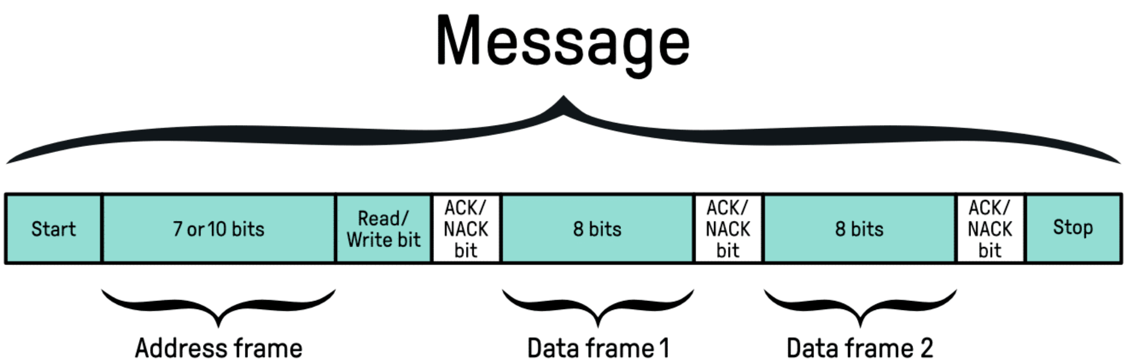
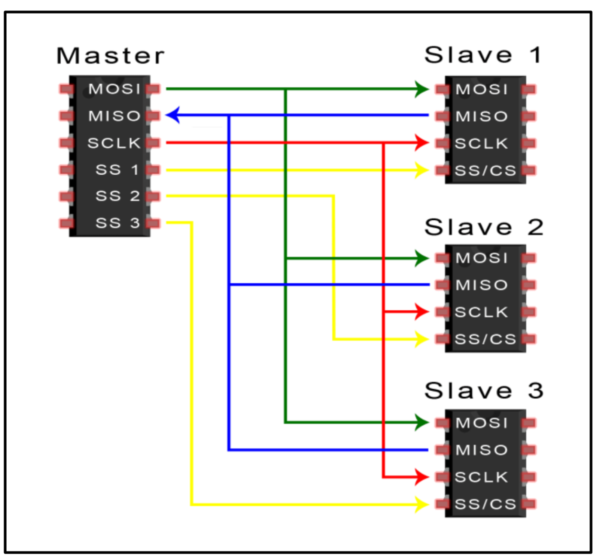


Figure 3: Above is a typical I2C message structure sent through an SDA line [6]

When programming the PIC18 with an editor, we require a chip programmer that can interface between the USB serial port of a computer and the microcontroller. The protocol with which the chip programmer communicates with the microcontroller is called Serial Peripheral Interface (SPI). SPI communication is a synchronous protocol, meaning devices share a common clock signal, and it is initiated by the master which generates the clock signal. Unlike some other communication protocols, SPI doesn't use start and stop bits. It allows for continuous data streaming without interruption, which is particularly beneficial for applications where data transfer speed is crucial. However, SPI does have some limitations, such as no built-in acknowledgment of successful data reception, no error-checking mechanisms, and it typically supports only one master. Nevertheless, its simplicity and high data transfer rates make it a suitable choice for many electronics such as SD card readers and 2.4GHz wireless applications, and for our purposes, as a chip reader for the PIC18.

In SPI, data is transmitted serially one bit at a time over dedicated lines: MOSI (Master/Output/Slave/Input) for data from the master to the slave, and MISO (Master/Input/Slave/Output) for data in the reverse direction. Additionally, there is an SCLK (Clock) line that synchronizes the data transfer between the master and slave, and an SS/CS (Slave-Select/ Chip-Select) line that allows the master to choose which slave device to communicate with. Some chips can support multiple slaves either tied to their own specific SS/CS & MOSI lines on the master chip or in the case where the master has a single SS/CS line, the MISO line is “daisy-chained” together [7]. Although fewer wires and pins are required, the daisy-chain method is slow in speed, requiring rewriting every slave with data even if the target is a single slave. Below are some of the architectures and chip configurations used in SPI.



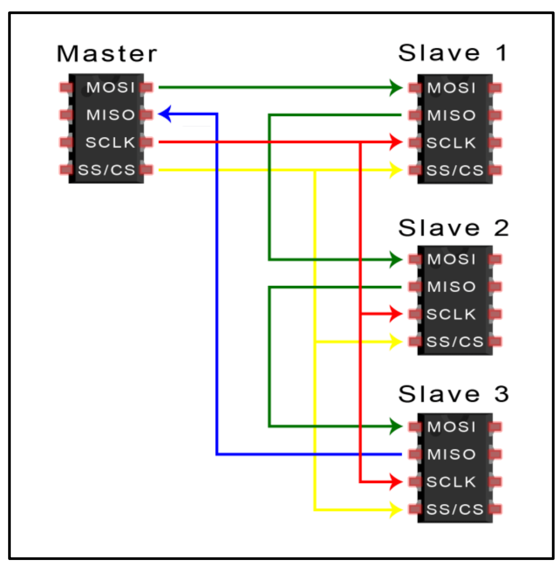


Figure 4: The master has multiple chip select pins to address each slave [left], master has single chip

The final communication protocol our robot employs is the STEP/DIR protocol which stands as the primary interface for controlling stepper motor drivers and ultimately motors themselves. It utilizes three signals—STEP, DIR, and ENABLE—to communicate with the microcontroller. STEP triggers a single-step rotation, with its frequency determining the motor's speed, while DIR sets the rotation direction. ENABLE enables or inhibits the driver, crucial for stopping the motor in a no-holding current mode. These signals are typically galvanically isolated to counteract interference and voltage shifts [3].

The STEP/DIR driver offers distinct advantages over other methods of driving stepper motors (using discrete MOSFETs, transistors, etc). For one, they stabilize phase current allowing connection to various stepper motors of coil winding resistances. They also facilitate a rapid rise of current in windings for increased rotational speed. The micro-stepping control mode ensures precise positioning with motors having limited physical steps. These drivers also provide protection against short circuits, optimize output switching for faster current drops, prevent overheating, and ensure high noise immunity through signal isolation [3]. Thus, the STEP/DIR protocol provides an efficient means of motor control between the microcontroller and stepper motor, with distinct signals governing rotation, direction, and driver enablement.

PID Control Theory and Balance:

We will explore the dynamics of the inverted pendulum, discussing its constant instability and the control strategies required for maintaining its stabilization. We will then examine the application of these principles to self-balancing robots, emphasizing the role of sensors and control algorithms, with a detailed focus on proportional-integral-derivative (PID) control. The paper provides a comprehensive explanation of the mathematical foundations and practical implementation of PID control in self-balancing robots, offering insights into the science behind these “magical” systems.

