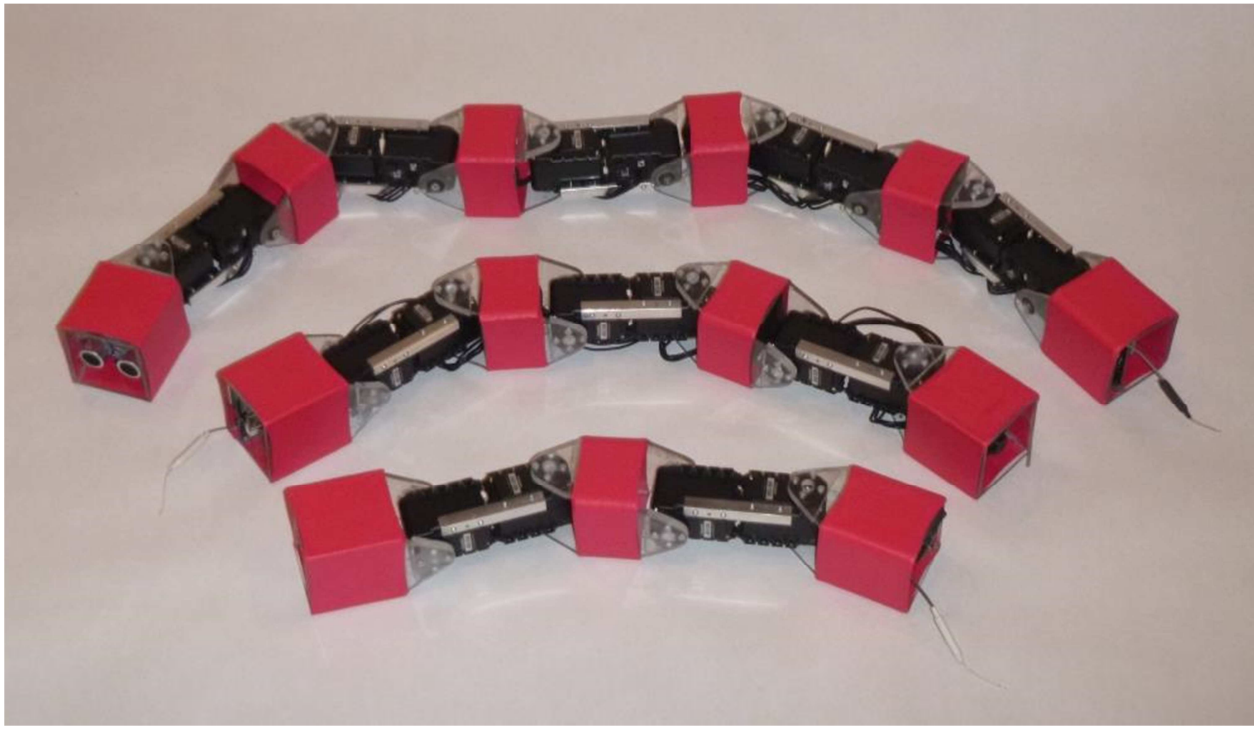

Modular Robotic Snake

Designed and Built by:

Christopher Atwood

Project # micro13CA295

Introduction

The goal of this project was to build a biologically inspired robot with several applications for military use. The modular aspect enables many aspects of the robot to be customized for the operation, including the number of segments, the sensor package at the front of the robot, and the control package at the rear. Custom mechanical structure and electronics enable the snake to be robust enough for operation in many types of environments while being quick to repair. After construction, many types of motion would be developed to help the robot move, including motions similar to caterpillars and snakes.

The robot can be deployed as a mobile robot to go investigate a situation, possibly before a medic team moves in to help with casualties. It can travel through pipes or small spaces and check for hazardous items or chemical scents. The robot can be deployed in a stationary location to provide a lookout where leaving a soldier would be too dangerous.

Design Choices

There were two main goals driving the design of this robot: to have it be self-contained with no tether, and to be modular so it can be reconfigured easily. The self-contained objective is a challenge because batteries carried onboard the robot take up both space and weight that the robot has to deal. However if the robot had to pull a tether behind it, that could add up to a significant amount of weight to pull depending on the length of the tether. For a robot to operate in the field effectively in a battlefield situation, it would need to be self-contained so that it is quick to deploy and the operator doesn't have to worry about a tether getting tangled, severed, or stuck on an object. Also there is the possibility during a stealth reconnaissance mission that an enemy soldier could trace the tether back to the operator.

