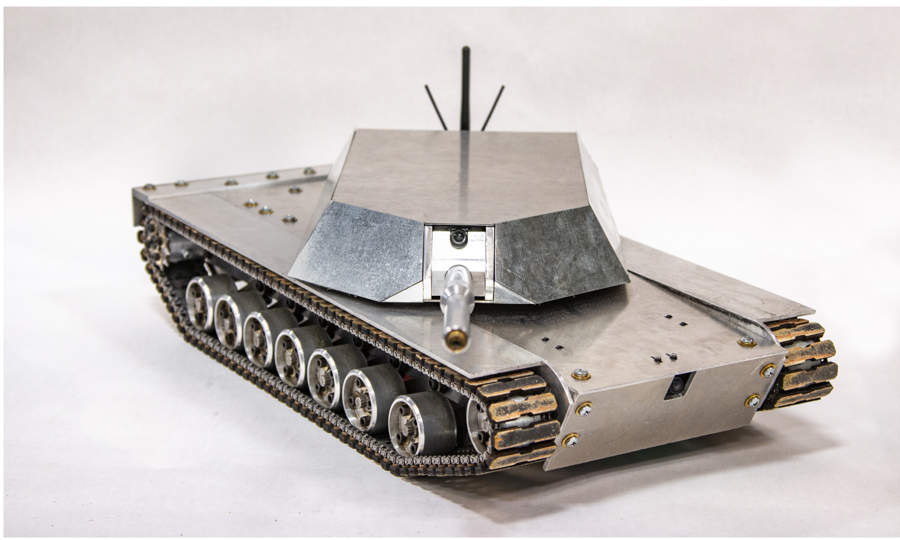
CS2951K Final Report

Gabe Weedon

May 2019



1 Abstract

The goal of this project was to upgrade an existing remotely operated tank platform with mechanical improvements, a new electronics/software package, significantly improved user controls, and a live video telemetry system. This provided an opportunity to practice hardware and software design as well as learn about remotely operated vehicles and human/robot interfacing. The first version of the tank was built in 2018 for ENGN 1931I Design of Robotic Systems and was modeled after the American M1 Abrams main battle tank. The project was ambitious and quite successful but inevitably a prototype. Among other issues, the laptop base-station used for both user input and telemetry was unergonomic and made precise control of the robot difficult. The new user interface utilizes a 16 channel RC radio transmitter and a pair of first-person view goggles. Together these components provide all the functionality of a laptop with significant improvements to functionality, precision, and ergonomics.

2 Introduction

Replacing humans with robots can provide massive advantages in many military, industrial, and research applications. These systems are extremely versatile, can explore places that a human could not, and can offer massive safety advantages as they move humans out of dangerous environments. While much research has gone into developing autonomous robots, remotely operated vehicles (ROVs) are still often the best tool for many jobs where human intelligence is needed. One of the challenges in designing an ROV is developing its user interface. There is no limit to the complexity of the robot, but coordinating many arms, actuators, cameras, wheels, etc. can quickly become impossible for a single person. Similarly, the user needs to understand the state of the robot in order to properly operate it. This type of feedback can include everything from camera feeds and location information to internal parameters like battery life. Developing a human/robot interface without sacrificing the flexibility and power of the vehicle or overwhelming the operator can take a lot of careful design.

As mentioned previously, the goal of this project was to build upon previous work, with an emphasis on improving the experience of the user. The 2018 version used a laptop base-station for both user input and telemetry. Though this strategy was relatively easy to implement it had several major problems. First, it made precise control of the tank's speed and turning rate impossible, as a keyboard only provides binary signals. It was also very difficult to operate if the laptop was not placed on a table or other surface, which severely limited the systems usability in the field.

This project replaced the laptop with an entirely new system. User input is now provided by a 16 channel RC transmitter, specifically the FrSky Taranis X9D. This remote is both significantly more ergonomic than a laptop and much more suited to this application. It has two analog joysticks each with two axis of control, as well as numerous configurable switches. The laptop's screen has been replaced with a pair of FatShark Attitude V4 first-person view (FPV) goggles. These display a live video feed from the tank with a custom on-screen display (OSD) overlaid to provide all necessary telemetry information.

