# Raspberry Pi Car HAT Redesign

ESE 498 Capstone Design Project Formal Proposal
Submitted to Professor D. Wang, Professor J. Feher, and the
Electrical and System Engineering Department
February 11, 2025

## Client:

Electrical & System Engineering Department


## Advisors:

PhD. James Feher - jdfeher@wustl.edu
PhD. Dorothy Wang - dorothyw@wustl.edu

## Engineers:

Matthew Kubas - kubas@wustl.edu
Ethan Morton - e.a.morton@wustl.edu
Keeler Tardiff - k.a.tardiff@wustl.edu

Washington University in St. Louis
McKelvey School of Engineering

# 0 Abstract

Under the advisory of Dr. James Feher and Dr. Dorothy Wang, this capstone project redesigned the current Raspberry Pi "HATs" used in the *Introduction to Engineering Design* (ESE 205), to better suit the academic environment and make the parts open source. The new design solves the issues with current hats and integrates onboard sensors, which are currently incorporated as breakout boards on the *ESE 205* cars. The new "HAT" is designed using Altium Designer, a PCB design software, to create schematics, layouts, and 3D models of the design. Using OnShape, a browser-based collaborative CAD tool, a 3D-printable body was developed. These changes and upgrades make the new HAT and cars easy to construct, easily maintainable, and available to everyone.

# 1 Introduction

Modern engineering education emphasizes the seamless integration of hardware and software. This interdisciplinary approach is highlighted in many fields like robotics/automation and embedded systems. *Introduction to Engineering Design, ESE 205* is a fundamental introductory course for students to engage in diverse engineering concepts and practices through hands-on experience. Our project focused on improving the functionality of the PiCar HAT. The PiCar offers many features for use in a digital control system, such as serial communication options for communicating with sensors and a PWM module for controlling servos. However, the current implementation of the PiCar hat also has many drawbacks and hinders the educational experience.

The current PiCar HAT, manufactured and sold by Adeept, has some drawbacks that make it difficult to maintain and use. For use in the *ESE 205* class, the main issues are the use of dedicated SPI pins on the Pi with an ultrasonic sensor as GPIO, the lack of integrated onboard sensors and controllers such as

- an Inertial Measurement Unit (IMU)
- a Battery Monitoring System (BMS)
- an Analog-to-Digital-Converter (ADC)
- a PWM driver
- a motor driver

Finally, there are major problems with the looseness/slop of the joints and the overall weight of the body.

The model used by the ESE 205 class is the Adeept PiCar-B Mars Rover Smart Car Kit, seen at [1]. This Kit includes the Adeept Motor HAT for Raspberry Pi. The technical resources for these components are limited, making it difficult for instructors to adapt the device to fit their curriculum.

Our redesign provides the ESE department with a more well-rounded PiCar HAT that provides more learning avenues for student exploration. The redesign addresses current hardware limitations through a newly designed PCB layout, with sensors and controllers integrated directly on the board. In addition to the hardware implementation, the chassis has been redesigned to be stronger and lighter using 3D-printed components that can be easily assembled and disassembled.

Using this newly designed PiCar, students will find it easier to learn about the development and control topics introduced in *ESE 205* without needing to fight the hardware. Integrated sensors and controllers allow easy access to data for control algorithms and easy use of the attached motors and servos. A cost-effective and light chassis allows for easy replacement and repair of worn and broken components. Having all the project files open source also allows for an easy path for upgrading and modifying components as new needs develop.

# 2 Technical Approach

Before any technical research was conducted, we reflected on what we thought could be improved upon and contacted Professor Feher to gather his expertise on proposed improvements to the current system. From these discussions, we decided to focus on these issues:

- Correcting SPI pins
- Integrated ADC chip
- Integrated 9-Way IMU (Accelerometer, Gyroscope, Magnetometer)
- Integrated PWM Driver
- Hardware and software interface for battery monitor system (BMS)
- Integrated H-Drive PCB
- PiCar chassis Redesign

## 2.1 Correcting SPI pins

The current HAT sold by Adeept has reserved the Serial Protocol Interface (SPI) clock GPIO pin for use with the ultrasonic sensor. This requires the use of software-emulated SPI using other GPIO pins to communicate with the ADC, using deprecated SPI over GPIO libraries. With each update to Python and RaspiOS, using these deprecated libraries has become increasingly complex. The new design does not use the dedicated SPI pins from the Raspberry Pi, nullifying this issue.

## 2.2 Integrated ADC

The analog-to-digital (ADC) converter used for ESE 205 is on a breakout board with the MCP3008 chip, requiring extra wiring to incorporate into the system. The new design allows full access to all the ADC pins, referenced to 3.3V. This is included on the newly designed PiCar HAT, with reserved pins for the physical SPI interface on the Raspberry Pi. The schematic for the ADC can be seen in fig. 1.