The inverted pendulum, an iconic example of a dynamically unstable system, has long captivated the imagination of engineers and researchers. Its seemingly simple yet highly unstable nature has made it a classic testbed for exploring control theory and advanced engineering solutions. The inverted pendulum consists of a mass (pendulum) mounted on a pivot point (fulcrum). In its upright position, it stands in a state of unstable equilibrium, a challenge to maintain without external support. The very essence of the inverted pendulum's dynamic instability mirrors the main challenge faced by self-balancing robots, whose purpose is to remain upright in the face of constant gravitational forces and external disturbances.

As previously stated, the inverted pendulum system, unlike many other control systems is naturally unstable [8]. When the pendulum is displaced from its vertical position, the natural restorative force acts in the same direction as the displacement, causing the pendulum to accelerate further away from the equilibrium that we want until it eventually just tips over. In order to stabilize this system, two fundamental approaches can be chosen from. The first option involves altering the direction of gravity, essentially changing the frame of reference. While this method might theoretically work, it is simply impractical and obviously impossible to implement in most real-world scenarios. The second and more realistic approach involves applying an external force to counteract the displacement and ensure that the restoring force is in opposition to the direction of the displacement. This method introduces controlled acceleration to the base of the inverted pendulum, in our case via the robot's wheels, allowing it to regain its vertical position. However, it's important to note that implementing this corrective force introduces an additional challenge. As the pendulum accelerates back towards an upright position, it experiences an inertial force in the opposite direction to the wheel's movement. The magnitude of this inertial force is directly proportional to the acceleration. In this way, controlling the base wheel of the inverted pendulum to generate the appropriate acceleration is a delicate balance of forces and dynamics. It's a complicated juggling act between controlling the external force applied and managing the resulting inertial forces, all hoping to achieve the ultimate goal of maintaining the inverted pendulum in a stable, upright position. This control strategy forms the foundation for self-balancing robots, allowing them to navigate any challenging terrain to maintain balance in a dynamically unstable environment.

For a better understanding, when the whole robot is connected by a rigid structure, the system structure model of the robot is established, as shown in Figure 5 [9], where L is the distance between the center of gravity of the body and the wheel axis, θ is the tilt angle of the robot, the mass of the body is m, and g is the acceleration of gravity.

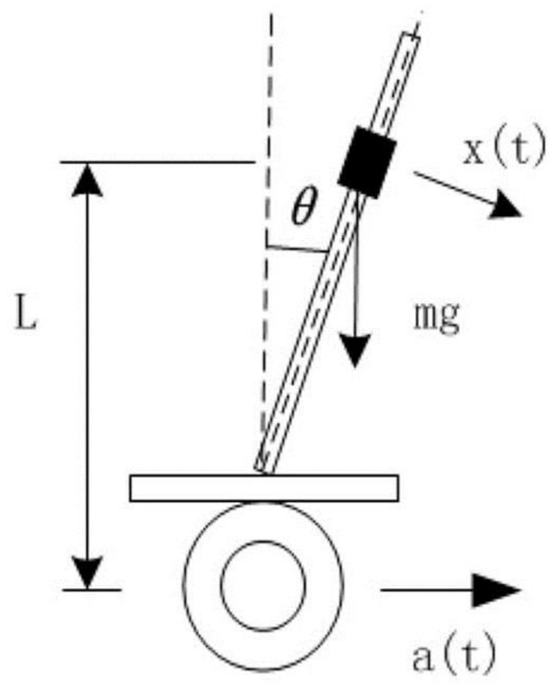


Figure 5: Force analysis of unbalanced robot

Assuming that the angular acceleration $x(t)$ is generated under the action of external force and the acceleration $a(t)$generated by the wheel, the force analysis is carried out on the robot model to obtain the motion model of the inspection robot:

$$L\frac{d^2\theta(t)}{dt^2}=g\sin[\theta(t)]-a(t)\cos[\theta(t)]+Lx(t) \tag{1}$$

In the range of small angles, the equation of motion can be linearized:

$$L\frac{d^2\theta(t)}{dt^2}=g\theta(t)-a(t)+Lx(t) \tag{2}$$

When the vehicle body is in a stable equilibrium state in the vertical position, and the inclination angle $\theta(t)$ is 0, then the equation becomes:

$$L\frac{d^2\theta(t)}{dt^2}=g\theta(t)+Lx(t) \tag{3}$$

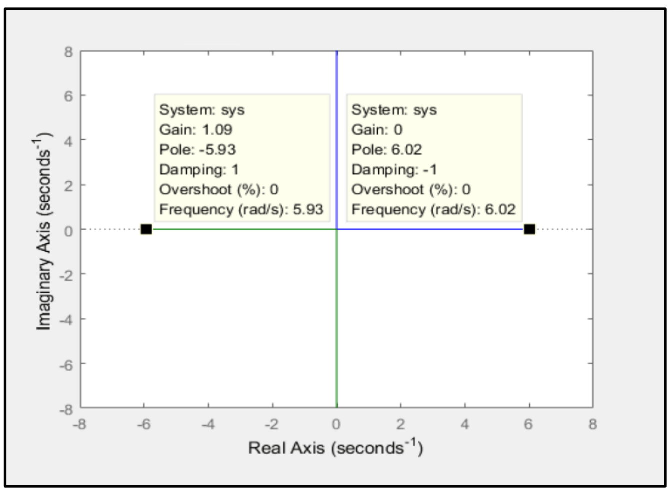
By transforming Equation (3), the system input–output transfer function is:

$$H(s)=\frac{\theta(s)}{X(2)}=\frac{1}{s^2-\frac{g}{L}} \tag{4}$$

According to the transfer function, the system has two poles, as shown below:

$$s_p=\pm\sqrt{\frac{g}{L}} \tag{5}$$

Substituting the gravitational acceleration, 9.8 m/s2, and assuming the center of gravity is roughly 0.27m from the axle center, the poles of the transfer function can be plotted on the real and imaginary axis. The result shows (graph on right [9]) that one pole of the system



is located in the right half plane of the system, and therefore the system is in an unstable state. This is where we introduce a negative feedback control that can shift the system from unstable to stable. In this case, we will be using a proportional-integral-derivative control, or as it will be referenced from here on out: PID control.