Figure 1: FrSky Taranis X9D Transmitter and FatShark Attitude V4 Goggles

In addition to the changes already mentioned, several other improvements were made. On the mechanical side, improvements to the front of the chassis were implemented to increase durability in a front-on collision and an entirely new aluminum turret was designed to replace the previous 3D printed version. The electronics system was also entirely overhauled to improve safety, reliability, and performance. The entire system is now soldered to a PCB, rather than mounted on a breadboard, and the single Arduino Uno used for communication and control was replaced with three Arduino Nanos: one for communicating and parsing data from the RC transmitter, one for managing the OSD and telemetry, and one to control all of the vehicle's moving parts. The latter change actually simplified the overall code requirements for the project, as each individual Arduino has a relatively simple and well-defined task. This updated electronics package is also now protected by a new acrylic enclosure that keeps wires away from the moving components of the vehicle and prevents dirt and moisture from causing short circuits or other problems.

Overall the project was successful in meeting its goals. Though the system is still not flawless, many of the biggest issues with the first prototype have been eliminated. The rigidity and durability issues of the chassis and turret have been addressed. The new electronics and software package has been implemented and tested and meets all functionality requirements. Finally, the new user control system is finished and works extremely well. After a test run across campus all systems have been verified and are working as expected.

3 Previous Work

A lot of time and thought was put into the 2018 version of the tank. The first step was designing the mechanical assembly. This meant building moving models in Solidworks of all systems including the drivetrain, suspension, treads, and turret. The tread design used two loops of roller chain per side connected by pieces of laser-cut plywood. This solution was inexpensive and gave excellent traction without compromising turning ability. Two independent drive motors were used, one per side, allowing for differential turning. Each motor was linked to its drive sprocket with a 3:1 gear reduction to increase torque, and the drive sprockets in turn interfaced with the bike chain to spin the tracks. This system required a lot of careful engineering and tight manufacturing tolerances but generally worked very well in the end. Each side also had 8 non-powered wheels, 7 of which supported the weight of the tank with the help of a spring-damper suspension system. This gave the tank excellent maneuverability over all types of terrain from sidewalks to grass and dirt.

The motors themselves were large 380KV brushless motors, each capable of drawing up to 90 amps. Power was supplied by a pair of 100 amp electronic speed controllers (ESCs) and an array of four 1300mAh lithium polymer batteries in parallel, giving a total capacity of 5.2 amp hours at approximately 16 volts and a peak current reservoir of over 300 amps. In theory, this gave the tank a top speed of 15mph.

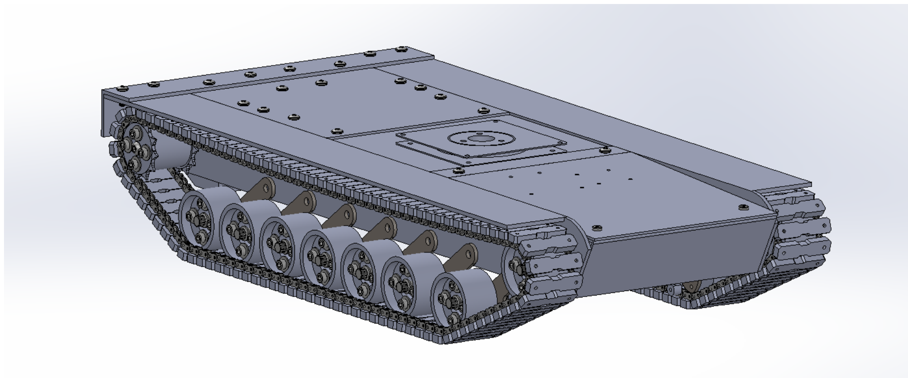


Figure 2: A top view of the 2018 model showing the chassis construction

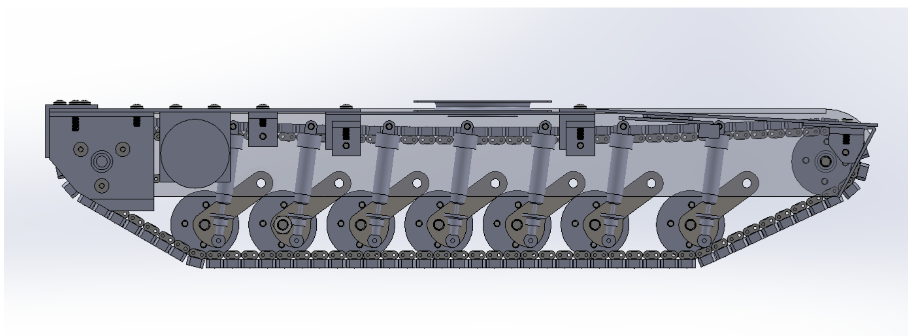


Figure 3: A side view of the 2018 model showing the suspension system

In addition to the two drive motors, the system used a stepper motor to spin the turret, which was placed on top of a turntable bearing. A pair of gears and a stepper belt were used to connect the shaft of the stepper motor to the rotating turret, and a 12 wire slip ring was used to allow electrical signals to be communicated between the chassis and the turret while still allowing for infinite rotation in either direction. The turret itself was 3D printed in 6 large pieces and glued together.

