A DISTRIBUTED SENSOR NETWORK FOR AN OFF-ROAD RACING VEHICLE

Kyle Boyer, Laura Brubaker, Kyle Everly, Richard Herriman, Paul Houston, Sean Ruckle, Rory Scobie, Ian Ulanday
Advisor: Dr. Michael Marcellin
The University of Arizona

ABSTRACT

The University of Arizona Baja Racing Team competes annually in an intense off-road racing competition. This year's car features a distributed sensor network capable of displaying useful data to the driver, the benefits and technical aspects of which are examined by this paper. Based on the ATmega2560 chip, the system is USB programmable, features hot-swappable batteries, and includes SMD components. Each sensor is custom designed, functions as an I2C slave, and contains its own ATtiny85 microcontroller allowing all the sensors to be addressable and enables them to be wired in parallel. The system also includes interrupts for almost every single sensor, which allows for more accurate data collection and guarantees that no important data will be missed. A custom-made board was created to connect these sensors and serve as a microcontroller data logger based on an Arduino reference design.

INTRODUCTION

In 2012, a few members had the idea that a speedometer would be a valuable addition to the vehicle. A speedometer would allow for constant monitoring of speed, which would enable the driver to make consistent turns and jumps. The following year, a new speedometer system was built, with the addition of a fuel gauge. The team also decided that receiving this data in real time would be extremely beneficial to the team. Therefore, the next system was designed to be capable of using radio communication to send speed and fuel data back to the pits, creating the team's first real time monitoring system. At the end of the year, it was decided that a separate subteam completely dedicated to the development of electronic and sensing equipment would be created. As the years have gone by, the Electrical Subteam has grown, gained experience, and increased its access to equipment. These resources have allowed for the system to develop into a completely custom sensor network capable of detecting and sensing dozens of parameters on the vehicle.

MOTIVATION

As a university club, we are first and foremost focused on education. As such, we approach each project with the goal of learning - and enabling others to learn - something new. At the same time, we strive with each project to improve our system in some way. As the complexity of our system increases, we are attempting to break up the project into a collection of smaller projects, each with reduced complexity, while also expanding the capabilities of the system.

DESIGN REQUIREMENTS

Durability

During any given race, the vehicle may encounter every type of terrain, from flat desert, to rock fields, to mud pits. To survive these conditions, all electronic device placed on the vehicle must be waterproof, dust proof, and strongly resistant to vibrations. To keep our devices waterproof and dustproof, we’ve implemented several different strategies. For anything we wanted to be accessible later, we designed housings that were either 3D printed or laser cut from polycarbonate, sealed with a silicone based gasket material. For devices that never need to be opened, permanent solutions are potting or conformal coating. In the interest of repairability, we preferentially use an ESD safe acrylic conformal coat over potting.

Connectors

In order to protect the internal electronics from dust and mud, all external connections in the system are made with IP68 rated (waterproof/dustproof) connectors. All wiring interconnects in the harness are achieved using either waterproof connectors or crimp connectors with integrated marine heat shrink tubing.

The kill switches on the vehicle work by shorting the spark plug contact to ground. When not engaged, unshielded kill-switch wire acts as an antenna. To avoid coupling between the kill switches and the data lines, we decided to use burial grade shielded CAT5e cable for all data transmissions on the vehicle.

While this does work, it turns out that ethernet insulation melts at a very low temperature. This is exacerbated by the copper data wires in the ethernet cable, whose high thermal conductivity cause the insulation to begin melting before the solder does. Because of this, we plan to restrict our use of ethernet cable to networking applications in the future, and use stranded wire for other data and power connections.

Modularity

We are constantly attempting to improve the system by adding additional features and removing bugs, so the ability to replace, add or remove entire subsystems without destabilizing the overall system is very useful. Further, by distributing the computational load across all devices in the system and using standard communication protocols wherever possible, we reduce the damage caused by a single component failing. This also reduces the additional development effort required to integrate additional sensors.

