Enhancing Field Testing of a Teleoperated Robotic System with Wireless Handheld Controls

Dan Napierkowski, Eli Verbrugge, Alex Wortmann
Advised by: Dr. Kurt Kosbar
Missouri University of Science and Technology, Rolla MO, USA

ABSTRACT

This paper examines the usage of a microcontroller to wirelessly send commands to its destination using the Transmission Control Protocol (TCP). The system was designed to assist the testing of a Mars rover designed for the 2020 University Rover Challenge. RoveComm, the rover's custom message protocol, is used to wirelessly trigger an emergency shutdown of the rover's power systems as well as manipulate the robotic arm and science systems while testing out in the field. To facilitate reliable communication to the rover during testing scenarios, a TCP implementation for RoveComm was developed to ensure reliable message delivery to all rover control systems. The wireless controller features a 32-bit microcontroller with an integrated wifi stack which transmits signals with a 2.4 GHz external antenna to a pair of circularly polarized omnidirectional antennas mounted on the rover. The controller allows for transmission of custom RoveComm commands, reducing the hazard of having a physical operator within reaching distance and enhancing the rigor of terrain and operation testing that can be undergone.

INTRODUCTION

The Missouri S&T Mars Rover Design Team (MRDT) designs, builds, and tests a rover to compete in the annual University Rover challenge (URC). This competition is split into four tasks that are meant to simulate the challenges that a rover might face on a Mars mission. The first task is picking up objects weighing less than five kilograms as well as accomplish objectives such as typing on a keyboard or opening a drawer. Task two requires taking and analyzing soil samples. Task three involves traversing mountainous terrain such as rocky paths, steep slopes, and sharp drop-offs. The fourth and final task is moving autonomously to checkpoints along a set path [1]. MRDT's process to build a rover each year ends with five to six months of extensive testing to ensure that everything will work as intended for the competition. Figure 1 shows the 2020 MRDT Mars Rover outside with its robotic arm attachment.

Every rover competing in URC must be equipped with a physical Emergency-Stop (E-Stop) button. This is to ensure that if any rover malfunctions it can easily be shut down. This solution works for many scenarios, however being close enough to press the E-Stop also poses many problems, especially during testing. In order to ensure the safety of everyone it is necessary to remain away from the rover until the team is certain that it works as intended. Because of this, a wireless solution is needed to allow for the safe testing of the rover. The Wireless E-Stop, and Auxiliary Testing System were created to satisfy these needs.

The Wireless E-Stop is designed to send a power-down command to the rover from a minimum distance of six meters. As a handheld device it can be armed with the flip of a switch to prevent accidental activation. Which then allows for a button to be used as either a dead man's switch or as a trigger. When activated the Wireless E-Stop sends a command to the rover over TCP which tells the Battery Management System (BMS) to power off. A successful Wireless E-Stop will send the command quickly in order to minimize damage to the rover.

To effectively complete the tasks for competition, a manipulator and science system are needed. The manipulator is a robotic arm with six degrees of freedom that can pick up and finely control objects. The science system is designed to dig into the ground to collect soil samples, and drop them in a tube to analyze. Both of these systems require careful testing in order to work effectively. The rover is modular, which allows for independent testing of both the manipulator and science system. A method of controlling these systems independent of the entire stack is required. The Auxiliary Testing System was designed to wirelessly send commands to the manipulator and science system to check their functionality from a safe distance. It takes user input from a range of buttons and sends the command corresponding to the button.

