Wireless Remote Motor Controller

ECE 445 Design Document - Fall 2023

Professor: Arne Fliflet

TA: Jason Zhang

Group # 20

Aaron Chen, Kyungha Kim, Leeboon Sheng

Contents

1. Introduction

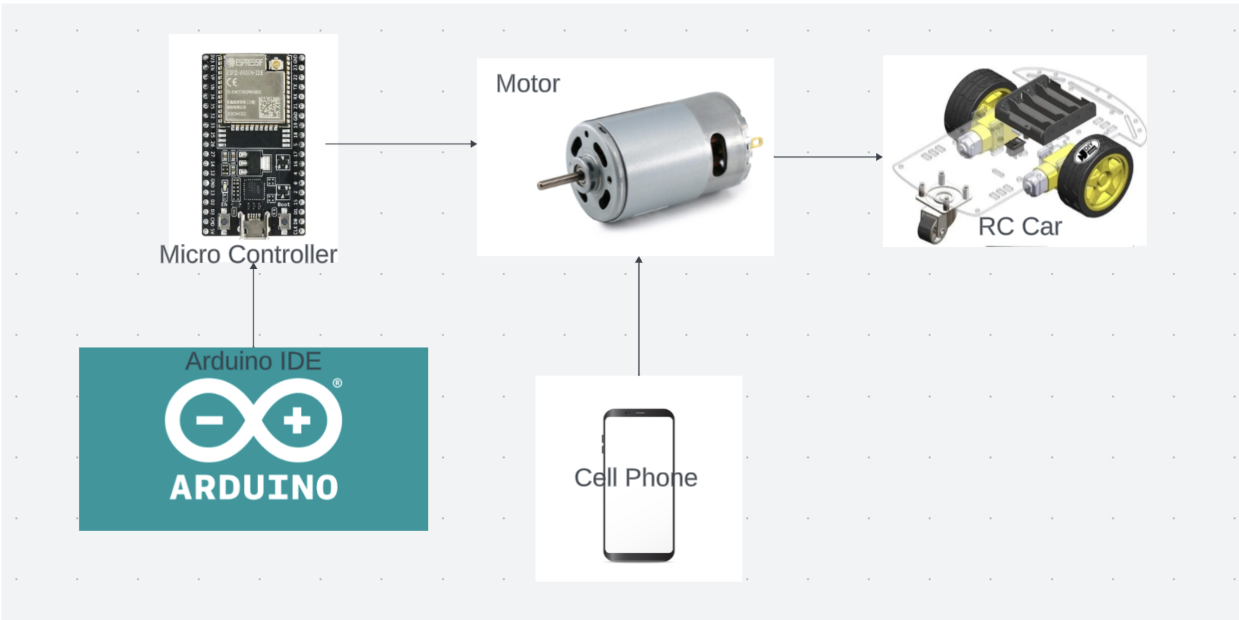
1.1 Problem

The need for efficient and convenient motor control is prevalent in various applications, such as robotics, automation, and remote-controlled vehicles. Existing solutions often lack simplicity and ease of use, making them less accessible to a broader range of users. Therefore, there is a demand for a wireless remote motor controller that is simple, user-friendly, and suitable for a variety of applications, including robotics and small wireless carts.In today's rapidly advancing technological landscape, motors play a pivotal role in the functioning of countless devices and systems. Motors are the driving force behind robots that automate tasks in factories, drones that survey remote areas, and small wireless carts that navigate through crowded environments. As these applications become more commonplace, the need for efficient and user-friendly motor control solutions becomes increasingly critical. One of the primary challenges in motor control is the complexity of traditional systems. Many existing motor control solutions require users to navigate through intricate interfaces, memorize numerous button combinations, or undergo extensive training to operate effectively. This complexity not only limits the accessibility of motor control but also introduces the risk of errors and accidents, especially in high-stress environments. The demand for simplicity and ease of use in motor control is particularly evident in the field of robotics. As robotics technologies continue to advance and become more accessible to a broader audience, there is a growing need for intuitive motor control interfaces. The rise of remote-controlled vehicles and drones has created a need for wireless remote motor controllers that can operate seamlessly and reliably. These applications often involve real-time decision-making, precision control, and the need to adapt to changing environments. A simple and responsive motor controller can be the difference between a successful mission and a failed one. The concept of simplicity in motor control extends beyond robotics and automation. It also encompasses applications in everyday life, such as remote-controlled toys and gadgets. Users, especially children and casual hobbyists, should be able to enjoy the benefits of motorized devices without the frustration of navigating complex control interfaces. To meet these diverse needs for simplicity and user-friendliness, a wireless remote motor controller is required. Such a controller should prioritize ease of use, responsiveness, and versatility to cater to a wide range of applications.

1.2 Solution

Our project envisions the creation of a Wireless Remote Motor Controller, addressing the pressing need for a user-friendly, versatile, and adaptable motor control solution. With a primary goal of delivering an adjustable speed range of 0 to 100%, this controller aims to redefine motor control capabilities for various applications, from industrial automation to hobbyist robotics. The core of our controller's design is its wireless functionality. It will be compatible with wifi, allowing users to operate motors from a distance without the hassle of wired connections. This wireless control not only enhances convenience but also reduces the risk of tripping hazards and enables the use of motors in diverse and dynamic settings. Simplicity is paramount in our design philosophy. The controller will feature a user-friendly interface with essential functions like start, stop, accelerate, and decelerate. These functions will be accessible wirelessly through an app that can be downloaded directly through your phone, ensuring that users of all skill levels can operate motors effectively. Recognizing the demand for more complex applications, we will explore an alternative design that enables the control of a pair of motors. This dual motor control capability opens up exciting possibilities for steering mechanisms, making it ideal for building highly efficient robotic platforms or small wireless carts. To enhance precision and performance, our controller will implement closed-loop speed control. This feedback mechanism ensures that the motor operates at the desired speed regardless of external factors, contributing to consistent and reliable performance. Our controller will incorporate current limiting control to prevent overloading, protecting both the motor and the user from potential hazards. This feature contributes to the long-term durability and reliability of the motor control system.