A single Arduino Uno located in the chassis controlled the entire system. It parsed commands from the base station and used them to set the speeds of the two drive motors and the rotation of the turret. It also used its analog inputs to read the battery voltage and current draw of the entire system. These statistics, as well as other diagnostic information, were returned to the base station.

Wireless communication between the Arduino and the base-station was achieved using two XBees to create a 2.4GHz two-way wireless serial connection. One XBee was connected to the laptop using a USB to serial converter and the second was placed in the turret of the tank for optimal signal reception.

A pair of cameras, one located in the front of the body of the tank and another located on the turret, sent analog video back to receivers on the base-station using two 5.8GHz links. Though the quality of the video was not incredible, the system was very low latency and relatively long range, which allowed the user to operate the vehicle without line of sight.

A custom user interface was built to run on the base station. This caught user input, communicated with the Arduino, and displayed both the live videos and telemetry information.

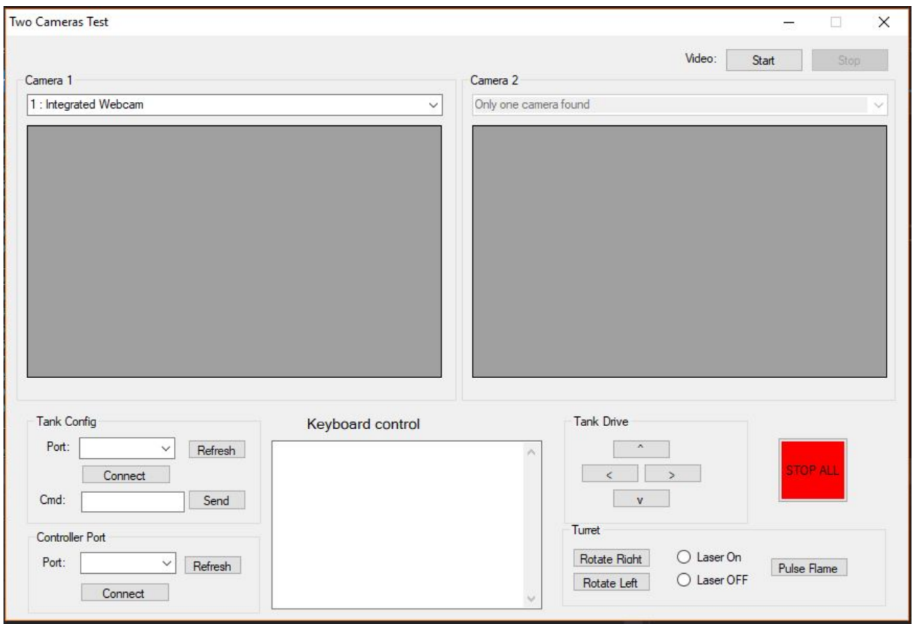


Figure 4: A screenshot of the original user interface

Despite a lot of careful design the 2018 prototype had several issues. The issues with the base-station and user interface have already been discussed. Beyond that, some glitches with the Arduino code lead to occasional system lock-ups. These could be extremely dangerous as they could lead to the loss of control of the very heavy and very powerful robot, so precautionary measures and failsafes had to be implemented. The entire electronics system was mounted on several breadboards. Though these made rapid prototyping much easier they were unreliable and caused issues that were often very hard to debug. The lack of protection around the electronics also lead to problems. With batteries able to discharge literally hundreds of amps, a safer electronics package was clearly needed.

The mechanical components of the system were also imperfect. The 3D printed turret was a weak point of the design, and the turntable bearing it was mounted on had a considerable amount of slop when torqued which lead to imprecise movements. Though the drivetrain was generally well thought out, the drive motors were able to apply so much torque that they could actually spin the drive shaft inside the drive sprocket despite 4 set screws. Finally, a front-on collision with a sidewalk bent part of the chassis and left one of the front wheels severely misaligned. Since this kind of a collision is relatively likely, reinforcing these components was necessary.