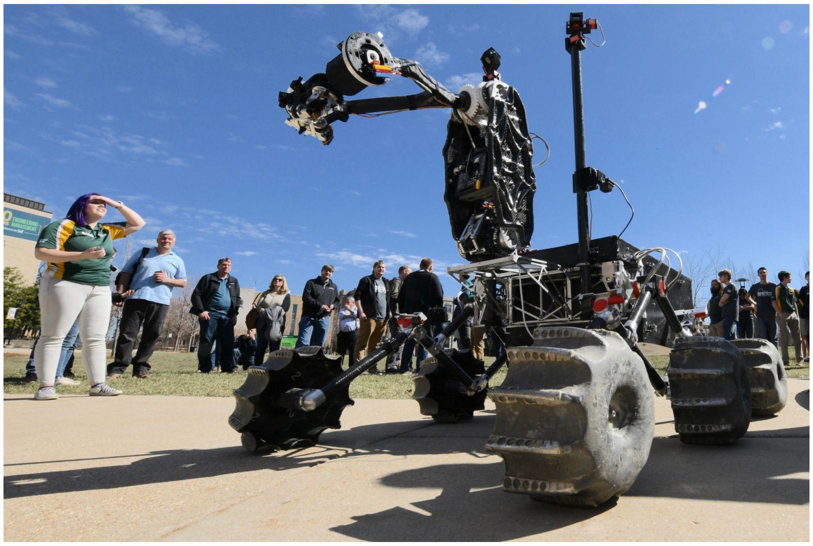

Figure 1: MRDT's 2020 Rover, Icarus, with the Arm attachment

The software was designed to work succinctly with the rest of the communication stack and uses TCP with a defined packet structure to communicate between the Wireless E-Stop or Auxiliary Testing System and the embedded management systems on the rover over the 2.4 GHz band. This allows for the transfer of data between different platforms, and alleviates issues in correctly parsing packets.

This paper focuses on the software and electrical systems of the Wireless E-Stop and Auxiliary Testing System used to wireless test rover systems. Many of the protocols used in this project are industry standards, and common practices which allow for the systems to be easily expanded upon by others. All of the code and electrical schematics are provided on the MRDT Github page under a creative commons license [2].

DESIGN

The Wireless E-Stop and Auxiliary Testing System use custom Printable Circuit Boards (PCBs). The Wireless Controller utilizes a small microcontroller with built in wireless and networking capabilities in order to minimize the amount of components needed.

The software is written in C++ and uses the MRDT RoveComm library to ensure that messages are successfully parsed on the Rover side. RoveComm was extended to provide support for TCP in addition to the User Datagram Protocol (UDP) as reliable communication was deemed necessary for a mission critical command such as that of the E-Stop. The software will read inputs from the various onboard switches/triggers and will also help process and indicate any incoming critical telemetry from the rover.

The Wireless E-Stop is designed to be ergonomic, and robust. Ergonomics are important, it must be easy to access and hold when out in the field as this system will be the first line of defense against rover failures. Additionally, the Wireless E-Stop uses the same frequencies as used by basestation in order to prevent the need to add hardware to the rover. This allows for the Wireless E-Stop to be used on any of MRDT's rovers and should allow for the Wireless E-Stop to be continued to be used in future years without additional work.

SOFTWARE

The core network library is shared for the Wireless E-Stop and the Auxiliary Testing System; custom messages are sent to the various onboard electrical systems. Both require acknowledgement from the receiver, to ensure that commands are executed and no communication issues arise. Each packet is sent using RoveComm, our custom message protocol which helps all systems on the rover quickly communicate with each other. This protocol is implemented in C++, Python, C#, JavaScript and Dart in order to be portable across various operating systems and embedded devices and can be implemented in any language with support for the TCP/IP [3] stack.

The RoveComm protocol defines a message to contain the following information:
- 1 byte to indicate the protocol version.
- 2 bytes to specify the data ID. This represents the identifier assigned to this piece of telemetry and is used by subscribers to identify what data is being received.
- 1 byte to represent the type of data being transmitted, e.g. char, float, integer.
- 1 byte to represent the number of unique data entries.
- Up to 1020 bytes for actual data in the payload.

Version	Data ID	Data Type	Data Count	Data
1 Byte	*2 Bytes*	*1 Byte*	*1 Byte*	*Up to 1020 bytes*

Figure 2: RoveComm Packet

The initial RoveComm implementation only used UDP due to low bandwidth and signal strength conditions imposed by competition in a hilly terrain. Because line of sight was not always guaranteed, having guaranteed delivery caused packets to back up and cause the rover to behave sporadically and unexpectedly when signal was reacquired. UDP was far more responsive and usable in this scenario. However, as this testing system is employed solely in light of sight situations, in this case TCP provides additional benefits in mission and safety critical communications. TCP is used to send and deliver critical commands as well as receive telemetry from the rover systems during testing such as battery voltages, motor positions and calculated velocities.

Both the Wireless E-Stop and the Auxiliary Testing System employ a variety of momentary tactile switches as inputs for various commands. The microcontrollers used for each of these systems poll for input from each of these switches and execute the specified command every twenty microseconds. For the Wireless E-Stop this is sending a kill switch transmission to the BMS, and for the Auxiliary Testing System a variety of extensible Science and Arm actions are performed. The software is modular, and designed for easy modification as testing progresses.

NETWORK

Data is transmitted between the rover and Wireless E-Stop/Auxiliary Testing System over a 2.4 GHz RF link, and packets are distributed over the Rover's network switch. The networking infrastructure is scalable, and thanks to the RoveComm protocol, packets can be transmitted to a subset of embedded systems on the rover, allowing for more direct communication and less network saturation. This distributed architecture allows for the Auxiliary Testing System to specifically target the Arm or Science systems when sending testing commands.