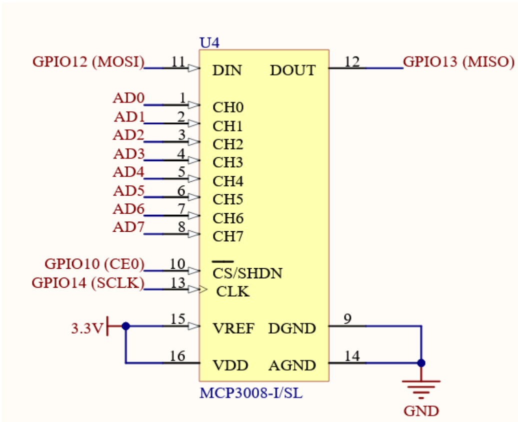



Figure 1: A simple schematic detailing how the ADC is connected on the new PCB

## 2.3 9-Way IMU

The current Inertial Measurement Unit (IMU) on the current PiCars consists of an accelerometer and gyroscope (6 DoF), recording the measurements of linear acceleration and angular velocity, respectively. This allows for basic inertial sensing, but there are limitations that impact the accuracy and reliability of the IMU's readings. The primary issue is the gyroscope's drift, referring to the gradual accumulation of error in orientation due to minor imperfections in sensor measurements. With no external reference, the gyroscope becomes increasingly inaccurate, leading to a cumulation of errors in the readings. The accelerometer can not distinguish between gravity and motion-induced acceleration. This increase in error requires frequent recalibration and adjustments to PID controllers to maintain stability. These issues led us to choose the more powerful 9-way IMU ICM-20948, which includes a magnetometer as well as the accelerometer and gyroscope.

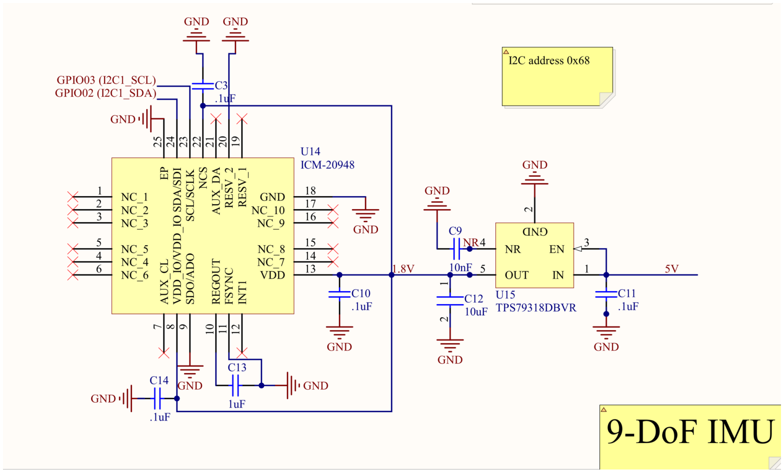



Figure 2: ICM-20948 Schematic with an LDO for proper voltage delivery to VDD and VDDIO

The addition of this sensor enables yaw correction, mitigating the gyroscopic drift and ensuring long-term orientation stability. This will be particularly useful when the battery power begins to drop since lower power levels can affect PID gains. Coupled with the battery monitoring system, students can adjust PID gains accordingly based on battery voltage levels. As Wescott(2000) [2] explains, a well tuned PID controller relies on precise and reliable sensor feedback to minimize error. Integrating the 9-DoF IMU, the PiCar will have more accurate control and consistency over long time periods, improving reliability, performance, and overall autonomous functionality. The current schematic IMU schematic can be seen in fig. 2.

## 2.4 Integrated PWM Driver

The current PiCar HAT for ESE 205 uses the PCA9685 to allow for 16 12-bit PWM channels. The current Senior Design PiCar HAT glues an Adafruit breakout board for the same chip to the HAT to obtain the same functionality.

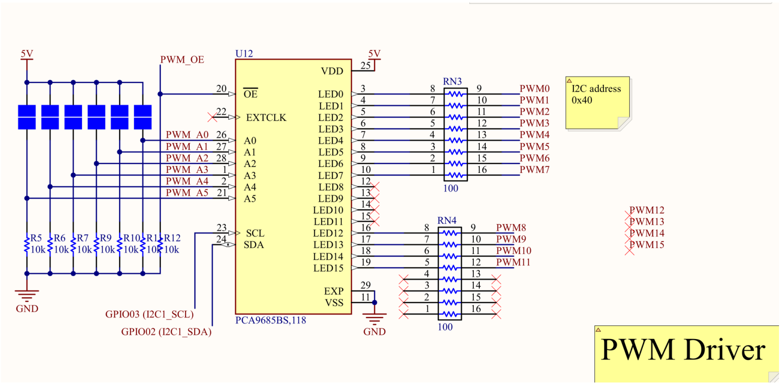
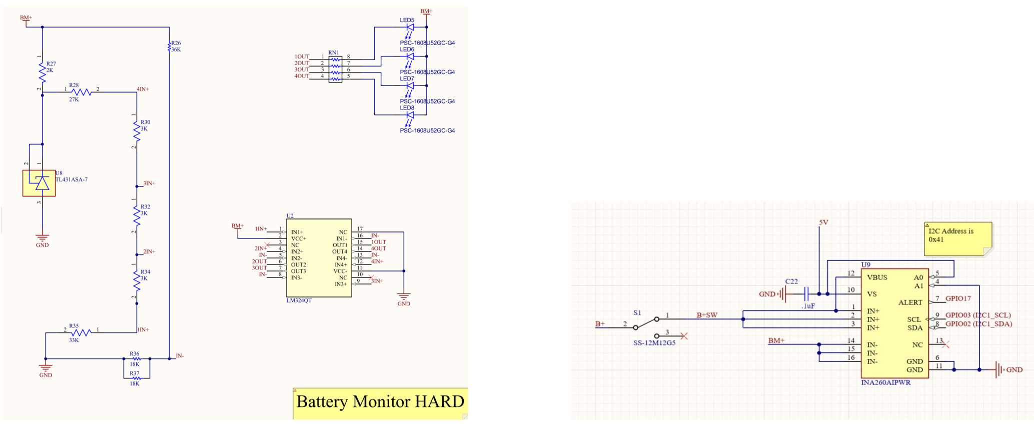