1.3 Visual Aid



Visual Aid

1.4 High-level requirements

1) Wireless Control
    a) The system must provide reliable wireless control of the motor e.g. WI-FI to control the DC motor remotely using a phone App. The ESP32 microcontroller will be programmed using Arduino IDE. This would allow users to operate it remotely from 50 to 200 meters away. This ensures improved convenience and accessibility for various applications, meeting the primary goal of wireless motor control. The wireless communication should be robust, with minimal latency and a stable connection to guarantee user satisfaction and safety.

2) Motor Controller Specifications
    a) The motor controller we design will be used with brushed DC, DC gear motors, and many linear actuators. The brushed DC motor controller must support operation under a specified input voltage range, with a preference for 12-24V DC and a maximum output voltage of 24VDC. This motor controller should be able to

supply a maximum current rating of 10A. This current rating is relatively high because it has to be about double of the motor's continuous operating current. The maximum output power of the DC geared motor should be 240W. This ensures compatibility with a wide range of motors commonly used in industrial and hobbyist applications like electric vehicles, home electric appliances, and levitation systems. Voltage stability and regulation should be a priority to prevent motor damage and ensure consistent performance.

b) The controller will utilize pulse width modulation(PWM) drive to regulate the DC voltage level output to the motor. A H-Bridge circuit will be incorporated in the motor controller to enable us to switch the DC motor on in a series of pulses. By controlling the duty cycle of the PWM, we can control the motor speed from 0-100%.

3) Speed Control

a) The motor controller must offer precise and adjustable speed control ranging from 0 to 100%. For easy control of speed and the direction of rotation of the DC geared motor, we have decided to use wireless methods particularly the RF technique. To achieve this, pulse width modulation(PWM) method is employed to control variable dc voltage for varying the speed of the DC geared motor. PWM is reliable, offers high performance, and low cost. This feature ensures that the controller can accommodate a wide range of tasks, from high-precision industrial operations to creative hobbyist projects.

b) The controller should maintain consistent speed control even when subjected to external factors such as load changes or variations in power supply voltage. Closed-loop speed control algorithms and feedback mechanisms should be implemented to achieve this requirement, guaranteeing reliable and stable motor performance.

4) User-Friendly Interface

a) The remote control interface should be intuitive and user-friendly, with clear labeling and easily recognizable buttons for functions such as start, stop, accelerate, and decelerate. Users of all skill levels should be able to operate the controller without extensive training or technical knowledge.

b) The controller should provide feedback to the user, such as visual or auditory indicators, to convey important information about the motor's status, including speed, direction, and power. This feedback enhances user confidence and ensures safe and efficient operation.

2. Design

2.1 Physical Design

This description outlines the key components and features of the controller's physical design, including two motors, wheels, chassis, a ball caster, and other essential elements. The motors are responsible for driving the wheels and, by extension, the entire system. They are placed on the chassis to provide the necessary power and control to move the device. We selected a brushed motor as controlling the speed of a brushed motor is relatively straightforward. By adjusting the voltage applied to the motor, you can control its speed easily without complex electronic circuitry. In addition to that brushed motors are inherently reversible. If you reverse the polarity of the voltage applied to the motor, it will change direction.

Image of the specific motor we plan to use

The wheels are essential components that enable the controller to move smoothly and efficiently. The design incorporates two wheels, each directly attached to one of the motors. The choice of wheels will depend on the intended application and terrain. We decided on these wheels as they were readily available and we were not too concerned about the type environment of where the car would operate.

Image of the wheels

The chassis serves as the structural frame of the controller, holding all the key components together. It is designed to accommodate the battery packs, PCB components, and other necessary electronics. The chassis is constructed from materials that balance strength, weight, and durability. It should be rigid enough to support the components and withstand any mechanical stress while remaining lightweight for optimal mobility. The chassis is specifically designed to house the battery packs, which provide the necessary power to the motors and other electronic components. Careful consideration is given to the placement and secure fastening of the battery packs to prevent unintended movement or damage.Within the chassis, the printed circuit board (PCB) components are strategically placed and securely mounted. These components include the motor control circuitry, the wireless communication module, voltage regulation components, and safety features. The layout of the PCB is designed for efficient heat dissipation and to minimize electromagnetic interference. The physical design may also include an enclosure or cover to protect the internal components from environmental factors, dust, and potential impacts. Accessibility features, such as removable panels or hatches, are incorporated to facilitate maintenance and component replacement.

To enhance stability and maneuverability, the physical design incorporates a ball caster. The ball caster is positioned at the front of the chassis and provides a point of contact with the ground that helps distribute weight evenly and allows for smooth directional changes. It ensures that the controller can pivot and navigate efficiently, especially when turning or changing direction.