Another major design choice was to design the robot with a square body instead of a round body. A round body robot might have an easier time navigating certain obstacles, but might not be able to tell its orientation predictably. There are several advantages of a square body robot compared to a round body robot. Most importantly, when the robot is navigating on most terrain, the robot can know that it is on a distinct side which aids in programming motion sequences by having a reference orientation. Also a square body enabled me to pack the contents of the robot into a smaller diameter for this design compared to the round robot I modeled, due to the circuit boards, battery packs and other items having rectangular shapes to begin with.

Systems Module Overview

The robot consists of individual modules that connect to each other. There are 4 basic modules: tail segments, motor modules, cargo segments, and head segments. The robots can be configured in many ways but need to alternate motor modules and other modules, and have a tail and a head module at either end of the robot, as shown in Figure 2. The real flexibility with the robot shows up in choosing segments that are outfitted with the right equipment for the mission.

Figure 1: Overview of Snake Configuration

Tail Segments

The main brain of the robot is located in the tail. There is a custom designed circuit board that allows the microcontroller to interface with the rest of the robot. There is a tilt sensor installed here so the robot can detect its current orientation and a wireless R/C receiver to receive commands wirelessly from an operator. There is also a power switch to turn the entire robot on or off, a voltage regulator to provide the correct power for the microcontroller, and a charging plug to charge all of the installed batteries at the same time.

The amazing thing about having such a modular robot is that it is not restricted to one microcontroller. Each microcontroller has its own pin-out, voltage requirement, and footprint, so the brain board needs to be custom designed, but once installed in a tail segment it is compatible with the rest of the robot. The first microcontroller that I developed the brain board for was the MBED LCP1768 microcontroller, and I have focused software development on this one. But I have also developed a brain board for a Parallax PropStick USB which is ideal for carrying out multiple tasks at the same time, so it would be a perfect choice if multiple environmental sensors were to be monitored as the robot navigated.

Having the option for various microcontrollers to be used allows complete customization of the hardware. If before a mission an operator needed to write a new program, they can have their choice of microcontroller based on personal preference as well as mission parameters. Once in the field, multiple pre-programmed tail segments could easily be swapped out to change the robot from one function to another without having to bring a laptop along to do the reprogramming.