Figure 3: Schematic of the PCA9685 PWM driver.

To maintain functionality and keep space consumption to a minimum, we will follow in the footsteps of the former, while only exposing 8 of the 16 channels on the HAT PCB itself. The chip communicates through I2C (like many of the chips we have found/chosen) and can easily interface with the Pi through software libraries provided by Adafruit. We will expose the address pins of the chip on the board to allow for external address setting since the default address (0x40) is also used by other chips on the market. A schematic of the PWM chip can be seen in fig. 3

## 2.5 Hardware and software interface for Battery Monitoring System (BMS)




(a) Hardware Interface Schematic

(b) Software Interface Schematic

Figure 4: Battery Monitoring System working in parallel

The current battery monitoring system consists of four LEDs that represent different charge levels of the battery. This system makes it impossible to adjust control parameters from real-time data acquisition. By integrating a software interface in fig. 4b using the INA260, students will be able to monitor the exact voltage of the two-cell battery pack. This allows them to adjust PID control parameters and control algorithms on the fly, based on the battery power levels.

The hardware and software monitoring systems operate independently of each other. They function in parallel to provide a more comprehensive battery monitoring solution. With this enhanced system, real-time battery data can be used with the 9-DoF IMU to implement dynamic PID adjustments. This parallel integration provides greater adaptability, precision, and motor control consistency, improving the autonomy of the PiCar.

## 2.6 Onboard H-Drive

The current PiCar used in *ESE 205* uses the LN298 motor driver chip on the Adeept HAT to power the motors. Although this did not have any specific issues, newer chips are more space-efficient and power-efficient. We chose to implement a new motor driver chip, the DRV8411 motor driver chip from TI, seen in fig. 5.

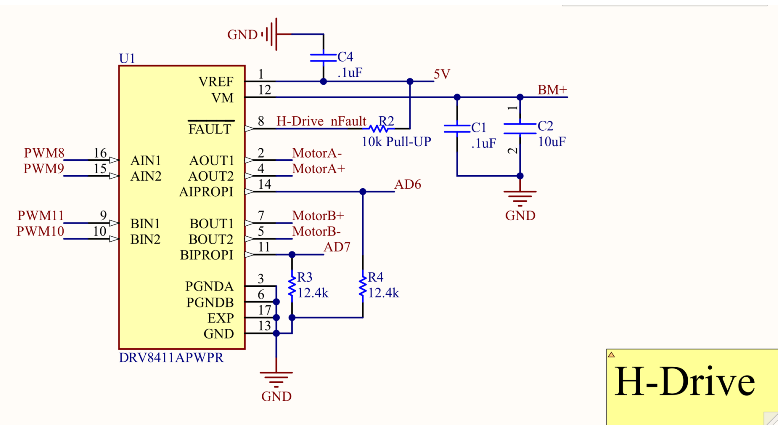



Figure 5: A schematic showing our implementation of the DRV8411 motor driver chip from TI.

This chip is rated for better power efficiency than the LN298, and is much smaller, making it easier to implement on a PCB. This chip uses two PWM signals to control a single motor, one signal controlling forward and one controlling back. It uses the differential between the two to power the motor. If both signals are on, then the motor is braking, if both are off, it is coasting, otherwise the motor goes forwards or backwards with the throttle determined by the PWM duty cycle (ranging from 0% to 100%).

## 2.7 5V Buck Converter

To power the Raspberry Pi and the other electronics on board, as well as provide power to the integrated motor driver and PWM channels, a 5V buck converter was developed. The current HAT implements a similar system, taking power from the batteries and using a switching regulator to step it down to 5V. This schematic can be seen in fig. 6.