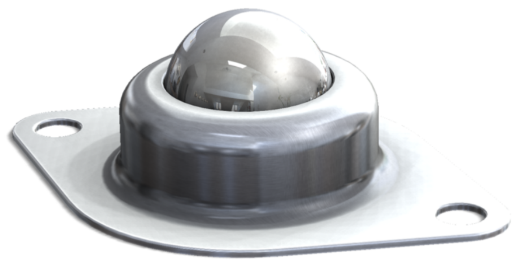
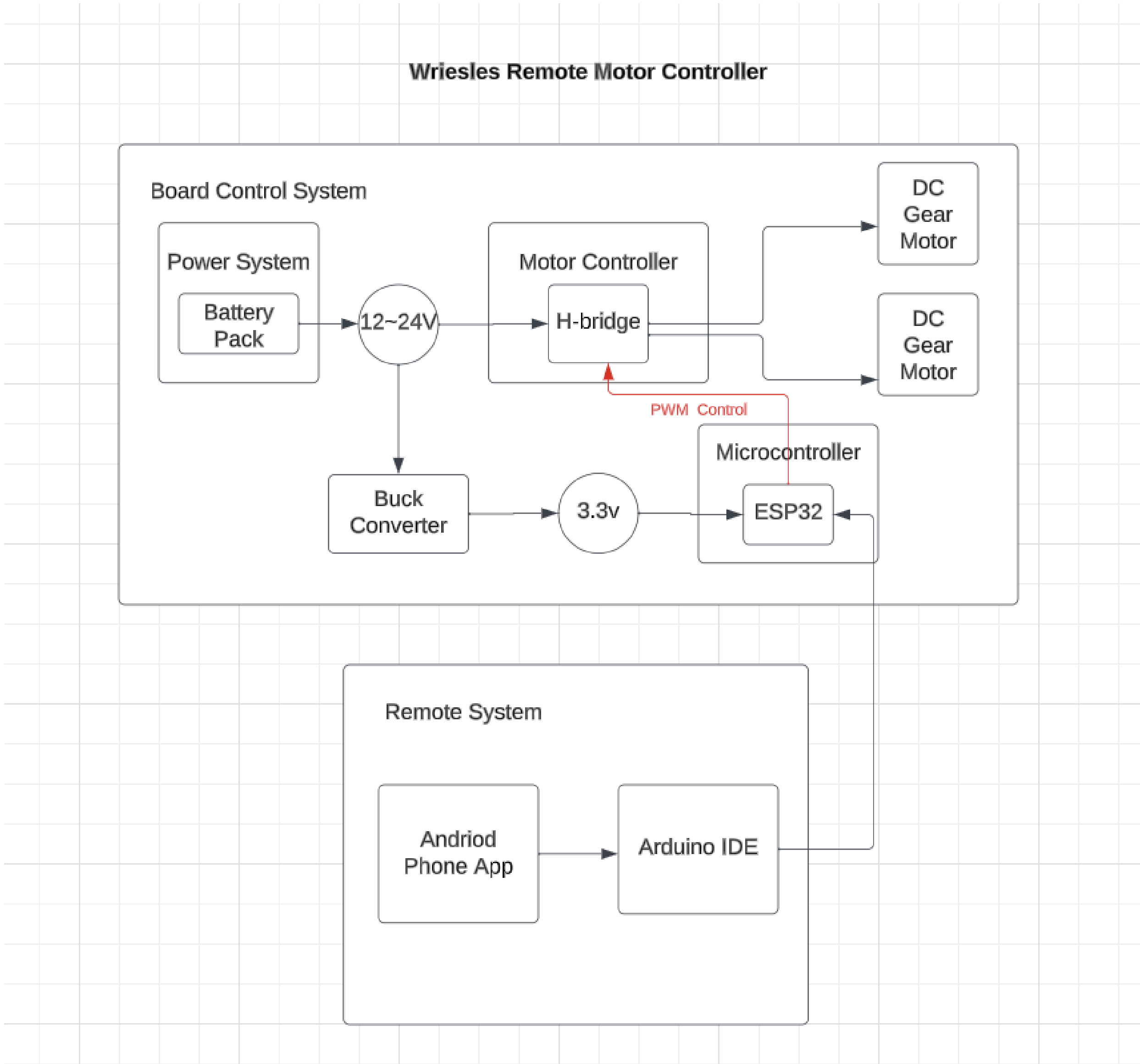


Image of the ball caster wheel

2.2 Block Diagram



Block Diagram

2.3 Functional Overview & Block Diagram Requirements

2.3.1 Board Control Subsystem

The Board Control Subsystem is responsible for receiving transmissions from the remote, taking data from the sensing subsystem, and commanding the electronic speed controllers. Based on these inputs, the board control subsystem will command the electronic speed controllers to power the motors accordingly. The Board Control Subsystem determines whether the DC geared motor will accelerate or decelerate and move in either forward or backward direction.

The microcontroller employed for this project is the ESP32S that interfaces with the motor controller. 3.3V DC supply is designed for the ESP32S microcontroller and it is used to feed PWM signals to the motor driver. PWM is achieved by varying the pulses applied to the enable pin of the H-bridge to control the applied voltage of the motor. The board system will then wirelessly communicate with the remote system via an Android Phone App designed using Arduino IDE. The DC gear motor will be driven by the H-bridge circuit. For more information on the software design of the Board Control Subsystem, please refer to Section 2.5.1.

2.3.2 Motor Controller Subsystem

The H Bridge includes four MOSFETS, two gate drivers for each side of the bridge with the associated bootstrap capacitors, and an Arduino Uno used to create the PWM signals that are fed into the gate drivers. The figure below shows the basic circuit of an H Bridge circuit. When Q1 and Q4 are on, the left lead of the motor will be connected to the power supply(battery pack for RC Car demo) and current will start flowing in the forward direction and the DC gear motor shaft will start spinning. Conversely, when Q2 and Q3 are turned on, the current will flow in the opposite direction and the dc geared motor shaft will start spinning backwards.. The top-end of the bridge will be connected to a power supply and the bottom-end of the circuit is grounded. In our H-Bridge design, we will be using four N-Channel MOSFETS to act as voltage-controlled. For the MOSFET selection, it is important to consider the Drain-to-Source resistance of the device when the device is operating in the active region. For this reason, we will be selecting the IRF530 N-Channel MOSFETs because the average series resistance is 200 milliohms. Therefore, the maximum power dissipation across this device would be:

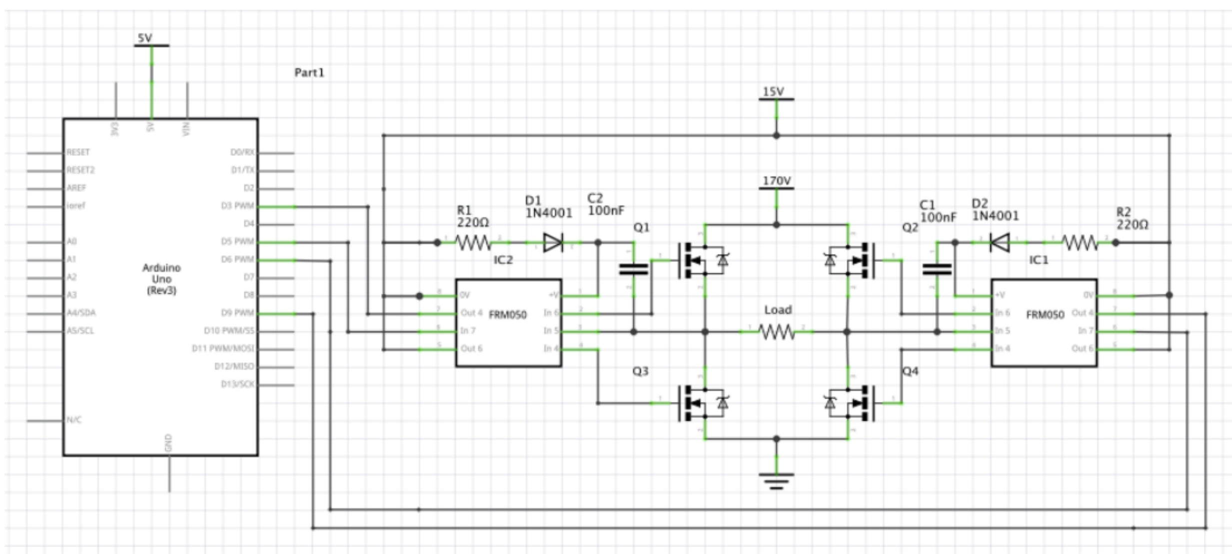
(.20 Ohms * 10 Amps) * 10 Amps = 20 Watts.