PID control is a control mechanism used to regulate and maintain a desired state or setpoint in various systems. It continuously adjusts an output based on the difference between the desired state and the current state of the system. This control method is effective in stabilizing and fine-tuning processes by considering how the system behaves in response to changes, making it a versatile and widely applied control strategy in fields such as engineering and automation. The proportional (Kp) component responds to the current error (the difference between the desired angle and the measured angle) and generates an immediate corrective action. The integral (Ki) component addresses steady-state errors by accumulating past errors and correcting over time. The derivative (Kd) component counteracts overshoot, reducing rapid changes in response to sudden errors. The mathematical representation of PID control is relatively straightforward. The control signal at any given time, U(t), can be defined as:

$$U(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt} \qquad (6)$$

where e(t) represents the error at time t, which is the difference between the desired angle and the measured angle. Kp, Ki, and Kd are the controller gains for the proportional, integral, and derivative terms, respectively. For clarity, controller gains represent the magnitude of each component's contribution to the control output. This parallel formula is also known as a decoupled PID form. This is because the PID controller has three decoupled parallel paths between the three variables. As can be seen in the following figure, a numerical change in any individual coefficient, Kp, Ki or Kd, changes only the size of the contribution in the path of the term. For example, if the value of Kd is changed, then only the size of the derivative action changes, and this change is decoupled and independent from the size of the proportional and integral terms. This decoupling of the three terms is a consequence of the parallel architecture of the PID controller [8],[10].

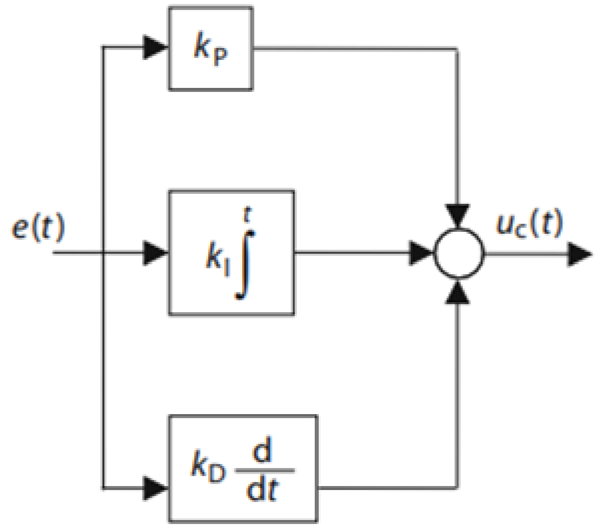


Figure 6: Parallel, time domain formula for PID controller [10]

Relevant Work

The domain of self-balancing robots has been an active area of research, with various innovative projects being undertaken worldwide. This section offers an overview of some recent developments in this arena, highlighting their significance and how our project distinguishes itself within this context.

The project "Control System for a Self-Balancing Robot" led by Martins and Nunes revolves around the development of a self-balancing robot called Bimbo. This project's central objective is to scrutinize the efficacy of various control algorithms in achieving equilibrium in self-balancing robots [11]. The project emphasizes the significance of the equilibrium control problem in two-wheeled mobile robots, a topic of high interest in the field of engineering due to its societal impact. The research explored different control algorithms, including the Proportional, Integral, and Derivative (PID) controller, pole placement, and adaptive control, among others. The project also tested an algorithm applied to the robot's position, creating a generic solution for self-balancing robots. The robot Bimbo was constructed with modules for movement and position control, and the authors implemented a Kalman filter to get the Roll angle from the Inertial Measurement Unit (IMU). The project also developed a mechanism to read encoders and control the two motors. What sets Bimbo apart is its mechanism to control system variables through Bluetooth communication, allowing continuous monitoring of any robot variable and real-time testing of the system and control variables. The importance of this project lies in its focus on developing a generic solution for self-balancing robots by experimenting with different control techniques. It also highlights the importance of continuous monitoring and real-time testing in improving the performance of self-balancing robots.

The paper "Fuzzy Control of Self-Balancing Robots" by Odry et al. introduces a unique control laboratory project that offers practical experience in feedback control concepts, particularly the design and implementation of fuzzy control. The project's core is a portable, cost-effective self-balancing robot (SBR), designed using commercially available breakout boards. This robot serves as a hands-on platform for students to experience the different stages of control system design, from system modeling and parameter optimization to the implementation and validation of the closed loop on the real robot [12]. In the second part of the paper, the authors introduce a control strategy based on fuzzy logic controllers. They describe a simple, lookup table-based implementation technique for manual interfacing and embedded coding of fuzzy control strategies. The authors emphasize that the proposed methods are clear, straightforward, and aid in understanding feedback control techniques and practical implementations of control systems. The project aimed to provide a methodological basis for the analysis of the controlled plants, detail some classical and advanced control techniques, and enable students to implement the acquired knowledge in controlling real plants. Furthermore, the solution under study aimed to be as generic as possible, offering versatility in its application. This project is of particular importance due to its focus on providing hands-on experience in feedback control concepts through dedicated assignments. This approach significantly enhances the learning experience and helps students recognize their professional interests, thereby improving their creativity and team management skills. Moreover, the project's emphasis on fuzzy control provides a novel perspective on control systems. Fuzzy control operates based on degrees of truth rather than the typical binary decision, making it a potentially more adaptable and flexible control method. The project also addresses the need for a more practical approach to learning, bridging the gap between theory and practice and providing a better understanding of fundamental concepts. It offers a unique opportunity to apply and test different control techniques on a real robot, thereby strengthening both social and professional skills in teamwork, problem-based thinking, and more.

Another paper, "Design and Implementation Control System for a Self-Balancing Robot based on Internet of Things by using Arduino Microcontroller", delves into the development of an autonomous self-balancing robot. This project stands out due to its integration of Internet of Things (IoT) technology with the traditional components of self-balancing robots, offering a glimpse into the future of robotics [13].

The project centers around a two-wheel robotic system composed of a microcontroller (Arduino), a DC motor, and sensors. The Arduino microcontroller reads sensor data and commands the motor based on a control algorithm to maintain system stability at different impediments. The system uses an Ultrasonic sensor to sense obstacles during movement and a 3-axis gyroscope accelerometer sensor to measure the robot's inclination angle. The controller laws in this project allow the robot to reach static or moving targets based on structured IoT interactions between the elementary controllers and the sensor with the actuator via a Cloud environment. The design's technical details are based on a mathematical model, which is used to create the system's transfer function. The MATLAB Simulink is used in the design of the controller, and a PID controller is used due to its simplicity and good activity in central systems. This project is of significant importance due to its integration of IoT technology with self-balancing robots. IoT technology allows for interconnected devices, which can greatly enhance the functionality and adaptability of robotics systems. This paper presents a practical example of how IoT can be leveraged in robotics, demonstrating its potential to revolutionize the field. In this project, the IoT technology allows for continuous monitoring and real-time testing of the system and control variables. This capability significantly enhances the robot's adaptability and responsiveness to different impediments, making it more efficient and reliable. Furthermore, the project highlights the potential of using IoT in the control of real plants, demonstrating how this technology can be applied in real-world scenarios. This practical application of IoT in control systems can provide valuable insights for future developments in this field.