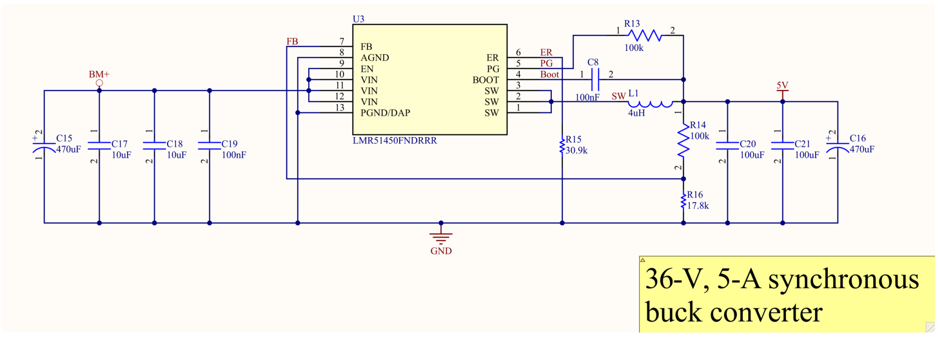



Figure 6: The buck converter schematic detailing the power system of the new PCB

## 2.8 PiCar Chassis

The Chassis, in its current form, functions but has some drawbacks. The current PiCar chassis by Adeept is difficult to maintain. Students tend to handle the car in ways that break certain components (usually the servos). These repairs take a substantial amount of time. This project has redesigned the chassis to be lighter, more solid, and with more easily replaceable parts.

Currently, the most difficult component to service is the swivel servo, requiring (on the new models) a complete deconstruction of the steering system to replace. This process takes roughly half an hour to complete (approximately ten minutes with experience), and with a class of 20-plus groups, each using a car, it becomes challenging to keep up with the number of broken servos.

The main issue as to the servos continuously breaking is the weight each servo has to endure. The head of the original car is constructed of approximately 15 parts and approximately 30 fasteners, which causes the head to be extremely heavy. This leads to the camera and ultrasonic sensor bouncing around while driving, which causes poor picture quality, inaccurate measurements, and high stress on the servo.

Since the car is constructed of laser cut acrylic, which allows for cheaper shipping and manufacturing costs, it leads to issues when designing a steering system. The current system relies on a laser cut steering arm that has round holes but due to limitation in the manufacturing style, the posts are square. This leads to a lot of steering play, which can ultimately cause the car to steer about $10^o$ to the left or right when steering perfectly straight. This is extremely frustrating for students testing their algorithm in an introductory class.

The new chassis, designed in OnShape, addresses these issues by constructing the head with only 6 parts and about 15 fasteners. This decreases the weight of the head to a manageable amount for the servos to handle. The design is compact to minimize the amount of material used and decrease the weight. The steering system is designed with a similar steering arm, but the shaft that fits into the round hole is cylindrical, dropping the play in the steering system to the tolerance of the fitment of parts and the play in the servo. This should drop the play in the steering system to less than $5^o$ total play.

The chassis is also designed that each servo can be accessed and replaced without needed to take multiple other parts out to get to. This means that if the swivel servo breaks, if can be replaced without needing to take the steering plate apart to access the screw needed to remove the steering servo. This is also true for all other moving parts in the system. If more parts need to be disassembled, the chassis as a whole is quite simple and will allow for easy maintenance of any part that needs replacing. A screen-capture of the new car design can be found in fig. 7. The car can also be viewed in OnShape at greater detail by visiting: `https://tinyurl.com/z7a4m2ur`.

# 3 Results

Towards the end of the semester, a PCB design was ordered to be tested, and the new chassis was printed and put together to allow for the comparison of the new system to the old. Overall, the new parts performed admirably. Other than a few small issues, and a major issue with the buck converter, all parts of the system work well.

Despite these issues, we were able to construct a fully functioning prototype, which was driven around via a Bluetooth controller during the poster session at WashU ESE Day Spring 2025. For some fun videos of the new

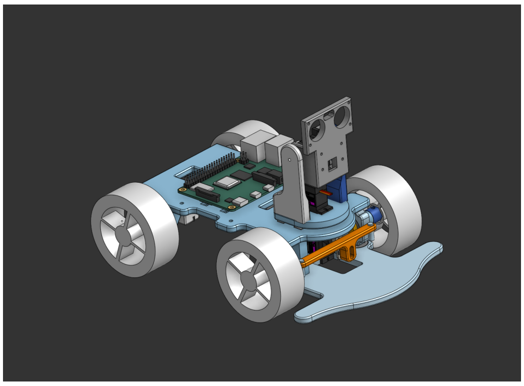


Figure 7: The CAD model of the new PiCar which is publicly available on OnShape.

PiCar actually driving around, click here.