For competition tasks data is transmitted between rover and base station over a 900 MHz RF link due to the necessity of long range non line of sight communication. For testing and task simulations usage of a 2.4 GHz link is usually preferred as 2.4 GHz usually provides a good balance of range and bandwidth. Additionally, 2.4 GHz is supported by a plethora of commercial hardware and it allows for laptops and mobile devices to communicate with the rover.

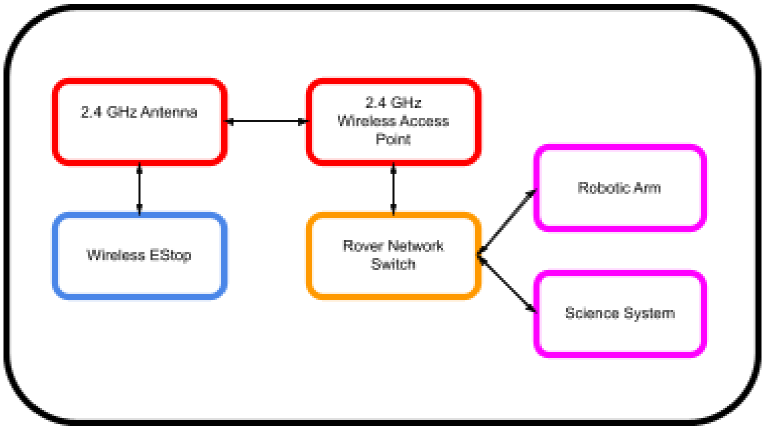

Figure 3: Network Diagram

PRINTED CIRCUIT BOARD

The Wireless E-Stop is controlled by a 32-bit microcontroller with a native full wireless stack. This microcontroller requires steady 3.3 V power so a TPS736 voltage regulator is used to drive down the 3.7 V battery to 3.3 V. Capacitors are used to smoothen out the voltage supply in order to reduce risk of destroying the device. The microcontroller uses Universal Asynchronous Receiving Transmitting (UART) to flash code. A CH340E USB to UART adapter was used to flash code to the microcontroller from a standard computer connection. A Micro USB port was used to connect the PCB to a computer to be flashed, but also to supply power to the battery. LiPo batteries require a charge controller to prevent overheating, as well as step down the input voltage from the USB. An MCP73831 charge management IC was used because it was within scope and had status LED indicators to alert the user that the battery is charging as well as when it is fully charged. To know when the battery is low, a simple voltage divider fed into an ADC is used. When software detects a low voltage an indicator LED is activated, and if the voltage falls low enough the microcontroller will turn itself off. This is to protect the battery from accidentally being left on when not in use.

The wireless E-Stop is activated with a button that connects to the microcontroller. Depending on the level of danger in a given test, the user can choose between running the E-Stop in an active-low or an active-high setting. This toggle was done on the PCB by utilizing a single pole double throw switch as well as an N-type mosfet. Using this method allows for the microcontroller to react on an active-high signal regardless of the switch orientation.To keep their eye on the rover, haptic feedback was added to provide information. Haptic feedback is created with a small 1.8 V motor being driven by a DRV26Q controller. It was selected for its embedded library of predefined haptic modes and Inter-Integrated Circuit (I2C) communication support.