4 Implementation and Improvements

With a set of problems and improvements in mind, the first steps in the project were purchasing new components, working through how everything would fit together, and updating the Solidworks model.

In order to interface with the RC transmitter a radio receiver was needed. A FrSky XM+ was chosen for this task. The module is small and lightweight, has a diversity antenna array for increased signal integrity, supports up to 16 channels, and outputs data on a single line using the SBUS serial protocol. This protocol is very common in the hobby community and Arduino libraries for parsing the signals are available, which made coding much easier.

The other important part of the new user interface was the OSD. Since the FPV goggles have no onboard processing, unlike the laptop, the telemetry data has to be overlaid directly onto the analog video signal before it is transmitted from the tank. Fortunately, OSD chips are available for exactly this task. This project used a MAX7456 OSD chip on a breakout board from HobbyTronics. The chip communicates over SPI making it compatible with an Arduino and the vendor even had example code available.

In addition to the OSD chip, the new video system needed a way to toggle between the tank's two cameras, since the FPV goggles can only receive one video stream at a time, unlike the laptop. Fortunately this is not an unusual problem, and an inexpensive analog video switching module was easy to find. The module simply takes in two video signals and uses a logic input to select which one gets sent to the output, with zero latency and fast glitch-free switching.

Other new electronic components included a pair of new FPV cameras, an upgraded video transmitter with a higher output power for increased range, a 9V and 5V step-down converter (the batteries output 14.5-16.8V depending on their charge level), and a pair of current sensors, one capable of reading up to 25 amps and one capable of reading up to 100 amps.

Finally, pair of massive 8Ah batteries were purchased. Each battery is rated at 96 amps, so together they can safely provide 194 amps, which is plenty for the project. These increased the battery capacity of the vehicle from 5.2Ah to 16Ah for longer run times. They also fit much more comfortably in the front bay of the tank, which made packaging much simpler and more secure.

Updating the Solidworks model took several weeks, as it meant meticulously thinking through every design change and any unforeseen problems it could cause.

The largest change to the chassis was with the front bay, which was modified to accommodate the new batteries and to increase strength in a front-on collision. Each front corner, where the top, front, and side plates met, and where the front wheels are mounted, was reinforced with a large block of steel. The front plate itself was made much thicker, and a cutout was designed to allow the camera to be mounted inside the chassis out of harms way without an obstructed view.

The rear of the tank was also modified with added plates to keep dirt and gravel out of the drive chain.

The single turntable bearing for the turret was replaced with a pair of bearings on opposite sides of the top plate. These two bearings were connected with bolts that compressed the stack enough to remove any slop without severely increasing rolling resistance. This system worked well and meant the turret was much more secure.

The new acrylic enclosure was designed to be a tight fit to the underside of the chassis for maximum dirt and moisture protection. Two side pieces extended the length of the chassis and a bottom plate completed the box. In addition to being protective, the side pieces provided mounting points for another acrylic plate on which all of the chassis electronics could be mounted. This plate was necessary as the larger batteries in the front and additional central bearing limited other mounting options.