Our project shares common ground with these concurrent developments, primarily in the design and application of self-balancing robots and advanced control systems. However, our project uniquely employs a PIC microcontroller, which is renowned for its robust performance and cost-effectiveness. Unlike the Arduino microcontrollers often used in other projects, the PIC microcontroller offers superior processing capabilities, making it an ideal choice for our self-balancing robot. Moreover, our project places significant emphasis on the educational value of the self-balancing robot. It serves as an engaging, hands-on learning platform for STEM concepts, particularly in the areas of robotics and control theory. This educational focus sets our project apart from many existing self-balancing robot projects.

Project Description

The project's objective is to craft a captivating and highly stable two-wheeled balancing robot, designed for both entertainment and practical utility, such as carrying a load. To achieve this, careful consideration was given to the robot's dimensions and weight. Opting for a height of approximately three feet, at "hip-level," and a width of about a foot, the robot aims for an optimal balance between stability and visual appeal. The decision to increase its height further enhances the robots stability based on the previously discussed kinematics, creating a visually striking impression as it masterfully balances its entire mass on a pair of wheels.

During the CAD model design phase, a friend drew a connection between our robot and "Rosey The Robot" from "The Jetsons" show, igniting inspiration. This prompted a deliberate shift in the robot's design direction to evoke a more friendly and servant-like appearance. The influence from this iconic character encouraged us to infuse the robot with an approachable and helpful aesthetic, aligning with the notion of creating a companion that not only excels in functionality but also resonates with a sense of familiarity and warmth. A rough 3D model was designed with this goal in mind and is depicted on the right (Rosey the Robot below).

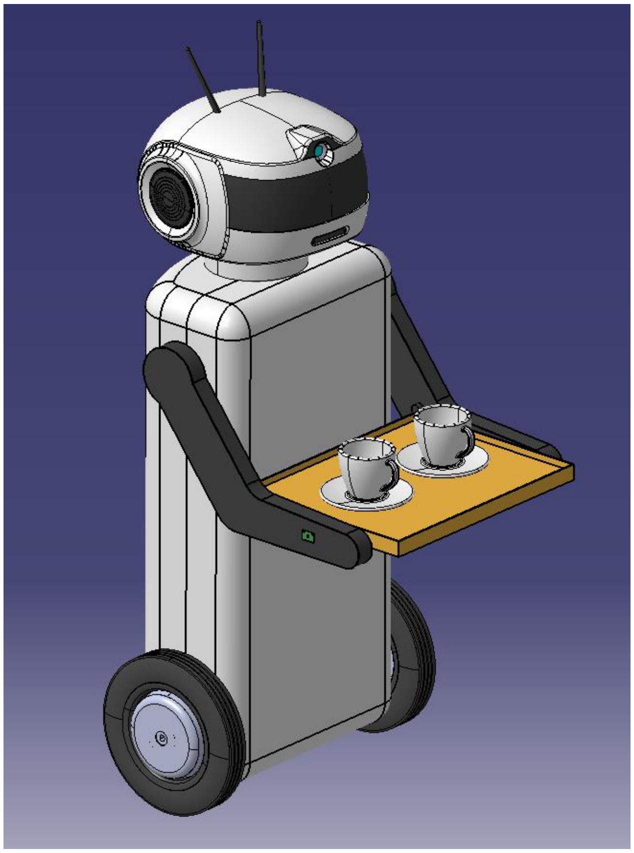
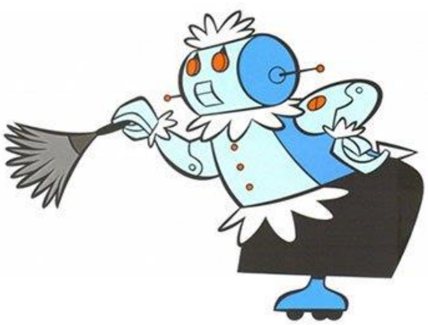


Figure 7: side profile of CAD design



How much can it carry? How much should it weigh? We arbitrarily decided that the robot should be able to carry around 1 to 2 kg of weight, enough to carry a few cups of tea and a meal. Given the expected load capacity, the rest of the robot's body should be substantially heavier to counteract the effects of an unbalanced load. However, too heavy a robot would be difficult to move and possibly even too dangerous; Too light, and we risk loss of control when a load is applied too far from the center of gravity. Therefore we struck a balance of power and weight: the robot would weigh around 7kg without load, and powered by two high torque stepper motors.

We chose stepper motors over DC brushed motors because they are much more precise and less likely to result in oscillations which is common in self balancing robots that use DC brushed motors. Furthermore, stepper motors have extremely high torque which makes them desirable for our project due to the weight of the robot, allowing for quicker reactions to counteract imbalances. Stepper motors come in sizes based on the NEMA standard, allowing for easy drop and replacement of motors. NEMA17 motors are typically used in 3D printers, but they are quite small and offer two little torque. The motor we chose would not only need high torque, but also should be able to handle the weight of the robot through its own construction, acting as the bearing of the axle. Going up a standard are the NEMA23 motors, offering substantially more torque up to 3 Newton Meters. Moreover, these motors boast a size and build quality well-suited for heavy machinery applications like lathes and CNC machines. Choosing these motors ensures robustness and heavy-duty performance, aligning with the demanding requirements of our robot. Thus, we went with the powerful

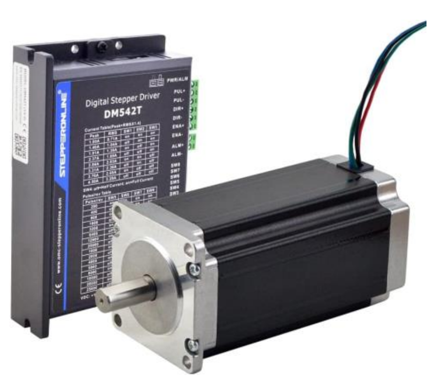


Figure 8: NEMA23 motor and driver

NEMA23 motors with their recommended drivers.

The recommended stepper motor drivers required a minimum of 20v and each stepper motor can draw up to 4.2 amps of peak power, so it was decided that two 12v batteries in series would be used resulting in an operating voltage of 24v. In terms of battery chemistry, we chose LiFePO4 Deep Cycle Batteries rated at 10AH and discharge rating of 3C. This means the batteries can supply a consistent 12A of power to the robot for about an hour, more than enough for our purpose.

Figure 9: 12v, 10AH LifePO4 Batteries

A major component that needs to be discussed is the two wheels that the robot will balance on. The motor shaft of the NEMA23 motors is 10mm so there needs to be a way to mount the wheel to the motors securely. We settled on a flange couple that could secure itself on the motor shaft of each motor, and also be bolted the hub of the wheels. We chose wheels first by the diameter we needed, then by their appearance and how closely the resembled our robots CAD design. It was decided that given the high 3NM torque of the stepper motors, a wheel with 5inch radius would offer the perfect power transfer from motor shaft to the ground. A 10x1.75" Semi-Pneumatic tire was selected due to its high traction and agreeable look.