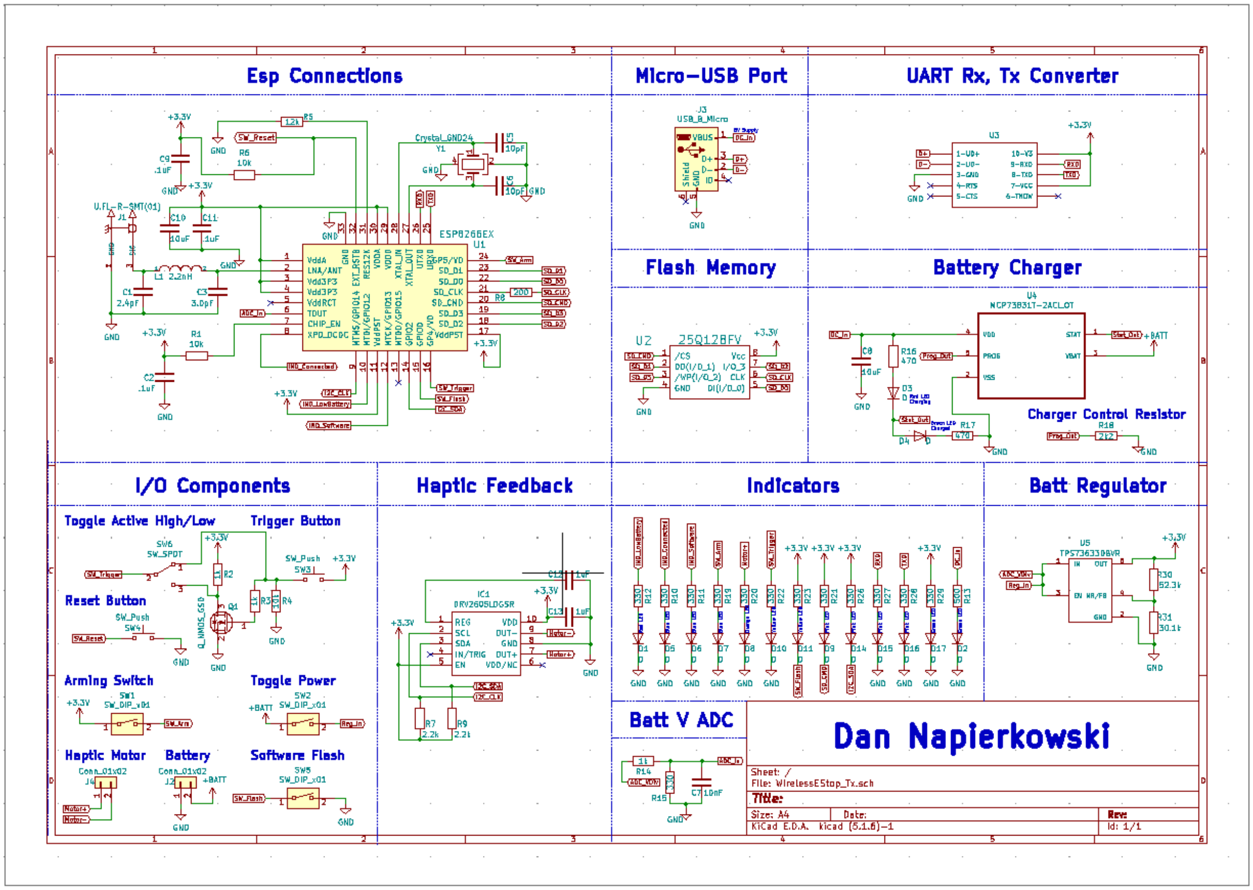

Figure 5: PCB Schematic

The wireless E-Stop was made on a single, double-sided PCB in order to reduce cost. The PCB needed to be compact in order to fit all the necessary additional components such as battery and haptic motor, and still fit comfortably in a user’s hand. All resistors, capacitors, and inductors were 0603 US standard in order to strike a balance between small form factor and ease of hand-soldering. Traces connecting to the microcontroller followed hardware guidelines provided by the hardware designer [4]. All signal connections were kept short to reduce capacitance on the line, and stitching vias were placed to keep the PCB well grounded. Holes were added to easily connect to a case for the device. Any non-critical components were removed after the board was tested and flashed to conserve battery life.