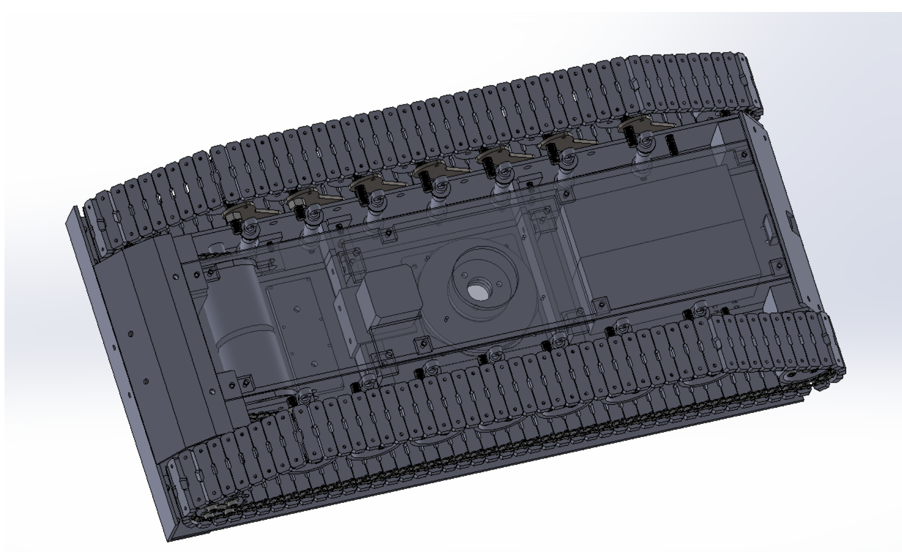


Figure 5: A bottom view up the updated chassis showing several design changes

While the changes to the chassis were relatively minor, the turret was entirely new. The previous version was 3D printed, which was fast to manufacture but not very strong or precise. The new turret uses carefully machined aluminum pieces for all load bearing and moving components. The gun is just over a foot long and machined from aluminum rod. It is mounted in an aluminum block that can rotate in another aluminum piece to elevate or depress the gun, as a real tank can. A high torque servo motor controls the motion of the entire assembly. The camera is mounted on the moving portion just above the gun, so the tip of the gun remains stationary in frame as the turret is rotated or elevated.

To enclose the turret, a non-critical 3D printed structure was made to mimic the relatively complex geometry of a real M1 Abrams turret. To keep with the all aluminum aesthetic, thin sheets of aluminum were cut and glued on top of the 3D prints. This solution ended up working very well, and the final product looked great.

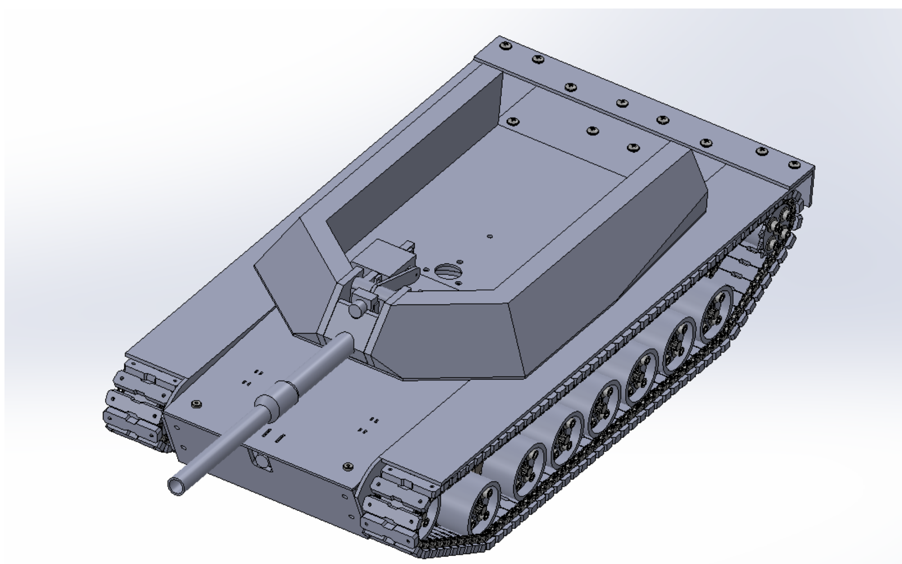


Figure 6: A bottom view up the updated chassis showing several design changes

With everything designed the next step was manufacturing. This took another few weeks in the machine shop but went smoothly.

In addition to the modeled changes, the drive shaft was modified with several flat spots to give the set screws holding it inside the drive sprocket significantly more holding torque. These fixed the slip issues that the previous version of the system was experiencing.

Changes to the chassis were completed first so that assembly of electronics inside the chassis could begin while the turret was being made. Packaging all of the necessary electronics inside a very tight space ended up being quite an interesting challenge. Since the goal was to have everything be soldered, extremely careful planning was required before assembly could begin. One major consideration was the limited number of wires available in the slip ring between the chassis and the turret. Both power and data for many different components had to be transferred over just 12 lines. Another consideration was the number of pins on the Arduino. The following simple schematic was developed to help organize everything:

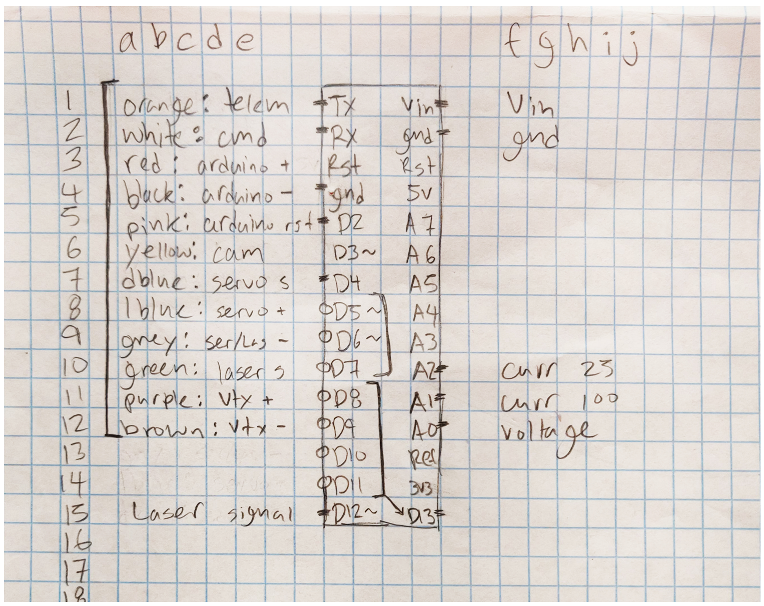


Figure 7: A simple schematic of the chassis electronic layout

All 12 slip ring wires are used: two for serial communication, two for power to the Arduinos in the turret, one to reset the receiver Arduino if communication is lost, one for the body camera's video signal, one PPM signal line for the turret servo, power for the turret servo, toggleable power for the laser in the gun, and power for the video transmitter and camera in the turret. Though fewer wires certainly could have been used, the this arrangement was adequate and nothing was left out.

Most of the Arduino's digital pins were used: D2 resets the receiver Arduino, D4 generates the PPM signal for the servo, D5-D7 are used to control the drive motors, D8-D11 and D13 control the stepper motor driver, and D12 controls the laser. For analog inputs, A0 measures battery voltage with the help of a voltage divider, and A1 and A2 measure current from the two current sensors.

This schematic does not show several other circuits that needed to be built. A PMOS transistor was used to toggle the laser, since the power requirements were more than an Arduino pin could deliver. Additionally, a NPN and a PNP transistor were used to activate and deactivate the drive motor ESCs. The

ESCs were desired to be activated with a manual toggle switch, but with the transistors digital logic from the Arduino could replace manual switching. This was actually a very important safety feature. Because of the way the circuit was set up, if the Arduino lost power the motors would deactivate automatically.

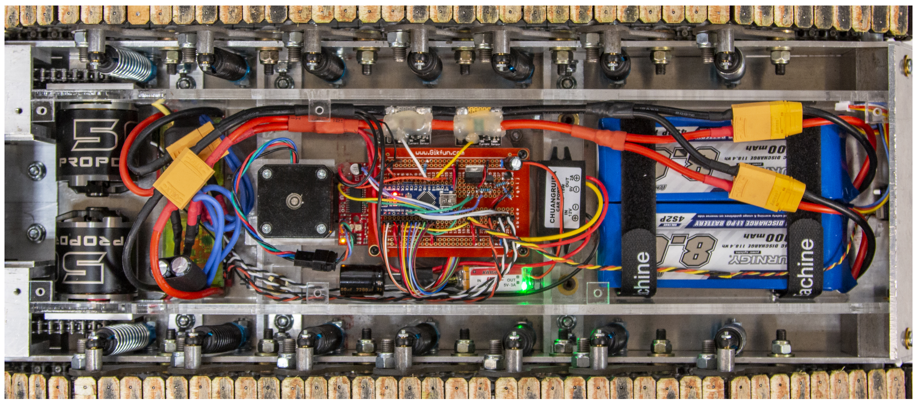


Figure 8: The finished chassis electronic layout

Figure 8 above shows the completed chassis electronic layout. As mentioned previously, packaging was extremely tight, but in the end everything fit. At the very front of the chassis (right) we have the camera with its bundle of red, black, and yellow wires running back to the main PCB. Just behind the camera are the two massive batteries with equally massive connectors. The two batteries are joined and go through both current sensors above the main PCB, after which power is split to be sent to the rest of the system. The larger wires lead back to the ESCs, mounted to the right of the two drive motors. The 9V stepdown regulator is mounted left of the batteries while the 5V stepdown regulator is below to the main PCB. A massive capacitor helps clean up the power going to the regulators, as brushless motors can create massive voltage spikes thanks to their power and active braking abilities, while additional smaller capacitors clean up the power coming out of the regulators. The main PCB itself is in the center with the Arduino mounted alongside other circuits. The two black and white twisted pairs leading back to the ESCs are signal lines and the black and red pair are the activation toggle. Left of the PCB is the stepper motor driver, and left of that is the stepper motor itself. The large bundle of multicolored wires below the Arduino are the slip ring wires, they run under the PCB through the bearings up to the turret.