It is safe to assume that the IRF530 will successfully operate within the calculated current specification because the highest possible power dissipation across the device is 48 Watts according to the datasheet.

Besides, we will have to consider the Gate-to-Source Threshold voltage required by the device to switch the MOSFET on. Referencing the datasheet for the IRF530 N-Channel MOSFET, the Gate-to-Source Threshold Voltage (represented as VGS(th)) is on 4 Volts when 250 microamps are flowing at drain. Furthermore, we need a gate driver to create a high enough charge to activate the high side MOSFETs in an H-Bridge. Let's say our source voltage is 20V and VGS(th) is 4 volts, the voltage applied to the gate of the high side driver must be:

4 Volts + 20 Volts = 24 Volts

In order to activate the High Side drivers, we will have to apply 24V to the gate. If a gate driver is used in the design of an H-Bridge then the IC itself has a built-in charge pump that can be used to amplify a charge that will in turn trigger the high side MOSFET. This internal charge pump is combined with a bootstrap capacitor that supplies the required charge needed to activate the high side drivers. The diagram below shows the full H-Bridge control schematic.



Full H-Bridge Control Schematic

2.3.3 Power Subsystem

Power System for Motor Controller:

The power system will be responsible to efficiently manage the energy source, provide stable voltage levels for various components, and incorporate safeguards to protect sensitive electronics. In our design it will comprise of a 12V DC battery input, a 3.3V solder jumper coupled with a buck converter for the ESP32-S3 module, and a battery voltage sensor with Schottky diodes to safeguard the system's stability and safety.

12V DC Battery Input: The Heart of the Power System

At the core of the power system lies the 12V DC battery. This will serve as the primary energy source for the entire controller. The choice of a 12V DC battery is deliberate, as it can be easily balanced between providing sufficient power for motor operation and being a common and readily available voltage source. This voltage level aligns with the requirements of many motors, making it an ideal choice for a wide range of applications.

3.3V Solder Jumper and Buck Converter: Powering the ESP32-S3 Module

One of the key components in the wireless remote motor controller is the ESP32-S3 module, responsible for wireless communication, control logic, and user interface. However, the ESP32-S3 module typically operates at 3.3V, which poses a challenge when powered by a 12V source. To bridge this voltage gap, the power system incorporates a 3.3V solder jumper and a buck converter. The 3.3V solder jumper plays a crucial role in voltage regulation. It enables the selection of the appropriate voltage level for the ESP32-S3 module, allowing for flexibility in the power system design. By connecting the solder jumper, the voltage is adjusted to match the module's requirements. The buck converter, an essential part of the power system, efficiently steps down the voltage from the 12V source to the required 3.3V level. This conversion process ensures that the ESP32-S3 module receives a stable and precisely regulated power supply. Buck converters are known for their efficiency, making them an excellent choice for conserving battery power while providing a clean and consistent voltage source.

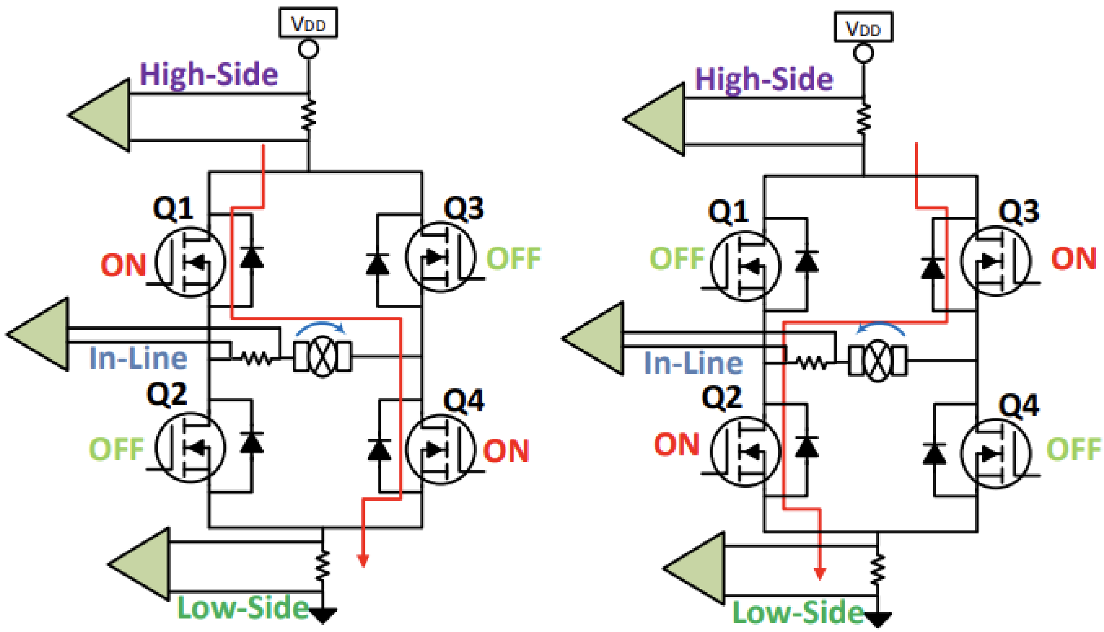
Battery Voltage Sensor and Schottky Diodes: Safeguarding the System