Figure 10: Rigid Flange Coupling 10mm

Figure 11: 10x1.75" Tire

When we had decided the robot would carry a load, we had not thought about what kinds of items the robot would carry. Initially, the concept involved the robot carrying a fixed basket or tray that would rotate in tandem with its entire body. However, drawing inspiration from "Rosey the Robot," we opted for a design where the tray operates on its own rotating axis, stabilizing independently of the robot's orientation. This decision led to the arms serving as the rotating axis for the tray, necessitating additional mechanisms for arm/tray rotation. Despite the increase in moving parts and wiring, this design choice introduced an extra layer of care and friendliness to our robot. It ensures that the robot takes proactive measures to prevent items it carries from spilling or falling, enhancing its overall user-friendly and considerate nature.

The mechanism for controlling the tray's rotation was determined to involve a lever arm that pivots according to the position of a threaded rod. This threaded rod, situated nearby, is manipulated by another stepper motor. We opted for a low-power NEMA17 motor to control the threaded rod, as it is not subjected to substantial torque loads. Below is the CAD design of this mechanism along with mechanical components making up the arm/tray assembly.

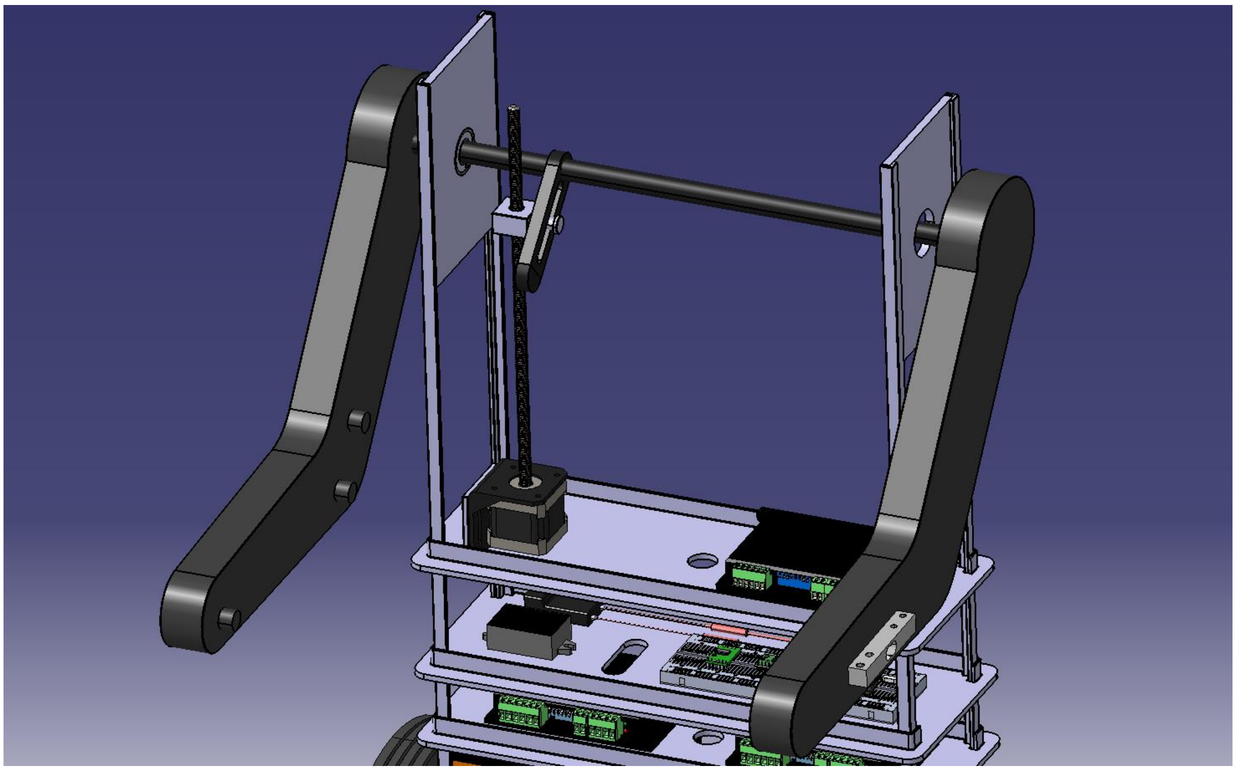


Figure 12: a threaded rod fed from the NEMA17 Stepper Motor Controls how much the arms /tray pivot. There are groves in the arm so that the tray can be removed and locked in place

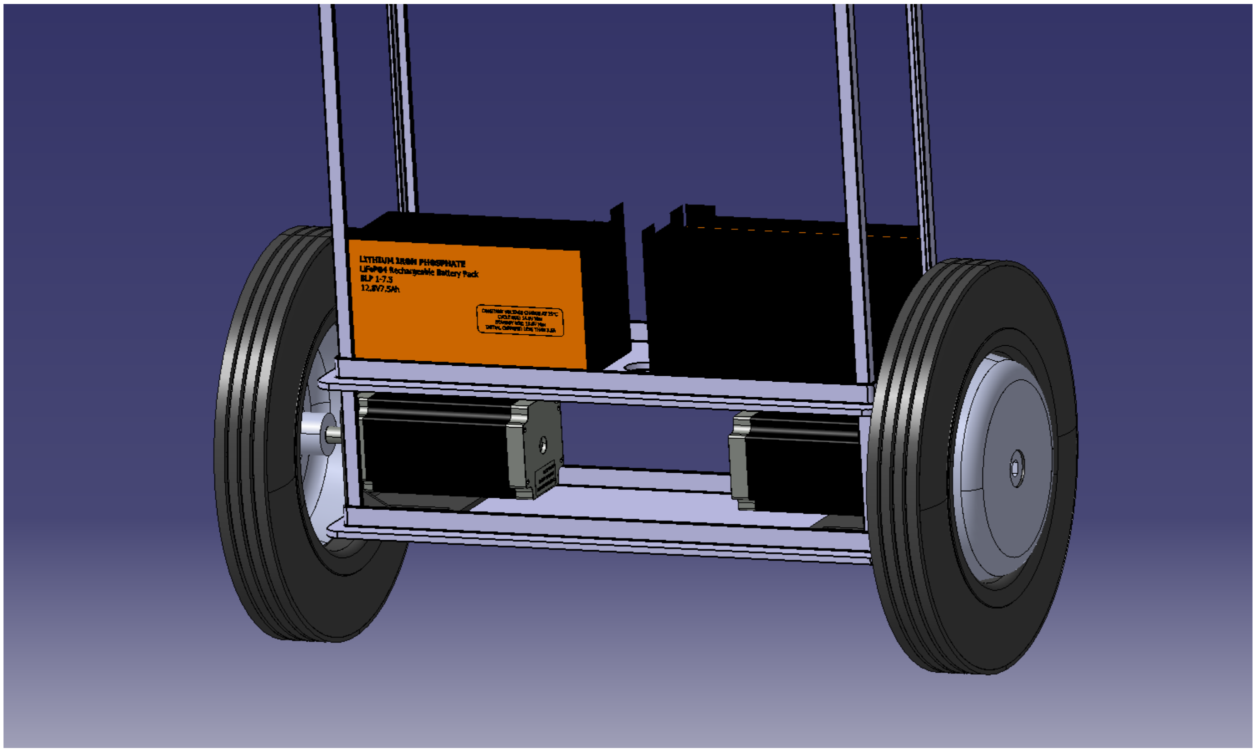


Figure 13: The CAD design above demonstrates how all of the power related pieces come together to make up the chassis and axle. The metal framing will be constructed from 2020 aluminum extrusion and the plates going up will be of steel sheet metal