Many different connectors were used to allow the main acrylic board on which all the electronics are mounted to be removed. This was incredibly useful and made soldering far easier, though it was still quite tricky at times.

Wherever battery voltage was exposed, such as at the solder pads of the current sensors, precautions such as covering the exposed contacts in hot glue had to be taken to prevent shorts. With almost 200 amps of current available even a minor short can instantaneously vaporize small wires.

Compared to the chassis electronic layout, managing everything in the turret was easy. Though there were still a lot of components to worry about, there was significantly more space to route everything comfortably.

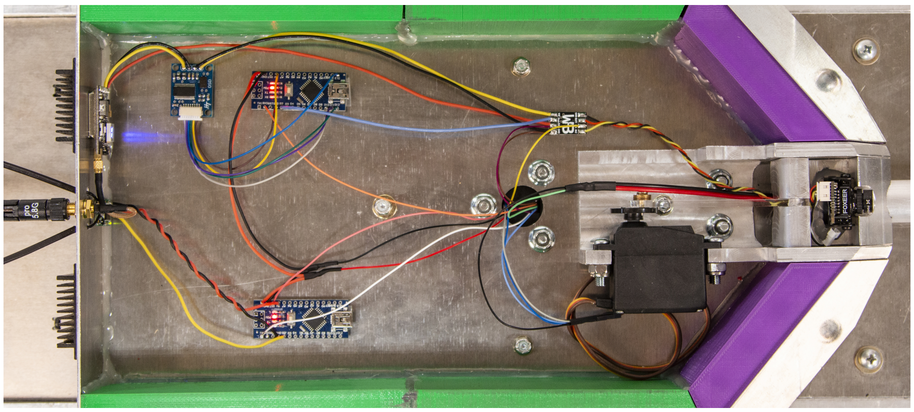


Figure 9: The finished chassis electronic layout

Figure 9 above shows the final layout. The bottom Arduino talks to the receiver module mounted center-left. The red and black twisted pair is power and ground for the module while the yellow line is the SBUS serial signal. These signals are parsed and passed along to the chassis Arduino on the white slip ring wire. The top Arduino gets serial data from the chassis Arduino on the orange slip ring wire and talks to the OSD chip mounted just left of it over a 4-wire SPI interface. At the front (right) is the camera with power and signal lines running back to the tiny video switcher module. The blue wire from the OSD Arduino selects which input gets sent back to the OSD module, either the signal from the turret camera or from the body camera on the yellow slip ring wire. To the bottom-right is the stepper motor that controls gun elevation which is powered directly from slip ring wires. Finally power to the gun laser is delivered over the green slip ring wire.

With all of the hardware and electronics done all that was left was programming the three Adruinos. Fortunately this process was relatively straightforward with the help of libraries and example code for talking to the receiver and OSD modules.

The receiver code was the simplest. Communicating with the module is done asynchronously. Every 10 milliseconds the latest information is parsed and sent over serial in 7 bytes. The first byte is a simple start byte, 255 or B11111111. The next 4 bytes represent the positions of the two joysticks, and the 6th byte represents the signal strength the receiver is detecting (RSSI). These all have a maximum value of 254 so they can never be confused for a start byte. The 7th byte encodes several switches and other binary states. The operator uses one switch to arm the tank, a second switch to fire the laser, and a third switch to toggle between the two cameras, while the receiver module can indicate either a loss of signal or a failsafe event using two additional bits.

The code running on the chassis Arduino reads these bytes and translates them to speeds and directions for all the motors. In the event of a loss of signal, all motors are deactivated after 100 milliseconds. A loss of signal event could either be a total loss of serial communication between the two Arduinos as would occur if the serial wire was cut, or a failsafe/signal loss event communicated by the receiver Arduino as would happen if the radio link was lost. In any case, if no valid new command information is received for 100 milliseconds, everything is deactivated. In practice this system worked very well and the tank never went out of control in a dangerous way. Some unexpected deactivations did sometimes occur. These could have been caused by radio link problems or with single bits getting flipped in the serial communication. Some error checking bits could have been added to fix the latter issue.

Translating user commands into movements was relatively straightforward. The drive motors could be commanded to spin either forward or backward and turning was achieved by spinning the two tracks at different rates. The stepper motor for the turret could be spun at different speeds using different step sizes, and the servo controlling gun elevation could simply be given a set position.