## 3.1 PWM Driver

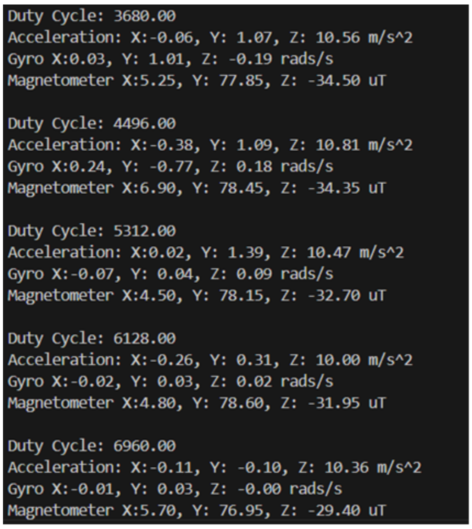
The PWM driver was tested using a cheap servo (found here) connected to the first PWM channel. An oscilloscope was used to look at the waveform of the PWM signal. Through some testing, it was discovered that the Adafruit python package takes a hexadecimal format input for it's 12-bit value, not an integer, which allowed us to drive the servo correctly. The PWM driver was also tested by controlling some external motor driver boards with the PWM signals, to ensure all the PWM channels worked correctly.

## 3.2 IMU

Figure 8, the data acquired from using the ICM-20948 on the PCB. The accelerometer, gyroscope, and magnetometer all work accordingly. No calibration was needed to perform these test.

## 3.3 5V Buck Converter

The buck converter is the major issue with this version of the board. The buck converter does not work with the current printed and ordered design of the PCB. Through testing with an oscilloscope and multimeter, it was determined that the buck converter is reaching only roughly 1V DC and has a ripple of multiple volts peak-peak. The exact issue is still unknown, but based on discussions with Dr. Feher, Dr. Wang, and Dr. Ivanovich, we have decided to replace the current chip, rated for 5A, with a different chip. The



```
Duty Cycle: 3680.00
Acceleration: X:-0.06, Y: 1.07, Z: 10.56 m/s^2
Gyro X:0.03, Y: 1.01, Z: -0.19 rads/s
Magnetometer X:5.25, Y: 77.85, Z: -34.50 uT

Duty Cycle: 4496.00
Acceleration: X:-0.38, Y: 1.09, Z: 10.81 m/s^2
Gyro X:0.24, Y: -0.77, Z: 0.18 rads/s
Magnetometer X:6.90, Y: 78.45, Z: -34.35 uT

Duty Cycle: 5312.00
Acceleration: X:0.02, Y: 1.39, Z: 10.47 m/s^2
Gyro X:-0.07, Y: 0.04, Z: 0.09 rads/s
Magnetometer X:4.50, Y: 78.15, Z: -32.70 uT

Duty Cycle: 6128.00
Acceleration: X:-0.26, Y: 0.31, Z: 10.00 m/s^2
Gyro X:-0.02, Y: 0.03, Z: 0.02 rads/s
Magnetometer X:4.80, Y: 78.60, Z: -31.95 uT

Duty Cycle: 6960.00
Acceleration: X:-0.11, Y: -0.10, Z: 10.36 m/s^2
Gyro X:-0.01, Y: 0.03, Z: -0.00 rads/s
Magnetometer X:5.70, Y: 76.95, Z: -29.40 uT
```

Figure 8: Data Acquired from ICM-20948

LM2596S-ADJ switching regulator is a slightly larger chip and rated for only 3A, but from our testing we are confident it is sufficient to power the current configuration.

In order to continue to test systems on the board despite the failing power system, we obtained a small breakout board for the new 3A regulator and connected it to the 5V pins on the GPIO pins of the HAT. This was able to power the full system, including the motors and servos. A photo of the small breakout board attached to the new chassis can be seen in fig. 9.

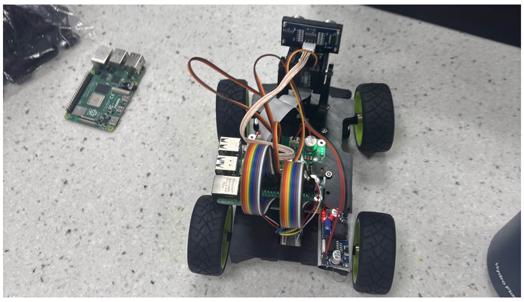


Figure 9: A photo of the make-shift solution enacted to test the system without a functioning buck converter.

```
Current: 117.50 mA Voltage: 8.05 V Power:1050.00 mW
Current: 132.50 mA Voltage: 8.05 V Power:1190.00 mW
Current: 152.50 mA Voltage: 8.02 V Power:1230.00 mW
Current: 187.50 mA Voltage: 8.03 V Power:1410.00 mW
Current: 323.75 mA Voltage: 7.93 V Power:2410.00 mW
Current: 342.50 mA Voltage: 7.90 V Power:2700.00 mW
Current: 330.00 mA Voltage: 7.92 V Power:2550.00 mW
Current: 356.25 mA Voltage: 7.90 V Power:2760.00 mW
Current: 132.50 mA Voltage: 8.04 V Power:1060.00 mW
Current: 123.75 mA Voltage: 8.03 V Power:1000.00 mW
Current: 136.25 mA Voltage: 8.02 V Power:1090.00 mW
Current: 128.75 mA Voltage: 8.05 V Power:1040.00 mW
Current: 128.75 mA Voltage: 8.04 V Power:1040.00 mW
```