A critical aspect of the power system is monitoring the battery's voltage to prevent overvoltage or undervoltage conditions that could damage sensitive components. To accomplish this, the system incorporates a battery voltage sensor. This sensor continuously measures the battery's input voltage, providing real-time information about the power source's health. To safeguard the ESP32-S3 module and other electronics, two Schottky diodes are employed. These diodes are strategically placed to clamp the voltage at the output of the divider circuit. The purpose of this clamping is twofold: to protect the ESP32-S3 module's analog-to-digital converter (ADC) and to ensure that the voltage does not exceed 3.3V or drop below ground potential. The Schottky diodes are chosen for their low forward voltage drop and fast switching characteristics. This makes them effective in limiting the voltage and preventing any unwanted spikes or deviations that could adversely affect the ADC or other components.

In summary, the power system will start with a 12V DC battery input, which provides the primary power source for the system. A 3.3V solder jumper and a buck converter ensure that the ESP32-S3 module receives the correct voltage level for operation. To protect sensitive components, a battery voltage sensor and Schottky diodes are employed, ensuring the stability and safety of the power supply. This meticulous attention to the power system's design guarantees the reliability and longevity of the wireless remote motor controller, enabling it to excel in a wide range of applications while ensuring the safety of its users and components.

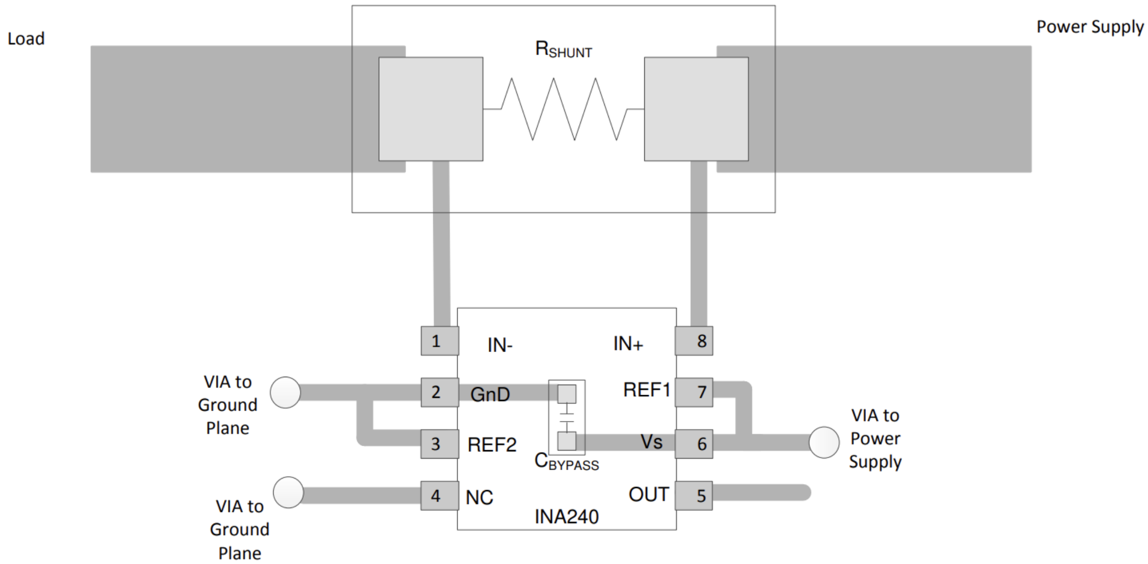
2.3.4 Current Sensing Subsystem

As mentioned before, the core of our motor controller circuit is an H-bridge. Because we are dealing with a maximum current rating of 10A, current sensing is used to monitor, manage, and control the load currents leading to improvement in safety, and reliability of our motor controller circuit. From the H-bridge circuit shown below, if Q1, and Q2 are both on or Q3, and Q4 are both on, it will cause a short circuit from battery to ground.