I2C

I2C allows all sensors on the vehicle (up to 112 - See Appendix 4) to operate in parallel on a single bus. This design decision was made after the electrical interface on the data logging and communications hub was finalized for this generation of the system. Using I2C reduces the number of pins required on the main board from 48 to 20, which will allow us to drastically reduce the footprint of the main board in future generations. I2C also opens up the possibility of integrating the microcontroller based data logger into a Raspberry Pi “Hat” as a means of further reducing the mechanical footprint of the board.

Unfortunately, the ATtiny85 does not natively support I2C (See Appendix 2). Using the ATtiny85 would require writing software to implement the protocol on GPIO pins. This functionality is not trivial to emulate in software, which is part of the reason we switched to the ATmega328 based Arduino Nano for the sensor implementations. We have since decided to switch from the ATmega328 chip to the ATmega32u4 for future implementations. This chip is used on the Arduino Leonardo, which is the same form factor as the Uno and has all of the same capabilities as the ATmega328. In addition to that, it has built in support for USB, meaning it does not require an external FTDI chip to be programed over USB.

Processing time

One of the major concerns with a growing number of sensors is the processing time required to poll them. While interrupts help by allowing some measurements to be made asynchronously, as the sensor load increases the interrupts begin to encroach on the functions that process and store data. Due to the small amount of memory available, data must be written to nonvolatile storage very regularly - roughly every 50 readings of the sensors. If this process is interrupted too often, it may not get a chance to complete and data samples may be lost. It also does not leave much room for calculations.

One solution is to use a microcontroller with more memory. Unfortunately, the ATmega2560 already the largest memory capacity of the Arduino supported 8-bit microcontrollers (See Appendix 1). Another solution - the one we decided to go with - is to use multiple microcontrollers to distribute the processing load and memory required to store multiple samples of data for averages and integrations. This data can then be polled at set intervals by a master device, which can then write the data to a microSD card or other nonvolatile storage.

Form Factor

The vehicle is a small scale single seat racecar, so form factor is a major concern. There is a limited amount of space available, especially with all of the standard racing equipment inside. There are only two small compartments that are available to place all of the electronics in. This forced us to minimize the PCB footprint.

SMD

In order to make the system more compact this year, almost all of the components on the circuit boards are surface mount, rather than through hole. Though SMD parts are more expensive and difficult to solder, we felt that it would be an excellent learning experience for our members. This did impose a learning curve, as no members of the team had experience in surface mount, but after a fair amount of research and practice, we were able to successfully mount all of the components. These included components that required reflowing or drag soldering, such as the main microcontroller which was a TQFP package ATmega2560 or the 28 pin SSOP package FTDI chip. Using these surface mount components made it possible to have more LED indicators and functionality, yet still maintain a relatively small form factor. We are in the process of sourcing inexpensive components and designing a board that is just to assist in training our members.

PCBA not shield

In previous years, an Arduino shield that was directly compatible with the Arduino Mega was designed and produced. As the system gained complexity, the shield’s footprint exceeded the footprint of the Arduino Mega. Because Arduino is an open source hardware with open source schematics, it was possible to produce an Arduino compatible circuit board that includes the extra functionality that we previously added with a shield. This circuit board used the same microcontroller as an Arduino Mega, had all the circuitry necessary for the microcontroller to function, and all of the components that would have previously been added by a shield. One of the convenient features of developing a custom board, was eliminating the typical Arduino headers and instead being able to route all the outputs and inputs to a custom waterproof connector (Figure 1).