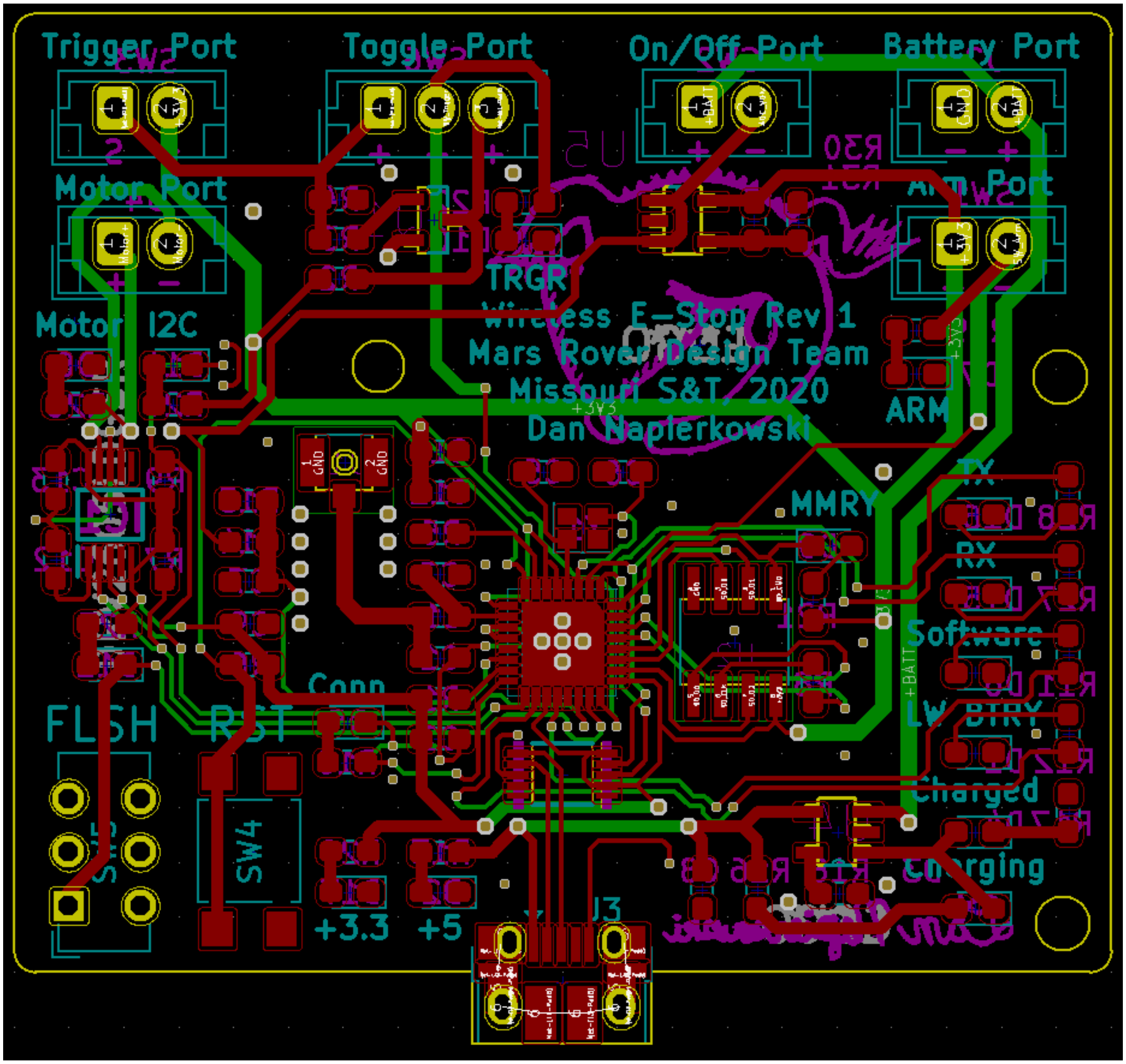

Figure 6: PCB Layout

CONCLUSION

The core software, hardware and network stack proved to be robust. After some experimentation with the chosen microcontroller, rigorous bench tests were done in which the controllers were able to send and receive telemetry and commands with no discernible latency. The systems were functional for their testing purposes, and provided the remote functionality desired.

Progress is being made in developing more fine tuned control hardware for the Auxiliary Testing System, as well as providing network status indicators to the remote control operator. Further testing will show the reliability of the chosen hardware in longer distance scenarios.

RoveComm has been successfully extended to include support for TCP and UDP, with all rover systems being able to initiate both forms of communication. Bench testing has been performed to ensure the utility of TCP in long range rover operation, and it was quite successful. Rover compute units can request to subscribe to a system's telemetry and depending on the criticality of the data it will be delivered over either TCP or UDP. It has also been tested with mobile devices connected to the rover access point, and can be extended to support future iterations of this platform, perhaps as a mobile application.

The Wireless Estop and Auxiliary Testing System are the foundations of advancements of testing capabilities the Mars Rover Design team hopes will allow for more rapid and robust testing in the field. The ability to test the rover in a variety of situations that would be unsafe with a traditional physical E-Stop is invaluable, and being able to rapidly test the functionality of the Arm and Science system allows for non-electrical/software members to test their systems without in-depth technical knowledge.

Due to the ongoing global pandemic, Rover testing was largely halted. In the testing that was performed, the Wireless E-Stop proved to be a well functioning system that will help enhance the testing rigor performed by the team. The auxiliary system will in turn be incredibly beneficial in performing system tests without the complexity of a full Rover network and control stack.

REFERENCES

[1] “Requirements & Guidelines.” *University Rover Challenge*, The Mars Society, 2019, urc.marssociety.org/home/requirements-guidelines.

[2] “Missouri S&T Mars Rover Design Team.” *GitHub*, Mars Rover Design Team, github.com/MissouriMRDT.

[3] “The Internet Protocol Stack”*,* Frystyk, Henrik, July 1994, www.w3.org/People/Frystyk/thesis/TcpIp.html.

[4] “ESP8266: Hardware Design Guidelines”, Espressif Systems, 2019 https://www.espressif.com/sites/default/files/documentation/esp8266_hardware_design_guidelines_en.pdf