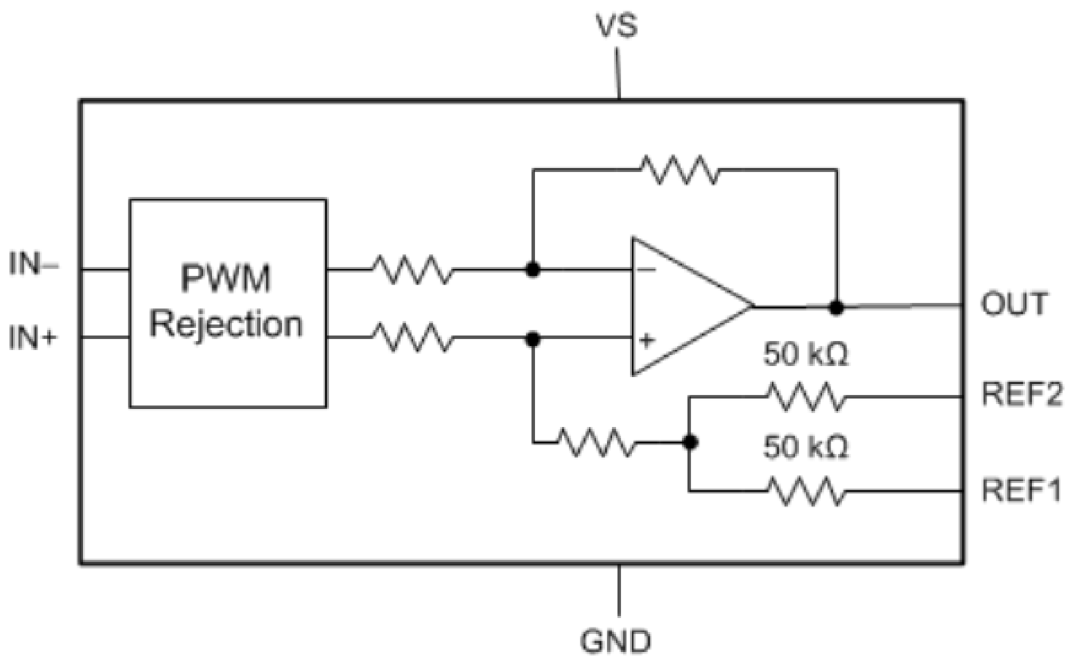


H-Bridge Circuit

In the figure, we can see that there are 3 different locations, High-Side, In-Line, and Low-Side, to measure current in an H-bridge. For our project, we use In-Line current Measurement for current sensing in an H-Bridge to direct motor current measurement and low-bandwidth amplifier. Accurate current measurement with an H-bridge is important to control motor torque. The PWM output often experiences overshoot and undershoot during transition from low to high and high to low transitions. Thus, it is important to have a current sense amplifier, which can endure these conditions while maintaining a fast response time and survive in harsh requirements of an inductive system. We are using an INA240 current sense amplifier with enhanced PWM rejection, which ranges from -4V to 80V. It is designed to reject or filter out unwanted signals or noise related to PWM (Pulse Width Modulation), which helps us make accurate measurements in systems that use PWM signals. It provides a high level of suppression for common-mode transients ($\Delta V/\Delta t$), which is important for real-time measurements of load current in in-line measurement positions. In other words, it can effectively filter out abrupt changes in voltage that occur in systems using PWM signals. Besides, INA240 is suitable for use in H-bridge as it can be used in various positions within the H-bridge: High-Side, In-Line, and Low-Side. The following figures show the functional block of the INA240 device.



This shows the functional block of the INA240 device.



Functional block of the INA240 device

2.4 Hardware Design

2.4.1 Operating Voltage & Regulation

In our project, we are designing a remote control system that utilizes an ESP32 microcontroller. To power these components, we plan to draw power from a 12 Volt battery or a 24 Volt source. To ensure seamless operation and compatibility with the ESP32's voltage requirements, we have decided to operate the microcontroller at a stable 3.3V with 3.3V logic levels. This voltage regulation is essential for reliable performance. Power to the system will primarily come from either the 12 Volt battery or the 24 Volt source, and we will incorporate an efficient voltage regulator to step down the input voltage to the 3.3V required by the ESP32. This voltage regulation is crucial to protect the microcontroller and other components from potential overvoltage issues. For convenient power supply and communication with the ESP32-S3 chip, we've included a Micro-USB port on the board. This port not only serves as a power source but also connects to the ESP32 via the on-board USB-to-UART bridge, enabling easy programming and debugging. Additionally, to provide visual feedback to users, we've integrated a 3.3V Power On LED that will illuminate when USB power is connected to the board. This LED indicator serves as a simple but effective way to signal the system's status to users or operators. In addition, we've taken steps to safeguard the ESP32-S3 module and other electronic elements by implementing two strategically placed Schottky diodes. These diodes serve a dual purpose: first, they protect the ESP32-S3 module's analog-to-digital converter (ADC), and second, they ensure that the voltage remains within the safe operating range, never exceeding 3.3V or dropping below ground potential. The choice of Schottky diodes is deliberate, as they offer low forward voltage drop and rapid switching characteristics, effectively limiting voltage and preventing unwanted spikes or deviations that could potentially harm the ADC or other critical components.

2.5 Software Design

The software is central to our project, as it controls the ESP32 microcontroller, allowing us to smoothly manage the motors with encoder. We've crafted the software using the Arduino IDE, making it easy to finely adjust motor speed and direction via wireless communication with a mobile device over Wi-Fi. We'll also consider the option of working with existing Bluetooth controller apps (such as Dabble - Bluetooth Controller) or creating a custom app based on our project progress.

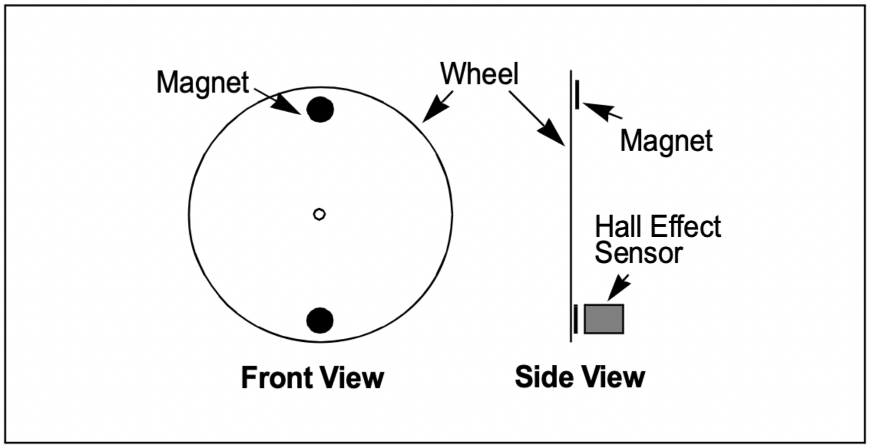
2.5.1 Motor Control

Our motor control algorithm is the core component of the software. It is responsible for interpreting user inputs and adjusting motor behavior accordingly. The algorithm takes into account a range of user commands and behaves as follows:

- Speed Control:
  - ACCELERATE: Command the electronic speed controller to increment the motor speed.
  - DECELERATE: Command the electronic speed controller to decrement the motor speed.
  - NO INPUT: If no input is received, gradually decrease the motor speed to zero, eventually coming to a complete stop.
- Direction Control:
  - STRAIGHT BACKWARD: Command the motors to move backward, rotating the front two wheels backward
  - STRAIGHT FORWARD: Command the motors to move forward, rotating the front two wheels forward
  - LEFT: Command to adjust the middle wheel (ball one) to the left, changing the motor's direction to the left.
  - RIGHT: Command to adjust the middle wheel (ball one) to the right, changing the motor's direction to the right.