Shifting focus to the electronics, the robot's 24V power supply requires conversion to 5V for the sensitive electronics and monitoring sensors. To accomplish this, a 15W Step Down converter was selected for effective regulation. The PIC18F4550 Microcontroller takes center stage as the robot's brain, boasting 35 I/O Pins and a 48MHz operating capability. It excels in communication through the I2C Protocol and swift Analog to Digital Conversions, overseeing all control signals.

For precise motion sensing, the MPU6050 serves as the Inertial Measurement Unit, delivering a 3-axis gyroscope and a 3-axis accelerometer within a single chip. Operating via I2C, it offers programmable ranges for components, making it a suitable choice for communicating with our robots microcontroller. There will be two MPU6050 chips used on this robot: one for reporting sensory data of the robots body to keep it upright, and the other chip will be implanted on the arms of the robot to keep track of the arm level so as to keep the arm level at all times. The robot integrates a 5kg load cell powered by an HX711, enhancing its ability to accurately measure and respond to varying weights. The HX711 serves as a precision analog-to-digital converter and can also send information using the I2C protocol, allowing the robot to process signals from the load cell with precision. This load cell will be instrumental in providing feedback to the PID Control loops that we will program into the microcontroller. Most likely we will use it to increase the gain of the derivative term so that the robot can increase its reactivness when it senses a load.

Completing the electronics setup, the FS-iA6B Receiver communicates with any compatible 2.4GHz transmitter. Outputting 6 channels through PWM, it establishes effective control and communication within the robot's operations.

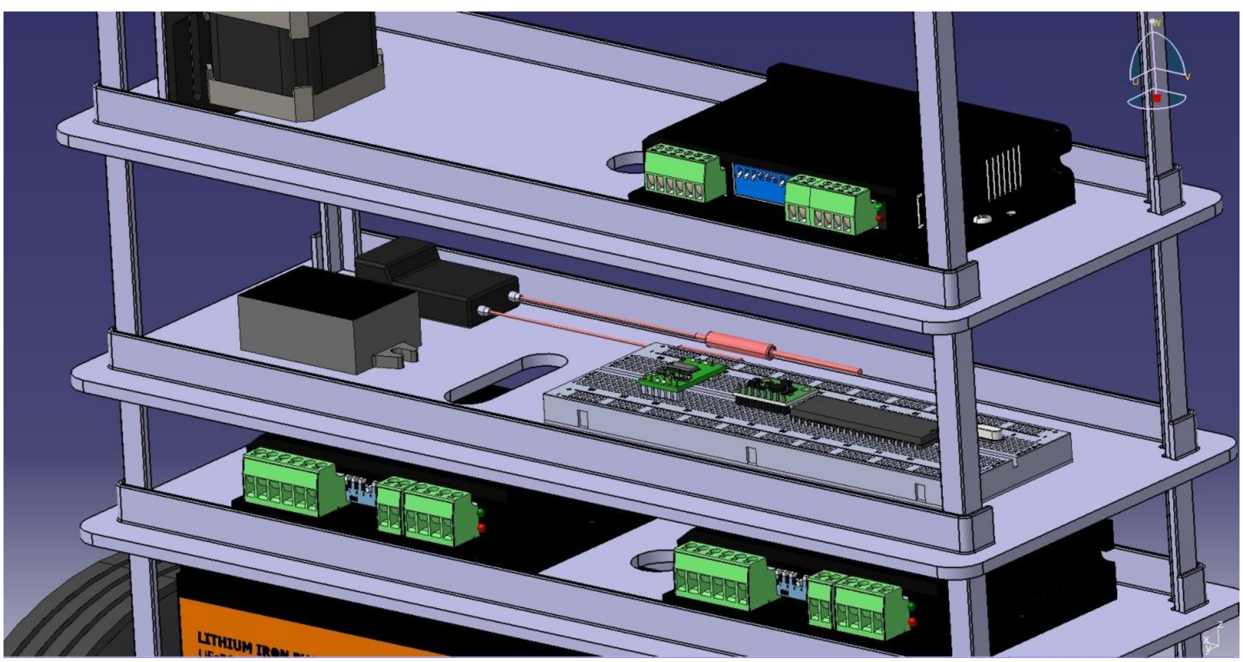


Figure 13: The image above shows most of the electronics that will be used in this robot. In the breadboard, there is the PIC18, HX711, and MPU6050. Further to the left we can see the buck converter and 2.4GHz reciever mounted above.

Ethical Implications

The development of self-balancing robots has gained significant interest in recent years, both from academia and industry. This emerging technology has various potential applications, from entertainment and personal mobility to healthcare and industrial settings. However, as with any new technology, ethical issues arise as we seek to integrate these robots into our daily lives. This essay will discuss the ethical considerations that must be taken into account when developing self-balancing robots and will address global, economic, environmental, and social concerns.

The global implications of self-balancing robots are closely tied to their widespread adoption and integration into various aspects of society. One ethical concern is the potential for increased military usage of robots, leading to the development of autonomous weaponry that could change the nature of warfare. While self-balancing robots may not initially be designed for such purposes, related technologies could be adapted for military use in the future. Another global concern is privacy and surveillance. As self-balancing robots become more prevalent in urban environments, there is a risk that they may be utilized for nefarious purposes, such as spying on individuals or organizations. The potential misuse of this technology raises questions about how to regulate its use to minimize unintended harm effectively.

From an economic standpoint, the widespread adoption of self-balancing robots may lead to job displacement in certain industries, particularly those reliant on manual labor or manufacturing processes. As this technology matures, it becomes more affordable and accessible to organizations and businesses around the world, potentially resulting in a reduction of human labor requirements. Consequently, job displacement raises ethical questions about how to ensure fair distribution of benefits from this new technology. Society must consider ways to retrain displaced workers or provide alternative opportunities so that they do not suffer undue financial hardship due to technological progress.