Figure 10: Preliminary Data from INA260 breakout board

## 3.4 BMS

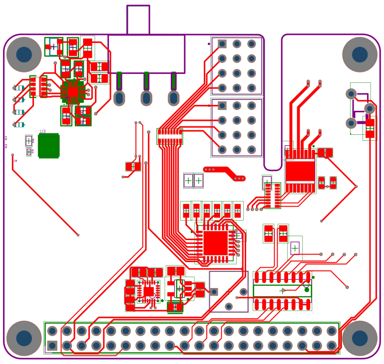
Figure 10, showcases the preliminary testing data using the INA260 breakout board from Adafruit. Unfortunately, the first prototype of a PCB is never fully correct. There is a minor error with the I2C communication pins of the INA260. The clock and data line for the I2C bus is flipped, and so it is impossible to communicate with the chip. However, due to the functionality of multiple other I2C chips on the I2C bus, we are confident that the simple fix of swapping the lines will fix the problem and enable precise voltage and current measurements from the battery.

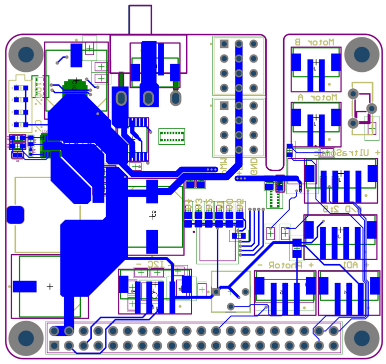
## 3.5 Motor Driver

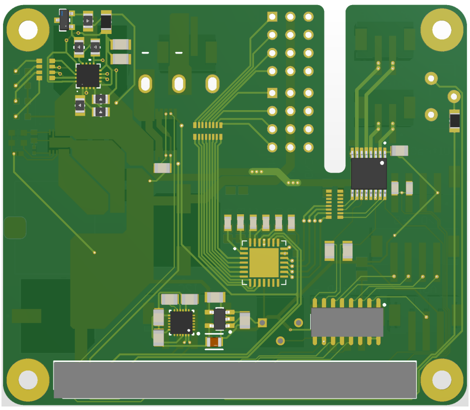
The motor driver chip performed exactly as expected. Once the system was powered from the external 3A buck converter, the motor driver was controllable via the PWM channels assigned to it and was able to power two motors continuously. By manipulating the PWM signals, braking, coasting, and forward and backward movement were achieved on both motors.

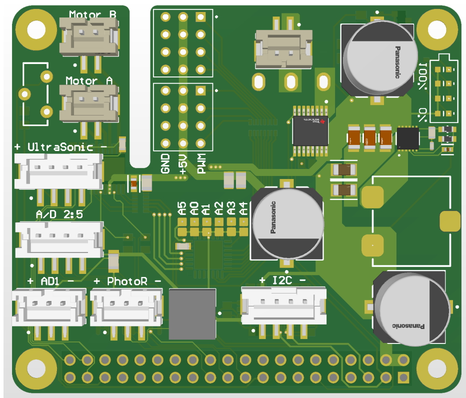
## 3.6 PCB

The PCB that was ordered functions well and has only a few small issues. The board is a 2-layer PCB, with signal on both sides. After the signal lines were completed, a copper pour was put over the entire rest of the board to act as large ground planes, stabilizing power and signal across the board. A key focus during development was the ease of use of the components. We placed multiple headers on the board with different purposes, all labeled appropriately. This can be seen in fig. 11b, which details the PCB layout. When viewing these figures, it is important to note that the *bottom layer* is the *top of the board*. So, headers on the bottom layer will be accessible from the top of the board, facing away from the Pi when mounted.


(a) Top Layer of prototype PCB


(b) Bottom Layer of prototype PCB


(c) Top Layer of prototype PCB



(d) Bottom Layer of prototype PCB

Figure 11: Prototype PCB top and bottom layers

The issues resulted from the quick timeline of design and order, and likely would not have been an issue with a bit more time to plan and develop. One issue is the lack of vias connecting the ground planes on either side. In our haste to order the board, we forgot to tie the ground planes on the top and bottom layer together evenly using vias, and as a result there is a slight voltage gradient across the board. This was easily fixed and will not be an issue for the next revision.

There was also an issue with the GPIO passthrough header. The header is intended to connect to all the Raspberry Pi's GPIO pins on the bottom using a female connector and present the same 2x20 array of male GPIO pins to be connected to. However, at some point in the design process, the 20-pin columns of the header were flipped. This means that what should've been the inside column was now the outside, and vice versa. This was fixed quickly on the schematic, but a make-shift solution was devised to continue testing where the HAT would be placed on the Pi offset by a row. So, the outside row on the HAT, which corresponds to the inside row of the Pi, was connected, and then some jumper wires were used to connect the inside row of the HAT to the outside row of the Pi. This enabled us to continue testing.

## 3.7 Chassis

The Chassis, for a first prototype, had a some tolerance issue, but performed great. One of the most significant issues with the original car was the large amount of head wobble. This was resolved by designing the new head with minimal hardware and being as compact as possible. It is extremely light and the servo has more than enough torque to hold the weight of the head without twitching. It also has a much wider base for added stability and rigidity. This will allow for more stable images when driving and more accurate distance measurements with the ultrasonic sensor. The steering was also significantly tighter, allowing only for an estimated $5^o$ of wobble, compared to approximately $15^o$ of wobble in the original. Since the new design has implemented a pair of motors instead of a single motor with exposed gears, it has eliminated a wear item from the car.

