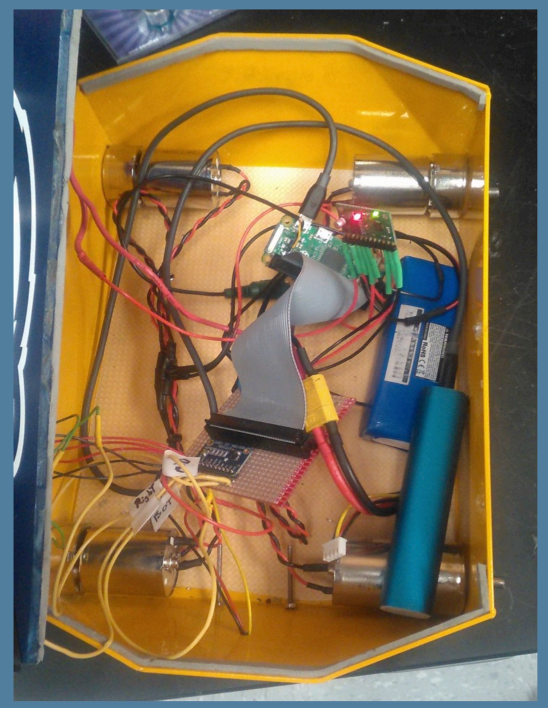
Increased Processing Power for an Autonomous Rover

Brandon Williams[1], Cody Helms[2], Arick Sweitzer[2], and Ryan Angstead[2]
[1]School of Mathematical Sciences and the [2]Department of Physics and Astronomy, University of Northern Colorado, Greeley, CO 80639

Abstract

Autonomous robotics consists of algorithmic protocols for independent navigation within an environment. With the growing field of extraterrestrial exploration, a key tool used are rovers. Rovers are complex or simple robots that move along given paths as they navigate towards a selected location. A major dilemma with current rovers, for example the NASA Mars Rover, is that they have little to no self-operating protocols to handle any adjustments needed given a change in their environment. An autonomous system allows the rover to react to circumstances requiring immediate action. Indeed, any transmission between a base station and the distant rover requires time; according to NASA, the time delay for communication to the rover is approximately 20 minutes. Our objective is to develop an autonomous rover for testing within a Mars-like terrain. The rover will have key embedded systems including a Raspberry Pi Zero Microcomputer, Arduino Fio Microcontroller, Xbee Radio Signal Receiver, and a Pololu Qik motor controller. Our system is more capable than our earlier designs in that it uses a microcomputer for a larger processor and a memory size, which is only limited to the size of the hard drive to store data and programs. This design is more efficient in that it accommodates more sensors and allows for processing of data from these sensors at a higher rate. Moreover, a higher number and wider range of algorithms for reactions to sensory input are possible.