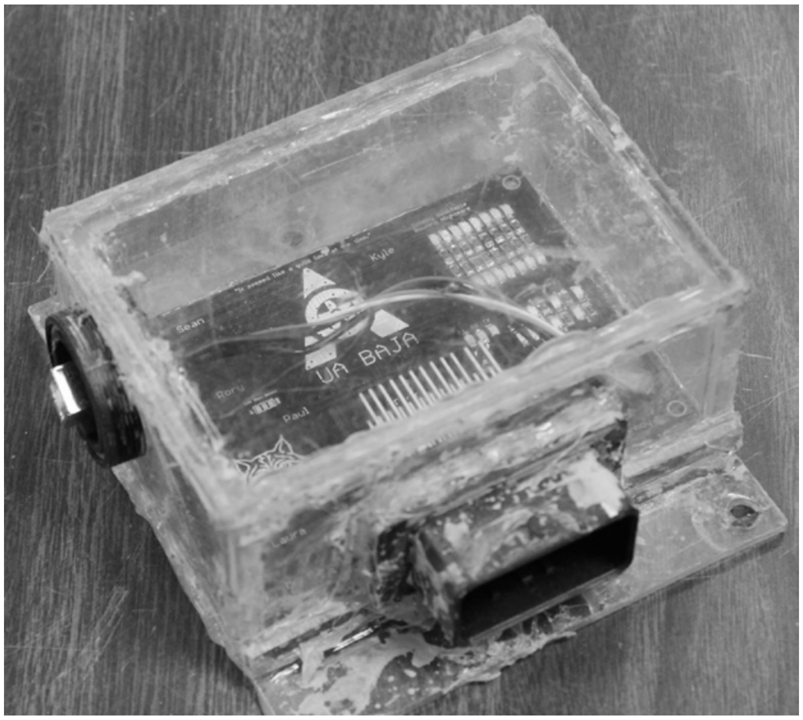

Figure 1 - Waterproof Connector Post-Race

Reliability

COTS batteries

This year the rules for the competition were updated to no longer exclude the use of lithium based battery chemistry. In previous years teams were restricted to sealed lead acid (SLA) batteries and this change allows for a much higher energy density. However, using lithium batteries comes with the added risks of overcharging or draining the batteries, which can result in the battery catching on fire or exploding. These risks are mitigated by using proper charge and discharge circuitry. In light of this, we chose was to use COTS lithium ion drill batteries as they are designed to be durable and have all the needed protection circuitry built in.

Power Supplies

Due to the high premium placed on weight in this competition, the batteries are necessarily quite small. To compensate, any unnecessary inefficiencies must go. For this reason, the linear regulators from the old system were replaced with higher efficiency switching regulators. To reduce weight, a single regulator for each voltage is mounted to the firewall and the output is routed to the rest of the vehicle through shielded wires.

Where switching regulators can achieve efficiencies above 90%, linear regulators can be approximated as voltage dividers for efficiency calculations, and frequently operate at less than 50% efficiency. The 12V to 5V converter on the board is intended solely as a backup in case the

5V power from the switching regulator fails. It operates at around 42% efficiency. The 3.3V regulator achieves 66% efficiency (See Appendix 3) by using a 5V input instead of a 12V input to reduce the voltage difference and therefore the wasted power.

Interrupts

One of the main reasons we created our own custom sensor modules was so that we could connect every sensor that was logically to an interrupt enabled pin. Using interrupts to collect the data means that no data points would be missed due to the system polling the sensors at the wrong time. When the sensor triggers the interrupt, the sensor module is able to store that data until the main board requests it. This means that the main board can poll the sensors without risk of losing any data points.

Availability

Battery swap

Last year, the telemetry system and brake lights both ran on a single SLA battery. Due to the small capacity of the battery, it had to be replaced every hour. Having the battery run down so quickly is a significant risk in this competition, as the car may be black-flagged (removed from the competition) if the brake light fails to function properly.

To help alleviate this concern, we have separated the electrical system into two distinct and isolated subsystems. One subsystem consists only of the brake lights and a small lithium ion drill battery. Based on the estimated power consumption of the lights, the battery should sustain several hours of continuous use. Since the brakes are only engaged intermittently, and the race is only 4 hours long, one battery is more than enough for the entire competition.

The second subsystem consumes far more power as it contains all of the telemetry and communications gear. For both ease of use and safety of the electrical system, we used 12V and 20V drill batteries for the brake system and telemetry system respectively. We also custom designed battery mounts so that the batteries could be removed and replaced quickly during pit stops.

Because the telemetry and communications system requires a non-trivial time (about two minutes to boot and reestablish connections) to come online after a power interruption, it is desirable to have a means of replacing the battery without a reboot. To that end, the system uses two batteries in parallel, with a diode in series with each to prevent them from back feeding into each other. Either battery may be removed without interrupting the flow of power to the load.