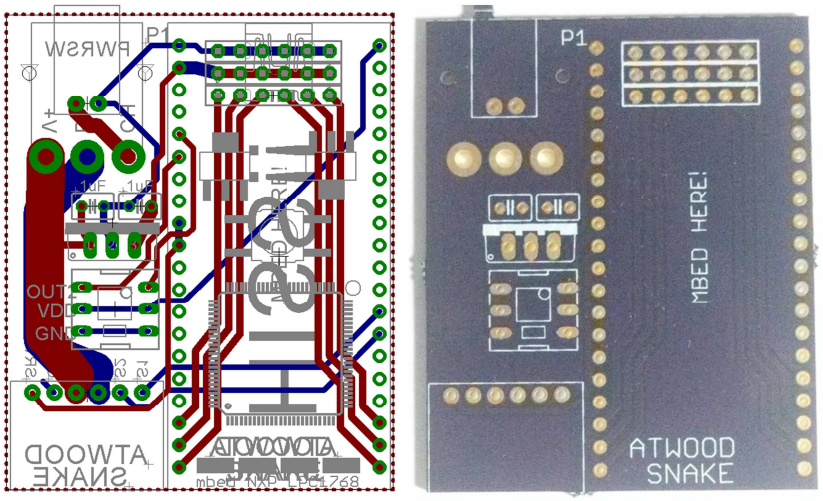

Figure 2: MBED Tail Board PCB Design and Blank PCB

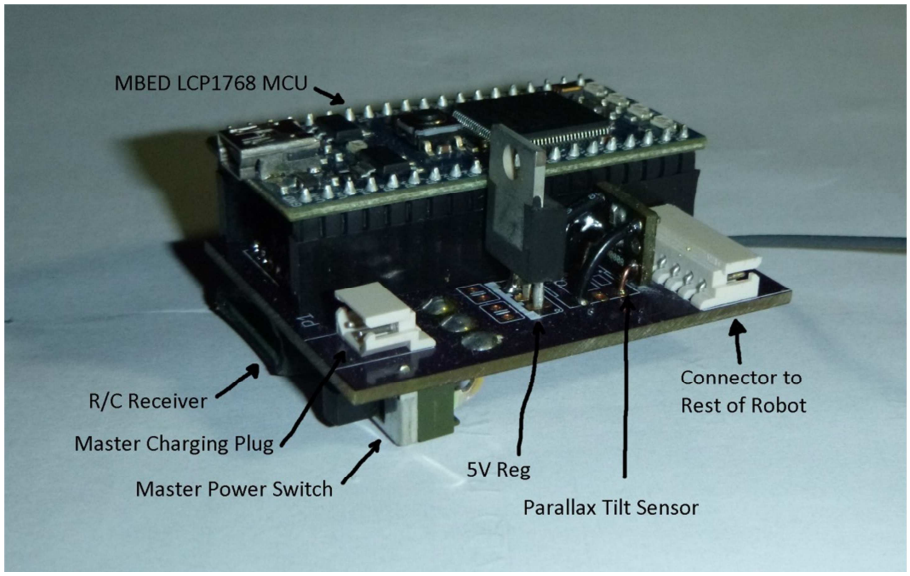

Figure 3: MBED Tail Board Populated PCB

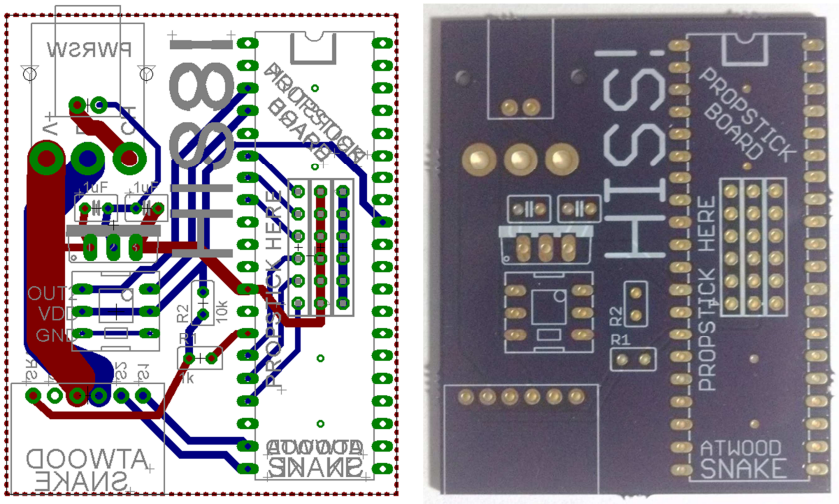

Figure 4: Propeller Tail Board PCB Design and Blank PCB

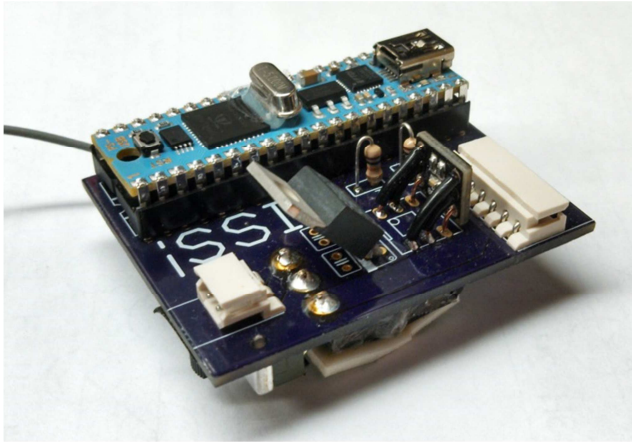

Figure 5: Propstick USB Tail Board Populated PCB

Cargo Segments

Each cargo segment has a small circuit board in it to facilitate the modular aspect of the robot. Only one wire connector needs to be disconnected to separate the robot at a specific joint. The cargo segment circuit boards also have plugs to interface with the battery packs installed in the segment, and an expansion header to allow a cargo segment to have a sensor package installed instead of a battery pack.