The car was also able to withstand some fierce driving. It was driven into multiple walls at full speed, and the chassis body was put to the bend test. The main chassis body was more than strong enough to withstand moderate bending stress, and if the part were to break, it could easily be replaced with just a dollar or two of filament from a 3D printer. There were also some errors in the first prototype, with a significant number of tolerance issue. This resulted in the amount of time required for assembly to quintuple. The sanding and filing required in this prototype was in the realm of 5-6 hours. Once all the parts were fixed the assembly only took about 30 minutes.

## 3.8 Full System Use

To truly test the system, an actual prototype was needed. This involved putting together the new chassis and wiring up components as needed. As referenced earlier in this section, the car successfully drove around for a while during the poster session at the WashU ESE Day Spring 2025. A photo of the completed prototype can be seen in fig. 12.


Figure 12: JJ, the PiCar Prototype

# 4 Discussion

Designing a complex PCB is an intricate task that requires close attention to detail to ensure functionality within its implementation. Small mistakes can lead to issues that must be worked around, which is what we performed. For example, the GPIO pin header on the prototype PCB was oriented incorrectly, with the odd and even pins being swapped in their positions. This mistake could have easily been mitigated through attention to detail.

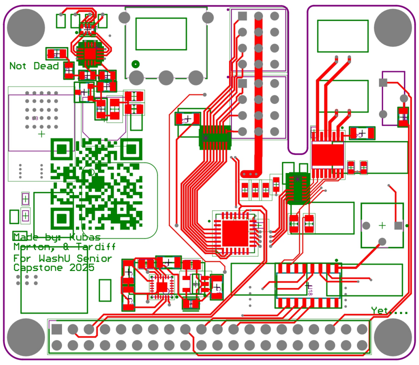
Not to mention, ensuring stable power to the PCB that in turn will power multiple motor types and the Raspberry Pi is essential. Unfortunately, our prototype PCB did not result in successful powering of the Raspberrry Pi off the PCB, so we bypassed it by using the LM2596S-ADJ buck converter breakout board, which can be found on Amazon and other retailers. The 3A buck converter successfully powered the PCB components and the Raspberry Pi. It's components and circuitry has thus been implemented in the newest version of our PCB and we await the completed PCB's arrival.

The diagnosis of why the 5A regulator on the prototype PCB does not work has not been determined. However, in fortunate circumstances a member of our team will be working closely with DC-DC converters over the summer during his internship. Hopefully, the two can diagnosis the true issue with the regulator and document it for future builds.

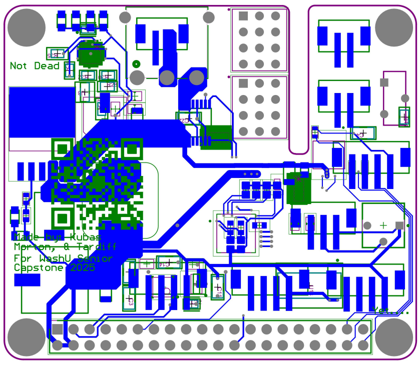
Our goal of being an open source project was met and will be updated in due time for users, personal and academic, to implement their own version of the PiCar as they see fit. The project files can be found at github/Raspberry-Pi-Car.

# 5 Future Work

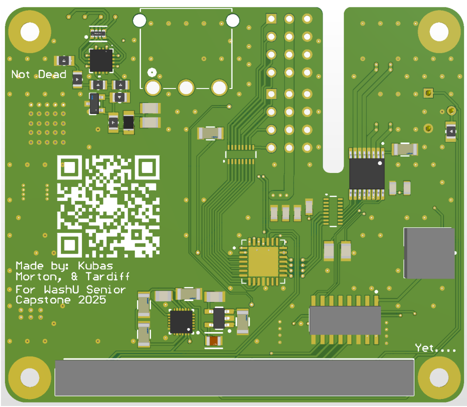
The LM2596S-ADJ 3A buck converter, having properly powered both the PCB and the Raspberry Pi, has been integrated with its supporting circuitry into the PCB HAT revision 1.1. As well other errors like the flipped GPIO pins and the INA260 I2C lines, have been corrected in this revision. Figure 13, showcases the top and bottom layers of the PCB. As well, revision 1.1 has a QR code directing anyone to the open source files for the Raspberry Pi Car project.