2.5.2 Hall Effect Sensors

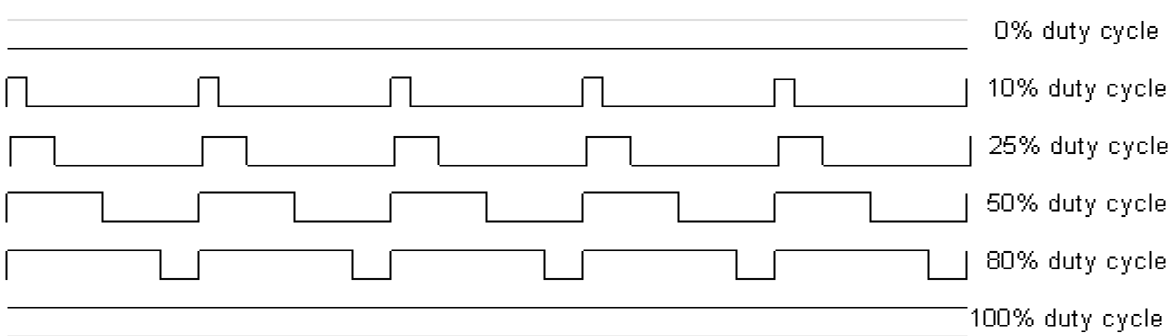
In our project, we use Hall Effect sensors from brushed DC gearmotors to provide accurate feedback on the motor's speed or RPM (Revolutions Per Minute). Placing a magnet on the rotating part of the motor and using a Hall Effect sensor nearby allows us to measure the strength of the magnetic field, which is proportional to the motor's speed. The following figure includes the fundamental components of the hall effect sensor. With these components, it determines the speed of rotating components (motors) at the correct sequence and timing. We are planning to use an encoder that provides 64 counts for every full revolution of the motor shaft. To determine the count per revolution at the gearbox output, we can simply multiply 64 by the gear ratio. We can calculate the speed using this algorithm in the Arduino IDE and display it on the screen to show the user the motor speed.



Hall Effect Sensor

2.5.3 PWM (Pulse Width Modulation)

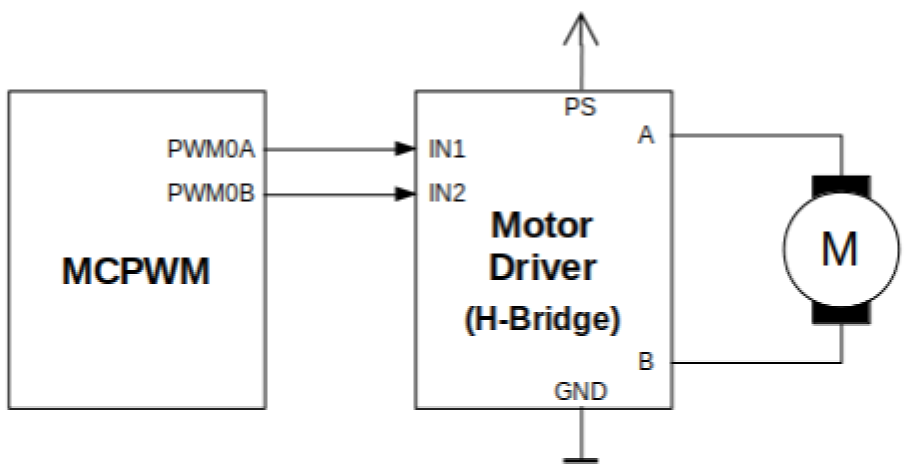
We use PWM to control the speed of motors in remote control systems. By adjusting the duty cycle of the PWM signal sent to the motor controller, we can control the average power delivered to the motor. A PWM signal is a square wave with a fixed frequency and a variable duty cycle. By expressing the duty cycle in percentage, we can represent the fraction of time during one cycle. The following figure represents a varying duty cycle. The duty cycle represents the percentage of time during each cycle that the PWM signal is in the "on" or high state. A higher duty cycle means the motor receives power for a greater portion of each cycle, resulting in higher speed. Conversely, a lower duty cycle results in lower speed. In other words, we can decide the time it takes to go from one rising edge to the next. This allows you to vary the motor speed efficiently. This algorithm can be implemented in the Arduino IDE, and the speed could be adjusted based on user input.



2.6 Tolerance Analysis

- **Voltage Regulator**
  - Because we are using the ESP32-S3-WROOM, we do not require a linear voltage regulator. Our input voltage will be ranging from 12 to 24V, and the motor controller can accept input voltage of this range. ESP32 microcontrollers usually draw a maximum current of 240mA. The dropout voltage would be 24-3.3=19.7V and P=V*I=19.7*240m=4.728W dissipated in the regulator. Therefore, we would need to use a buck converter to step down the high voltage for the 3.3V for the ESP32-S3 module. This provides a much greater power efficiency than the linear regulators.
- **Choosing MOSFETs for H-Bridge**
  - The majority charge carriers of p-channel MOSFETs are holes, it has a lower mobility compared to electrons. This has resulted in higher on-resistance of p-channel MOSFETS compared to n-channel MOSFETS. The formula for power is I^2R, to minimize the power dissipated in the H-bridge circuit, four N-MOSFETs are used at the low side and high side instead of two n-channel types at the low side and two p-channel types at the high side.
- **Capacity of Microprocessor**
  - The ESP32 microcontroller is a dual-core processor, Wi-Fi and Bluetooth connectivity, and a variety of I/O options. The figure below shows how MCPWM works with H-Bridge. Since we will achieve speed control using the PWM waveform and duty cycle, it is important to discuss the PWM frequency of the microprocessor. The user is allowed to specify PWM frequency from 1kHz to 40MHz and duty cycle resolution from 1 bit to 16 bits.Let's say with a theoretical frequency of 80MHz, the maximum PWM frequency for ESP32 can be calculated using the formula below: **Max_PWM_freq = 80,000,000/2^Duty-Cycle-Resolution-Bit**The following table lists some of the recommended PWM frequencies and duty cycle resolution for some common applications.

- The ESP32 microcontroller also has an inbuilt dead time generator that allows the user to specify the dead time on rising edge and falling edge. Besides, we can also generate signal pairs(PWMxA and PWMxB) to be active high, active low, active high complementary, and active high complementary. Furthermore, if we wish to integrate deadtime in the generator module, we are able to bypass the dead time generator module.