Interior view of the rover

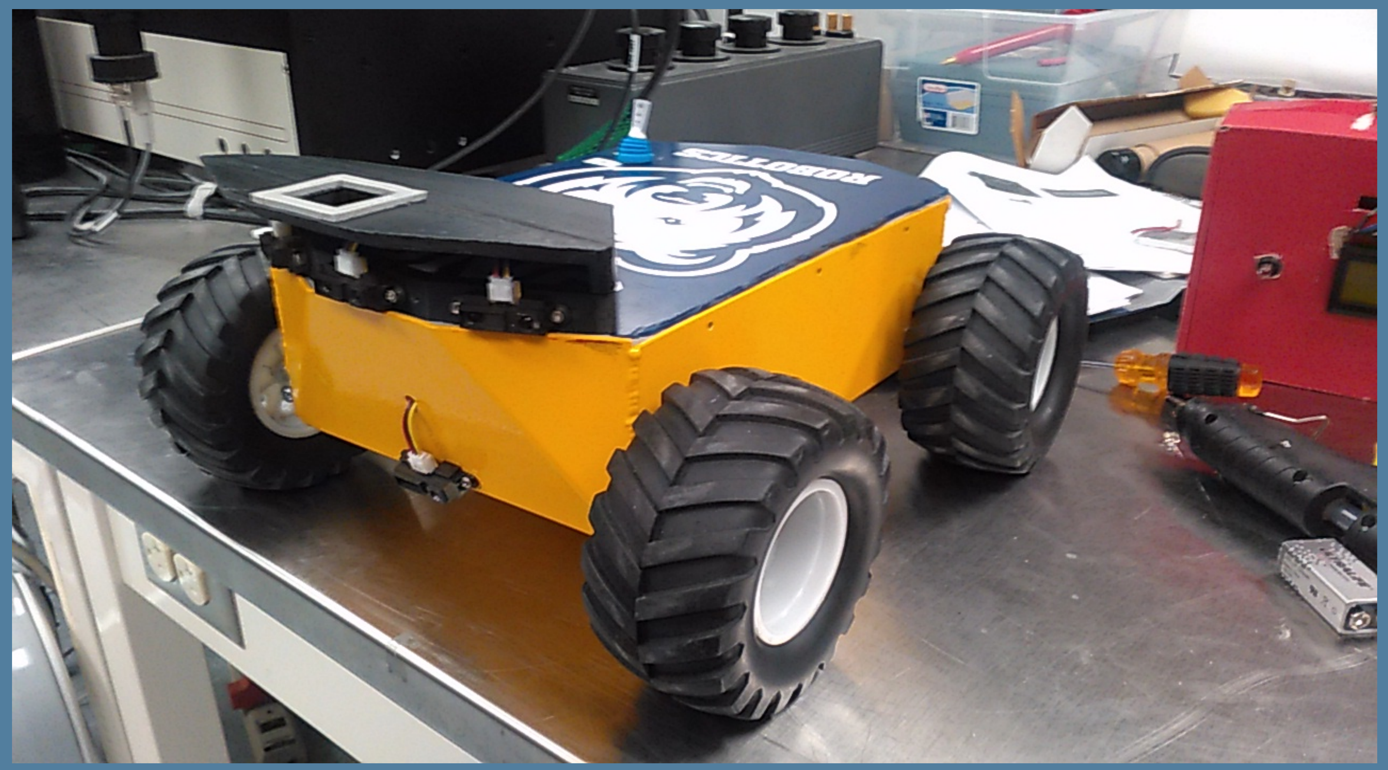


Primary Objective

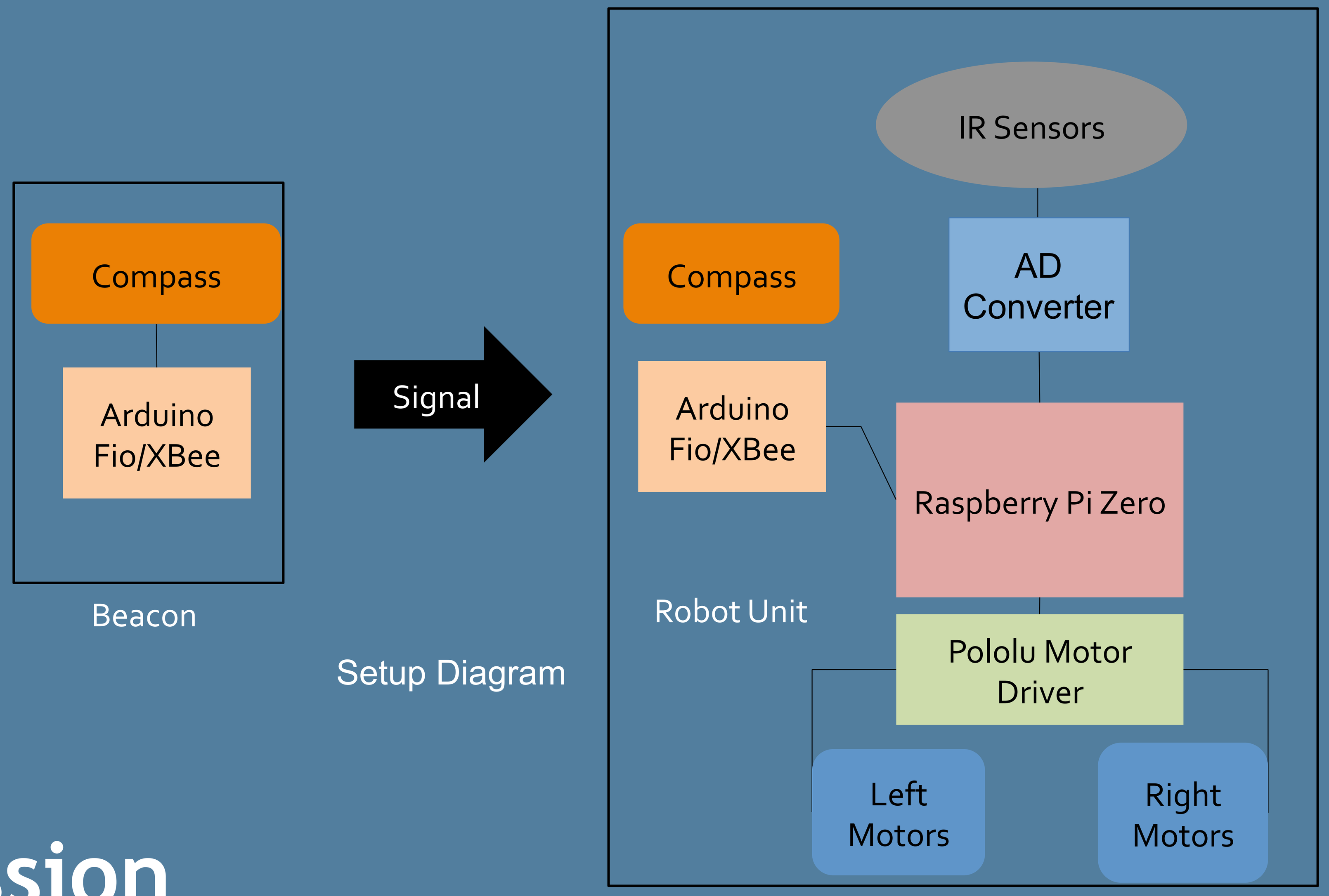
The primary objective is to increase rover performance through a change of the main control unit. Previously the main control unit was an Arduino Uno microcontroller along with two Uno compatible motor controllers. This system was limited to a constant memory size for data and software. The benefit of this system, however, was that it had a compact collection of inputs/outputs for conveying either digital or analog data; you could control many devices from a single source. This control was limited to only 32 KB of flash memory space. With the goal of exploration, especially extraterrestrial, data collection and data processing is the highest priority. With this need, we decided to use a small computer, the Raspberry Pi Zero, rather than a microcontroller. They compare as below:

	Raspberry Pi Zero	Arduino Uno
Processor Speed	1 GHz	16 MHz
Memory Capacity	N/A*	32 KB

As shown above, the Raspberry Pi Zero has a higher processing speed. The memory capacity of the Zero is indeterminable due to it being limited to the capacity of the SD Card* that contains the operating system. These cards can vary from 2 GB to 32GB and are constantly being improved. Also, the processing power is largely different. The Raspberry Pi Zero is 62.5 times faster than the Uno.



Exterior view of the rover



Setup Diagram

Discussion

Seen above are pictures of the exterior and interior of the rover. In addition, a block diagram shows the communication between the beacon and the rover, along with the principle components of each. During the Colorado Space Grant Robot Challenge, our rover will attempt to navigate an obstacle course while moving toward a radio beacon. The increased processing power and additional sensors are expected to improve the rover's ability to navigate the course.

Some issues for concern include insufficient battery life, potential central processor unit (CPU) over heating, and module port size limitations. Firstly, the Raspberry Pi Zero's power consumption is rated at approximately 120 mA (0.7 W) for the worst case idle state. An estimation of power usage of our system using the Raspberry Pi Zero alone reaches 250 mA. This issue is not a main concern due to cheap battery options, such as a 2,500 mAh Li-ion battery, being able to supply power during this state for approximately 10 hours. CPU overheating could exist due to long complex programs running continuously. Yet, during tests of motor driving algorithms, the CPU temperature never rose enough to be hot to the touch. So, since the temperature rise is small, we can neglect this drawback as well. Lastly, the module port size limitations are the most significant due to the small size of the Raspberry Pi Zero. To address this problem the main control unit could be simply swapped for a larger Raspberry Pi, such as their model 3 version.

Further development using this platform could vary greatly to adapt to multiple research projects. This development could include module application program library development, advanced autonomous algorithms, and parallel computing cluster development.

Acknowledgments: Matthew Semak, Robert Walch, Cynthia Galovich, Brian Sanders, Chris Koehler, Bernadette Garcia, COSGC, NASA, and Rocky Mountain WaterJet