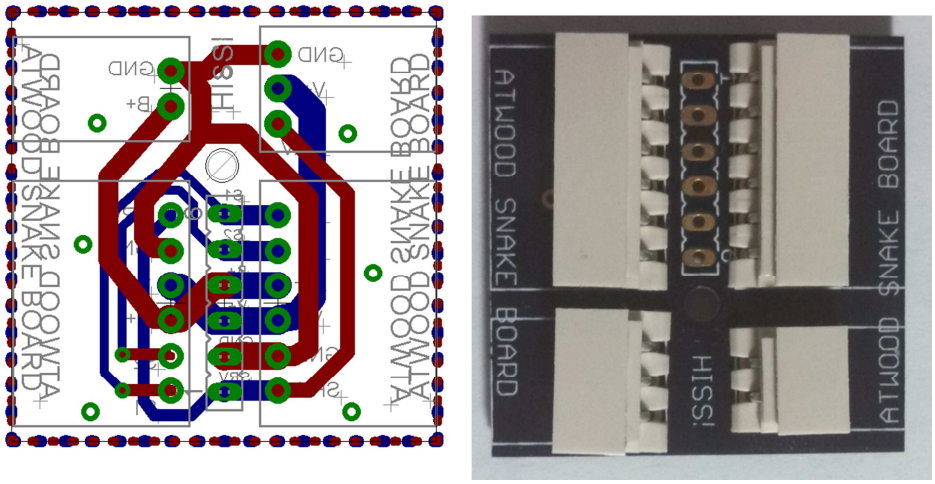

Figure 6: Cargo Segment Electrical Boards PCB Design and Assembled PCB

Head Segments

At the front is the best place for visually oriented sensors. The segment uses the same plastic piece as the tail segment, but contains a different electrical board. The Camera/Sensor Board has both a 5V and 8V regulator built in, but those could be switched out for other voltages as necessary depending on which sensors are desired.

One of the current modules has a PING))) ultrasonic sensor, which is useful for the robot to know where it is going when programmed autonomously. There is also a wireless camera module that sends live video to a receiver. I have hooked this up to a television to be able to view the robot live and also to my laptop to be able to capture the video. More software could be installed on the laptop to do advanced tasks like motion detection which would be useful if the robot was acting as a lookout or object tracking which could be useful while the robot is performing reconnaissance on a specific target.

There are several other sensors that could be installed in a head module to provide unique feedback. A color sensor could be useful for helping the robot identify colors especially when operating by itself. An I2C or serial camera could be installed and connected to the microcontroller to do onboard video processing. Small LIDAR, SONAR, or RADAR arrays could be installed to help the robot map its environment as it navigates.