Example of Brushed DC Motor Control with MCPWM

PWM Frequency	Duty Cycle Resolution
10 KHz	13-bit
8 KHz	10-bit
5 KHz	14-bit
1 KHz	10-bit
1 KHz	13-bit
1 KHz	16-bit

2.7 Cost Analysis (shopping list)

Item	Cost
Current sensors	$3.23
13-V synchronous buck controller with fixed 3.3-V output	$9.81
2x Brushed DC Motor Gearmotor 10000 RPM Incremental 12VDC	$32.95
ESP32-S ESP32-S3-WROOM-1-N8R2 Transceiver; 802.11 b/g/n (Wi-Fi, WiFi, WLAN), Bluetooth® 5 2.4GHz Evaluation Board	$14.70
MOSFET N-CH 100V 9.7A TO220AB	$1.04
BATTERY PACK NIMH 12V AAA	$29.95
Total	$124.63 + tax

2.8 Schedule

Week	Task	Person
Oct. 2nd - Oct. 8th	Order parts for prototyping	Everyone
	Start PCB Design	Everyone
Oct. 9th - Oct. 15th	Continue PCB Design	Everyone
Oct. 16th - Oct. 22nd	Revisions to PCB Design	Everyone
	Begin Board Assembly	Everyone
Oct. 23rd - Oct. 29th	Individual Progress Reports	Individuals
	Continue Board Assembly	Everyone

	Begin Arduino IDE	Kyungha
Oct. 30th - Nov. 5th	Continue Arduino IDE	Kyungha
	Revisions to Board Assembly	Aaron, Boon
Nov. 6th - Nov. 12nd	Finalize remote System	Everyone
	Integrate Board Control system and Remote System and Test	Everyone
Nov. 13rd - Nov. 19th	Mock Demo	Everyone
Nov. 20th - Nov. 26th	Fall Break	-
Nov. 27th - Dec. 3rd	Final Demo	Everyone
Dec. 4th - Dec. 10th	Final Presentation and Paper	Everyone
	Lab Notebook	Individuals
	Lab Checkout	Everyone

2.9 Risk Analysis

Controlling a Wireless Remote Motor Controller demands a deep understanding of motor control mechanisms. One critical risk involved signal loss between the remote controller and the motor might lead to loss of control or unintended motor behavior. To mitigate this risk, we will employ robust wireless communication protocols and implement error-checking mechanisms to minimize the chance of signal disruption. Extensive testing across diverse environments will be conducted to ensure the reliability of the wireless connection. Another significant concern is the possibility of malfunctions or software bugs in the controller leading to unintended acceleration or deceleration of motors, posing a safety hazard. To prevent such incidents, we would design the algorithm to stop under emergency. Plus, rigorous testing, including simulated emergency scenarios, will be performed to ensure the controller's ability to swiftly and safely respond to anomalies. Additionally, the complexity of the mobile app's user interface poses a risk of user confusion, incorrect inputs, and accidental adjustments that could result in accidents or damage. To address this, we will prioritize a user-friendly and intuitive mobile app interface design, conduct usability testing with potential users, and provide clear instructions and safety guidelines within the app.

3. Ethics and Safety

In the development of the Wireless Remote Motor Controller project, we are committed to upholding the highest ethical and safety standards as outlined in the IEEE and ACM Code of Ethics. Specifically, we will prioritize safety by ensuring the device complies with ethical design and sustainable development practices. Safety is a paramount concern throughout the development and operation of the Wireless Remote Motor Controller project. It extends to various aspects, including power control, voltage regulation, soldering practices, and the proper use of equipment. Here, we emphasize the safety measures and considerations associated with these critical project components: To prevent overheating and protect the motor and other components, the power system must implement current limiting mechanisms. This ensures that the motor operates within safe limits, reducing the risk of damage or accidents. Maintaining a stable voltage supply, as achieved through the buck converter, is essential for the safety of the entire system. Fluctuations in voltage can lead to erratic motor behavior and pose risks to users and equipment. We will continuously monitor the voltage levels as a safety measure. The battery voltage sensor helps in this regard, allowing the system to take corrective actions if voltage levels fall outside safe operating limits. When working with electrical equipment, including the buck converter and voltage sensor, it is crucial to follow electrical safety practices, such as isolating power sources when making connections. Incorporating these safety measures and considerations into the Wireless Remote Motor Controller project not only ensures the safety of the development process but also contributes to the overall safety of the end product. Prioritizing safety at each stage of the project's lifecycle, from design and assembly to testing and operation, demonstrates a commitment to delivering a reliable and secure product. should have incorporated voltage input protection mechanisms to safeguard against voltage spikes, surges, or reverse polarity, reducing the risk of damage to the controller and connected motors. This requirement enhances the controller's durability and reliability. I, Aaron Chen, Kyungha Kim, and Lee Boon Sheng Adhere to this.

References

1. Microchip Technology Inc. "AN905 - Stepper Motor Control Using the PIC16F684." Microchip Technology Inc., http://ww1.microchip.com/downloads/en/appnotes/00905b.pdf. Accessed 27 September 2023.

2. Yavuz, Hasan. "" Ozderya, https://hasanyavuz.ozderya.net/?p=437. Accessed 27 September 2023.

3. Modular Circuits. "H-Bridges – the Basics" Modular Circuits, https://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/. Accessed 27 September 2023.

4. Michigan State University. "Application Note Regarding H Bridge Design and Operation " Michigan State University, https://www.egr.msu.edu/classes/ece480/capstone/fall14/group07/PDFs/Application%20Note%20Regarding%20H%20Bridge%20Design%20and%20Operation.pdf. Accessed 27 September 2023.

5. Texas Instruments. "*Current Sensing in an H-Bridge*" Texas Instruments, https://www.ti.com/lit/an/sboa174d/sboa174d.pdf?ts=1685998152220&ref_url=https%253A%252F%252Fwww.google.com%252F. Accessed 27 September 2023.

7.No specific author. "ESP32 Technical Reference Manual." Espressif, https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf#mcpwm.

8. No specific author. "Micropython on ESP8266 and ESP32 - PWM LED Fading." EngineersGarage, https://www.engineersgarage.com/micropython-esp8266-esp32-pwm-led-fading/#:~:text=While%20the%20base%20clock%20APB_CLK,cycle%20resolution%20of%201%20bit.

9. Jason🙂