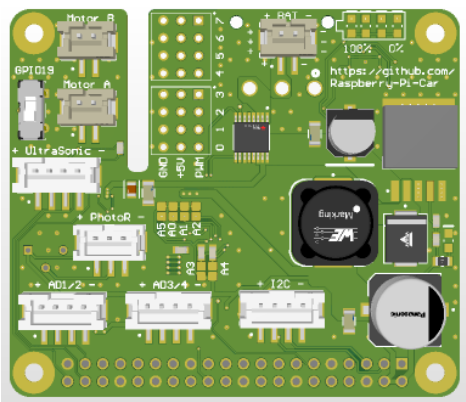

(a) Top Layer of PCB revision 1.1




(b) Bottom Layer of PCB revision 1.1




(c) Top Layer of PCB revision 1.1




(d) Bottom Layer of PCB revision 1.1

Figure 13: PCB Revision 1.1

# 6 Conclusions

Over the course of this project, we have learned a lot about PCB design, project structure, and planning. The experience of designing and ordering a full PCB, complete with multiple ICs, a power system, and multiple communication buses, as well as designing a physical chassis to house that PCB, is invaluable. The prototype produced worked well and is a promising replacement for the current PiCars in use for *ESE 205* despite a few issues. The end-to-end development cycle reinforced in this project is great experience for developing real products, and the skills learned during development will help to guide future growth.

The open-source nature of our project also lends itself to a wider academic and hobbyist audience. By producing documentation describing the design and use of the system, many people, student or otherwise, will have access to an easy-to-build starter project that they can analyze and improve upon themselves.

# 7 Anticipated Results and Deliverables

This project had three main deliverables:

1. A HAT PCB for the Raspberry Pi that supports relevant functions to act as a controller for a small "rover" and is compatible with previous iterations

2. A revamp of the PiCar codebase used to support the *ESE 205* Class at Washington University in St. Louis

3. 3D printable chassis (stretch goal) allowing for easy customization and easy repairability.

4. Relevant documentation and support to allow for easy integration and use, not only in the WashU class but also for the broader public.

Due to unforeseen issues with the power system, we were unable to deliver a completely working HAT before the end of the semester. A revision has already been completed and a new HAT will be ordered in the near future to test a different buck converter, which is the one we implemented to bypass the non working one on the prototype HAT. If this HAT functions as expected, the ESE department will be able to order enough to replace the current HATs if desired.

Under the advise of Prof. Feher we did not revamp the PiCar codebase to focus on possibly creating a functioning and reliable HAT to replace the current ones used in *ESE 205*. Since the goal is to make this project completely open-source, the codebase will we revamped in the near future and made available on GitHub.

To ensure the delivery of a product to Prof. Feher and the ESE department, the chassis and all components to compliment the chassis were redesigned in a manner that allowed them to be 3D printable and easily repairable. The current version is approximately 95% optimized and almost all issue with initial prototypes fixed. The next steps forward will be to print one and test the ease of assembly, the goal is to make it easy and intuitive to assemble with a minimization of screws and other hardware used. More work will be done to decrease the number of screws needed in the future.

Relevant documentation for the pinout of the HAT and relevant information on how to communicate with all components on the HAT will be detailed on the GitHub repository relevant to the particular deliverable. The repositories will be found at github.com/Raspberry-Pi-Car, where there will be separate repositories detailing the HAT, Chassis, and a separate repository for the PiCar codebase, allowing for easy download and use as a package. This will allow for easy referencing and easier future iterations of each individual part separately without interfering with each individual piece. Due to time constraints and multiple project coming to a close at once, these will be completed in the coming days and will be found on the GitHub page.

# 8 Timeline

The Gantt Chart in fig. 14 outlines the project timeline and major milestones.

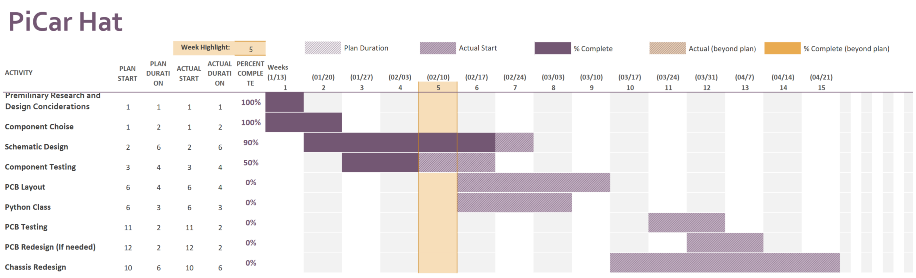



Figure 14: Gantt Chart for project completion

Due to limitations in our PCB design experience and a malfunctioning power system the initially proposed Gantt chart was not adhered to strictly. The PiCar class was temporarily postponed, under the advise of Prof. Feher, and will be continued over the summer. There was more unscheduled testing done at the beginning of the semester for the 5V system that postponed the first prototype HAT from being printed and tested. The HAT and prototype PCBs were continuously worked on throughout the semester and had taken significant time to redesign, test, and find a functioning buck converter for the 5V system. This led to significant delays to our timeline. The chassis was designed and test prints done in the last 3 weeks of the semester.

# References


[1] Adeept, "Adeept picar-b mars rover smart car kit for raspberry pi 5/4b/3b/3b+, obstacle avoidance, line tracking, light tracing, camera, speech recognition," Jan 2019.

[2] T. Wescott, "Pid without a phd," 2018.