Compatibility

CAN

The CAN protocol is used in most commercial vehicles as well as a host of other products. In an effort to be compatible with as many devices as possible, the main board has a CAN controller and transceiver chip. The associated data pins are broken out to the board mount connector and can be used to connect any device that supports CAN.

USB Programmable

The ATmega2560 must be programmed before it is usable. Two options are available for this: ICSP and USB. Each has its own benefits.

Programming over ICSP is reliable and works with no modification to the system. The main drawback to this method is that it requires an external device to perform the programming operation. If the device must be programmed repeatedly, the ICSP header must be broken out to an accessible location.

Programming over USB requires burning a bootloader to the device, which reduces the system resources available for the program itself. The benefit of this method is that the microcontroller can be programmed over UART. Consequently, a host computer can both program the system and communicate with it easily after programming by using a USB to serial converter IC. A common USB to serial converter is the FT232RL by FTDI, which has freely available drivers.

The Arduino bootloader is the first thing that runs when the chip is powered and it listens for programming attempts on the UART receiver - if it detects one, it proceeds to program the microcontroller. If it doesn't find a device trying to program the microcontroller, the bootloader exits and the program that was previously loaded on the microcontroller runs. This means that the microcontroller can only be programmed immediately after it is powered.

There are a few ways to accomplish this. The method we used was to start uploading the program and hit the onboard reset button at the right time. This required exact timing, or the upload would fail. The other (more ideal) method allows the computer to reset the microcontroller. The programing sequence on the computer sends a command to the FTDI chip when the reset is needed and one of the FTDI's pins is pulled high, which causes the reset. On our first boards, the trace that connected that pin on the FTDI to the reset line was missing, which is why a manual reset was needed. We tested that adding that trace would fix the problem by soldering a jumper wire between FTDI pin and the reset line. With this modification, the upload process worked with no manual reset. We have added the trace on our new designs.

Microcontroller Choice

Arduino is one of the most popular and well-known platforms for rapid prototyping, and as such there are a wide variety of hardware and software components available for it. By maintaining electrical (but not mechanical) compatibility with the Arduino, the custom board can use libraries written for the Arduino with little or no modification.

ATmega2560

The ATmega2560 that we chose to use on the main data acquisition board is the same chip that is used on the Arduino Mega. We chose this microcontroller over others like the ATmega328 (used on the Arduino Uno and Nano) or the ATmega32u4 (used on Arduino Leonardo and Arduino Pro Micro) because it was the most versatile of the different chips we considered. With multiple UARTs, built-in support for I2C and SPI, and the largest memory capacity in its line, it provides a wide range of options for future expansion.

ATtiny85

This year the custom sensor boards used the ATtiny85 was the microcontroller. This is a small IC with only 8 pins in total. Its small size and commensurately low cost make it an attractive choice for this application. In addition to that, there are also existing open source designs that make use of it. The small package does come with some drawbacks, however. The ATtiny85 lacks a UART, reducing compatibility with devices. It also lacks native support for I2C, which meant to be used in our distributed sensor network we would have to emulate I2C in software, reducing the speed and reliability. I2C and SPI are actually both listed as supported on the product page, though third-party libraries are required to make use of either. We are looking at the ATmega32u4 as a more fully featured replacement with a similar form factor.

CONCLUSION

This year provided our team with a lot of learning opportunities. Designing our own microcontroller board was an entirely new challenge for our team, but each of our members came away from the experience with greater knowledge, skills, and appreciation for telemetry. This project has also opened doors for our team in terms of future advancement. Using 5.4 GHz was also a new advancement that taught many lessons, but also exposed the team to new ideas and methods.

ACKNOWLEDGEMENTS

Thank you to Dr. Michael Marcellin for making this project possible, our teammates on the Baja Racing Team for putting up with the billowing clouds of blue smoke and solder flux, and Josie Ruiz for always being there for us.

APPENDIX: DATASHEETS

1. Microchip Technology Atmel ATmega640/V - 1280/V - 1281/V - 2560/V - 2561/V
2. Microchip Technology Atmel ATtiny25/V - 45/V - 85/V
3. National Semiconductor LM1117
4. NXP Semiconductors UM10204 I2C Bus Specification and User Manual