Additionally, the chassis Arduino sends 5 bytes every loop to the OSD Arduino. These bytes are a start byte, the RSSI byte, two bytes for battery voltage and current, and a byte for switch positions and failsafe status.

Figure 10: OSD Example Images

The OSD Arduino uses this information to draw the OSD, as seen in Figure 10 above. The information available to the operator includes the battery voltage, current draw, power-on time, and amp hours drawn. Both amp hours drawn and voltage are indicators of battery status. Over-discharging large lithium polymer batteries like the ones used for this project can be very dangerous, so understanding battery health is critical. The ”ARMED” flag at the top means that the drive motors are powered and the ”FAILSAFE” flag indicates that a communication timeout has occurred. Finally, ”LOW BATTERY” and ”BATTERY CRITICAL” warnings are displayed if the battery voltage drops below 14.8V or 13.3V respectively.

5 Testing and Evaluation

The goal of all this work was to develop a significantly improved remotely operated tank platform. To that end the project was very successful. The system was completed on time, and after a lap around campus all components demonstrated themselves to be operating as expected. Control was never lost in a dangerous way and no major unexpected behavior was experienced.

The new user input system was a vast improvement. The RC transmitter was not only significantly more ergonomic than the laptop but also a lot more intuitive and easy to use. Fast and precise control of all systems was made much easier, which made relatively risky high-speed maneuvers possible with high levels of confidence. At the same time, real-time first-person feedback made the vehicle navigable well beyond the operator's line of sight. It also made for a very immersive and entertaining driving experience.

All of the other improvements also added up. The increased drivetrain performance made navigating at high speed more reliable, and made getting out of sticky situations possible. The improved electronics layout and added protection inspired confidence even on bumpy terrain or over tall grass. And finally the increased battery capacity was huge. A trip from Young Orchard to the CIT to the main green and back was easy to do even when preforming expensive high speed tricks and other fun maneuvers.

6 Future Goals

In summary, this project took an existing remotely operated tank platform and upgraded it with mechanical improvements, a new electronics/software package, significantly improved user controls, and a live video telemetry system. Though the project was an overall success and the new system worked very well, even the second generation of the tank was imperfect, and there are always areas of possible improvement.

As was already mentioned, the failsafe automatic deactivation functionality occasionally tripped when it wasn't expected to. It is possible that these were genuine failsafes caused by a poor radio link. The RC transmitter/receiver are designed for long-range applications like aircraft, so they actually perform far worse in very close-range situations when the strength of the signal can overwhelm the receiver. It is also possible that these failsafe events were caused by flipped bits in the serial communication. Specifically, if the switch indicating the position of the "armed" switch was flipped from on to off, even for a single cycle, the system would disarm itself until the switch was properly flipped off and back on again. To prevent serial glitches like this from occurring, additional error checking code could have been added.

The most notable mechanical problem with the new system was the stepper motor for turret rotation. The updated aluminum turret was heavier than the previous version and the stepper motor proved underpowered for the task, especially at higher speeds. In fact, the much more powerful drive motors were able to spin the chassis of the tank fast enough to overcome the holding torque of the stepper motor and throw the turret around, which was not ideal. To solve this issue a more powerful stepper motor could have been used, though finding one that would have fit in the very tight chassis would have been difficult. An alternate approach could have been replacing the stepper motor with a more powerful DC or brushless motor. With the addition of either a rotary encoder or an IMU, a servo-like system could have been made to rotate the turret at constant speeds.

Given the flexibility and power of the current platform, many improvements can be imagined. One such improvement would be the addition of an IMU on the turret to detect and counteract the movement of the chassis. This kind of a system is often found on real tanks, and the effect would be a stabilized view from the turret that does not move as the body of the tank does. Compensating for both rotation and tilt could theoretically be done, so long as the motor controlling turret rotation can keep up with the rotation of the chassis.

Additional sensors could be added to aid in navigation and control. Front-facing object detection could be implemented to warn the driver of possible obstacles ahead. Low level autonomous driving could even be added for when the operator is looking through the turret camera. With more powerful onboard computing, a system that tracks objects in the line of sight of the gun could be added too. Systems like these would allow the human operator to switch back and forth between the roles of driver and gunner without the leaving the unattended system completely useless.

Finally, of course, higher levels of autonomy are always possible. With such large batteries and powerful motors, power consumption and weight are not real concerns. A very powerful computer could easily be added to the system along with many advanced sensors. Though the focus of this project was building a great ROV, with some work it could become a real autonomous vehicle, eliminating the need for a user interface altogether!