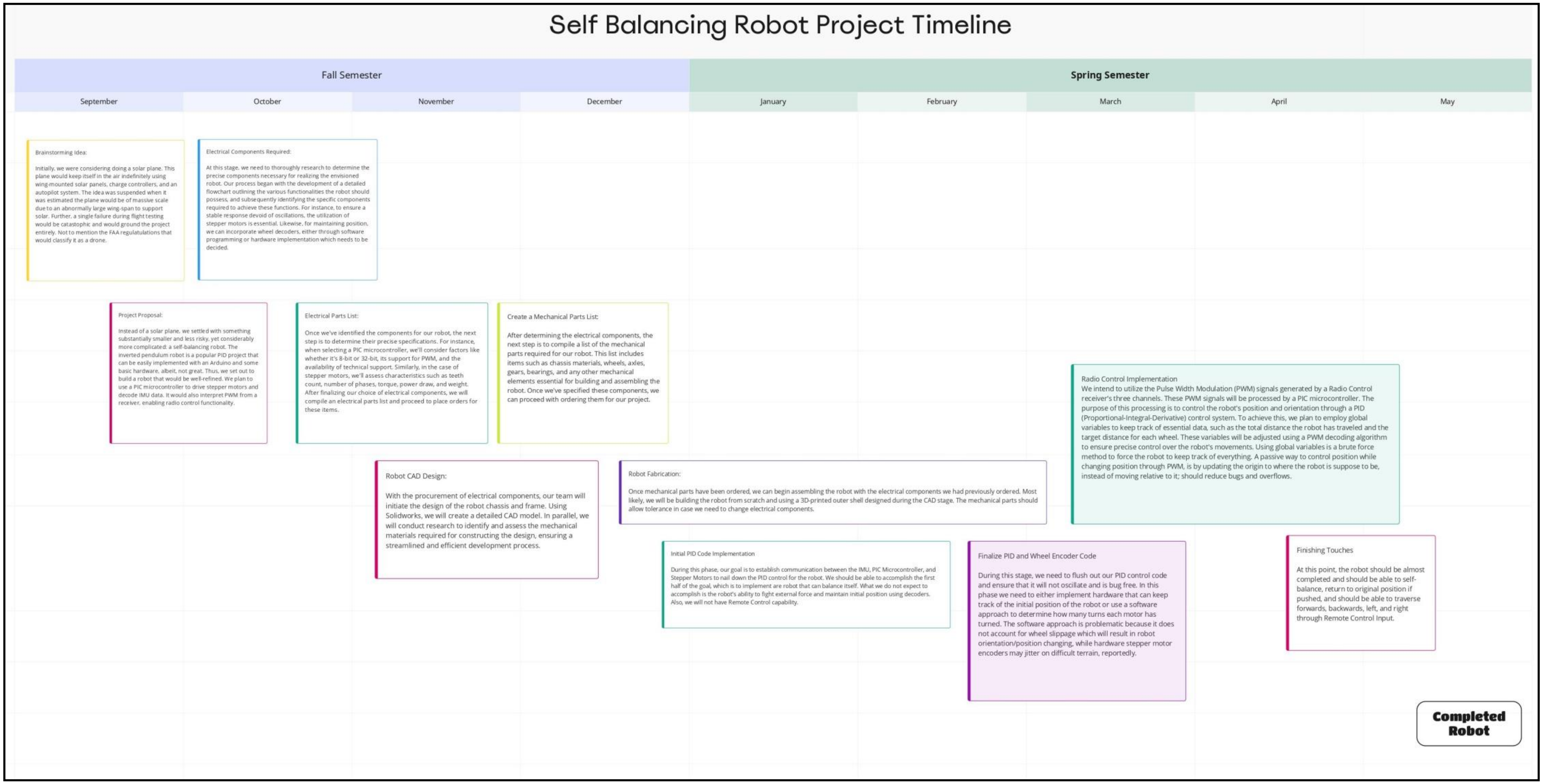
The production and usage of self-balancing robots also have implications for the environment. The manufacturing process, including the extraction and refining of raw materials, consumes considerable amounts of energy and resources. Additionally, robots require batteries and electronic components to operate, which may contribute to the growing problem of electronic waste disposal. It is crucial to develop sustainable manufacturing practices that minimize environmental damage and investigate renewable energy sources to power self-balancing robots. This includes considering the full life cycle of these robots and implementing strategies for recycling or disposing of them responsibly.

In addition to the aforementioned concerns, there are many social implications of self-balancing robots. As these machines become more integrated into our daily lives, issues of accessibility and equity arise. There is a risk that only certain segments of society will be able to afford these robots, exacerbating existing social inequities. Moreover, how we interact with these robots will affect our social relationships and self-perception. Humans may become reliant on self-balancing robots for activities previously requiring human interaction or skills, resulting in a diminished sense of self-efficacy or even isolation.

While ethical considerations should be an integral part of your project development process, addressing these concerns can help ensure a more responsible and beneficial impact on society. In creating a self-balancing robot, it is crucial to consider global, economic, environmental, social, and accessibility-related issues throughout the inception, design, implementation, and deployment stages. By actively considering potential harms and responsibly managing them, you contribute towards the development of a technology that serves the greater good.

Project Status

Below is the time line we adhered to throughout this semester. It also outlines what we should have completed in the spring semester.



According to the timeline, we should have brainstormed our idea and should have created a project proposal of our project. Obviously we have accomplished this goal since we have settled on the idea of a self-balancing robot because we find it is an interesting and challenging endeavor. Furthermore, we should have determined what electrical components we need for the robot and should have generated a parts list of all electrical components. Almost all of the parts we needed we found on Amazon.com and we filled the cart with all necessary electrical items we would need to by after researching them extensively. By the middle of November and beginning of December, we should have come up with a CAD design of the robot so that we could figure out what mechanical components would be necessary. At this time we should also have finalized all electrical components and we did ship all of the electrical items during this time. Alongside the CAD design, we should also finalize a mechanical parts list, which we have already done and have placed orders on all mechanical items. All of these items we have accomplished on time. From now until the end of February, we need to fully fabricate the robot and begin writing code to microcontroller. Below is a complete parts list of both mechanical and electrical components that have been ordered and are in hand now ready for assembly. There are only a few items that have not been placed which are the loadcell and the horizontal plates that make up the frame of the robot.

	Parts List			
#	Part Name	Price	Quantity	Total
1	*YVSPTIK 57mm NEMA23 Stepper Motor Mounting Bracket w/ M4 Screws (3 PCS)*	13.99	1	13.99
2	STEPPERONLINE Nema 17 Stepper Motor Bipolar 2A 59Ncm(84oz.in)	13.99	1	13.99
3	XMHF 2 Set 15mm FL002 Zinc Alloy Self Aligning Pillow Block Flange Bearing	9.79	1	9.79
4	2020 Aluminum Extrusion Connector Bracket Corner Brace Set	9.99	1	9.99
5	CNCYEAH Aluminum Profile Extrusion 2020 1000m 4PCS	31.39	1	31.39
6	DMiotech Aluminum Round Rod Bar 15mm Diameter 500mm Length	19.99	1	19.99
7	uxcell 10mm Inner Dia H13*D16 Rigid Flange Coupling Motor Guide Shaft Coupler 2 PCS	11.49	1	11.49
8	Usongshine TB6600 4A 9-42V Nema 17 Stepper Motor Driver CNC Controller	9.98	1	9.98
9	JCSPBYL 20pcs Silver 2020 Series T Slot L Shape Corner Bracket	9.99	1	9.99
10	Marathon 10x1.75" Semi-Pneumatic Tire on Wheel	27.06	2	54.12
11	CNBTR 0.5 Modulus Brass Metal Speed Reducer with 60 T Wheel 5mm Bore Gear Shaft	16.00	1	16.00
12	KUMGROT 42mm Stepper Motor Mounting Bracket Nema 17	9.59	1	9.59
13	HOMEXO Worm Gear and Worm 0.8M Precision Turbine Worm Gear Set 1:60 Teeth Stainless Steel	24.43	1	24.43
14	RadioMaster TX16S Mark II Carbon Fiber Edition 2.4GHz Radio Transmitter	219.99	1	219.99
15	AUKENIEN Quartz Crystal Oscillator Kit 15 Values	9.90	1	9.90
16	TYUMEN 50FT 12/2 Gauge 2pin 2 Color Red Black Cable Hookup Electrical LED Strips 12V/24V DC 12AWG	17.99	1	17.99
17	Nilight 12PCS 12V Blue Round Toggle LED Switch 20A 12V DC On/Off SPST	6.29	1	6.29
18	DC 12v 24v to 5v Step Down Converter Regulator 5A 25W	12.99	1	12.99
19	*NERMAK 2 Pack 12V 10Ah Lithium Ion LiFePO4 Deep Cycle Battery*	74.99	2	149.98
20	Gy-521 MPU-6050 MPU6050 Module 3 Axis Analog Gyro Sensors+ 3 Axis Accelerometer Module	5.99	1	5.99
21	*Flysky FS-iA6B Receiver 6-Channel 2.4G PPM Output*	16.77	1	16.77
22	Goldby EC5 Connector, 2-Pairs EC 5 Battery Connector Male Female with 10cm 12AWG	7.99	1	7.99
23	STEPPERONLINE 1 Axis CNC Kit 3Nm(425oz.in) Nema 23 Stepper Motor & Driver	64.00	1	64.00
24	PICcircuit iCP01 - USB Microchip PIC Programmer (with ICSP & PICkit 2 SW)	26.90	1	26.90
25	PIC18F4550-I/P PIC18F4550 18F4550 USB Microcontrollers DIP40 IC PIC MCU Flash 16KX16 New 1PCS	7.48	2	14.96
			Total TAX	75.13
			TOTAL	863.62

Conclusion

An ongoing desire to learn more about robotics and control systems drove the development of a self-balancing robot with PID control using a PIC microprocessor. From the beginning of the project, the fundamental goal was very evident: to create a robot that could balance and remain stable on its own. This project was inspired by a deep fascination in the complex interactions between hardware and software, as well as a strong desire to see abstract ideas materialize into working, working gear.

The goal of achieving movement control precision and accuracy motivated every stage of this project. Because of this, it became necessary to install encoders or stepper motors on the wheels to enable the fine-tuning necessary for precise motion control. The project sought to not only to build a robot that could balance itself skillfully while also making sure it stayed motionless when necessary—a demonstration of the care and attention to detail that went into the project's planning and execution.

Additionally, the effort to incorporate radio control functions added another level of complexity, but it was a task that was gladly accepted in order to improve the robot's versatility and broaden its applications. Even though it wasn't the top priority, this addition demonstrated the project's dedication to adaptability and potential growth.

In conclusion, this endeavor is evidence of the never-ending interest and commitment to creativity that advances the robotics industry. Beyond the accomplishment of building a self-balancing robot with PID control, this project highlights the value of ongoing education, flexibility, and the inner fulfillment that comes from turning abstract ideas into real, working objects. This project will not only create a working robotic system but can also lay the groundwork for further developments that will fuel the never-ending search for automation and technological innovation. There are obstacles to be conquered along the way, and every one of them offers a chance to learn something.

Citations

[1] “Introducing the Segway Human Transporter,” Provoke Media, .https://www.provokemedia.com/latest/article/introducing-the-segway-human-transporter
[2] “Handle,” Robots Guide, https://robotsguide.com/robots/handle
[3] Lesson 34: Step/Dir Stepper Motor Drivers - Basic Concepts of Step/Dir Protocol,” My Practichttps://mypractic.com/lesson-34-step-dir-stepper-motor-drivers-basic-concepts-step-dir-protocol/
[4] MDPI. “Sensors | Free Full-Text | A Review of Wireless Sensor Technologies and Applications in Agriculture and Food Industry: State of the Art and Current Trends.” Sensors, vol. 18, no. 1, 2018, p. 93, https://www.mdpi.com/1424-8220/18/1/93.
[5] Kester, Walt. “UART: A Hardware Communication Protocol for Microcontrollers.” Analog Dialogue, vol. 39, no. 3, 2005, https://www.analog.com/en/analogdialogue/articles/uart-a-hardware-communication-protocol.html.
[6] “How I2C Works and I2C Bus Protocol Explained with Examples.” The Geek Pub, 2019, https://www.thegeekpub.com/18351/how-i2c-works-i2c-explained-simply/.
[7] “Basics of the SPI Communication Protocol.” Circuit Basics, 2015, https://www.circuitbasics.com/basics-of-the-spi-communication-protocol.
[8] Junoh, Suhardi Azliy Bin. “TWO-WHEELED BALANCING ROBOT CONTROLLER DESIGNED USING PID” January, pp. 1, 23-24, 2015.
[9] Zhao, J.; Li, J.; Zhou, J. “Research on Two-Round Self-Balancing Robot SLAM Based on the Gmapping Algorithm”. Sensors 2023, 23, 2489. https://doi.org/10.3390/s23052489
[10] Johnson, Michael A., and Mohammad H. Moradi. PID control. London, UK: Springer-Verlag London Limited, 2005.
[11] Martins, R. S., & Nunes, F. (2017). Control system for a self-balancing robot. 2017 4th Experiment@International Conference (exp.at’17). doi:10.1109/expat.2017.7984388
[12] Odry, Á., Fullér, R., Rudas, I. J., & Odry, P. (2020). Fuzzy control of self-balancing robots: A control laboratory project. Computer Applications in Engineering Education. doi:10.1002/cae.22219
[13] Ahmed, Ehsan Qahtan. Design and Implementation Control System for a Self-Balancing Robot Based on Internet of Things by Using Arduino Microcontroller. (2021) Periodicals of Engineering and Natural Sciences (PEN) 9(3):409-417 DOI:10.21533/pen.v9i3.2178