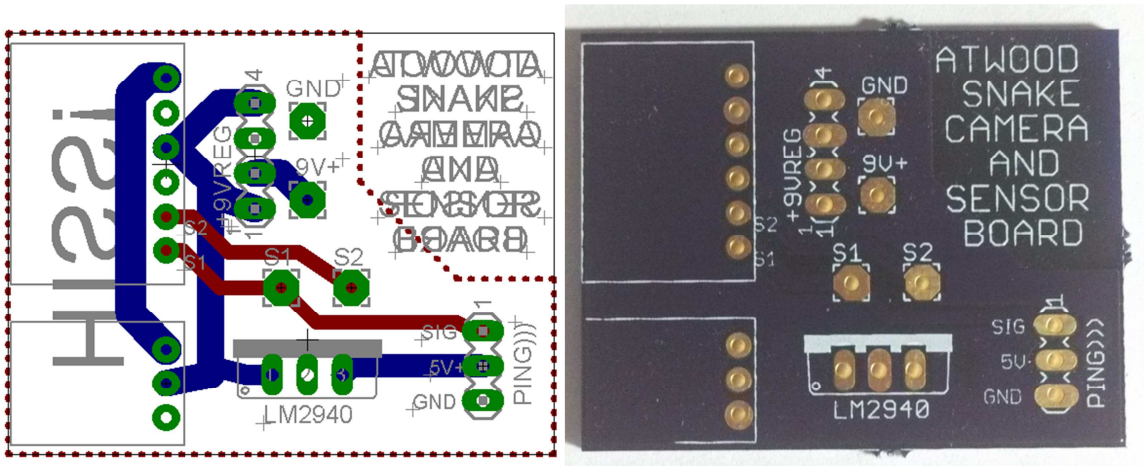

Figure 7: Camera/Sensor Board PCB Design and Blank PCB

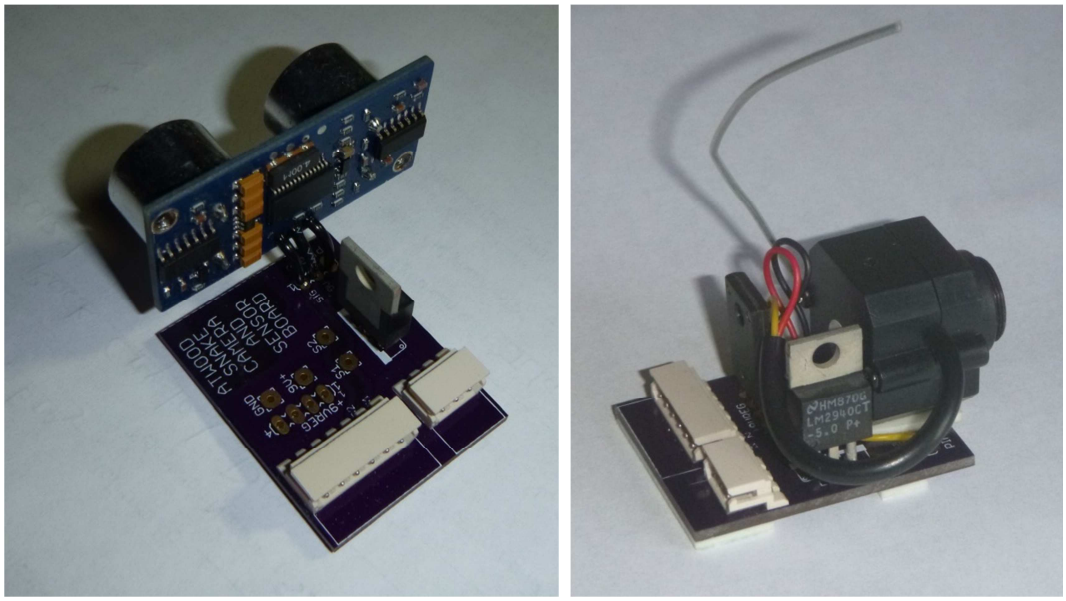

Figure 8: Camera/Sensor Board Assembled with PING))) and Wireless Camera, Respectively

Mechanical

There were several important considerations when designing the structure for the head, tail, and cargo segments. Most important was keeping the segments lightweight while robust enough to survive operation in harsh environments.

The cargo segments and the head/tail segments were constructed out of polycarbonate. From the CAD design, a flat drawing was laid out as reference for pieces to be machined to the right dimensions. The work for the polycarbonate was all done on a manual milling machine. Figure 9 shows the flat pattern for one cargo segment. The tabs on each end of the segment interlock snugly together for a strong mechanical fit. Solvent cement was used at the joint, but since the segment is held in compression the glue is not structurally necessary since the tabs take all of the loading.

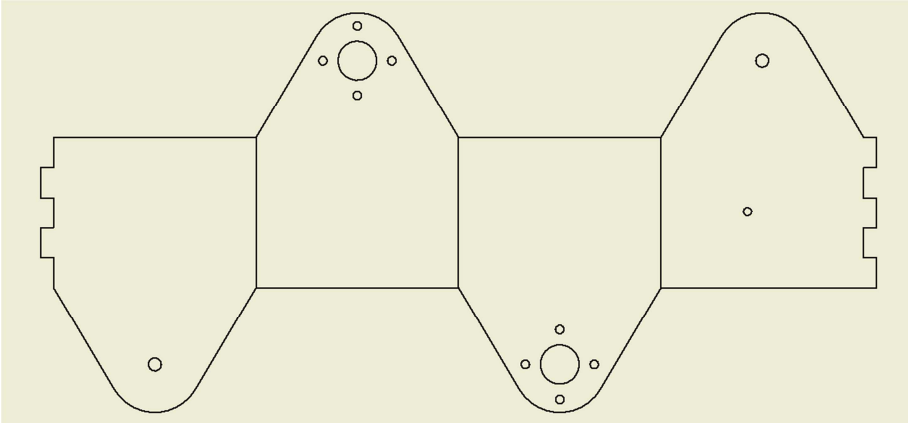

Figure 9: Flat Pattern for Cargo Segment Plastic

I built a hot wire bender as seen in Figure 10 with a piece of plywood and Nichrome wire to provide a linear heat source for heating the flat polycarbonate segments so they could be bent into the 3D finished shape. I also machined a bending jig to help guide the bends on each segment. Locating pegs were used as seen in Figure 11 to correctly align the bend relative to the motor mounting holes.

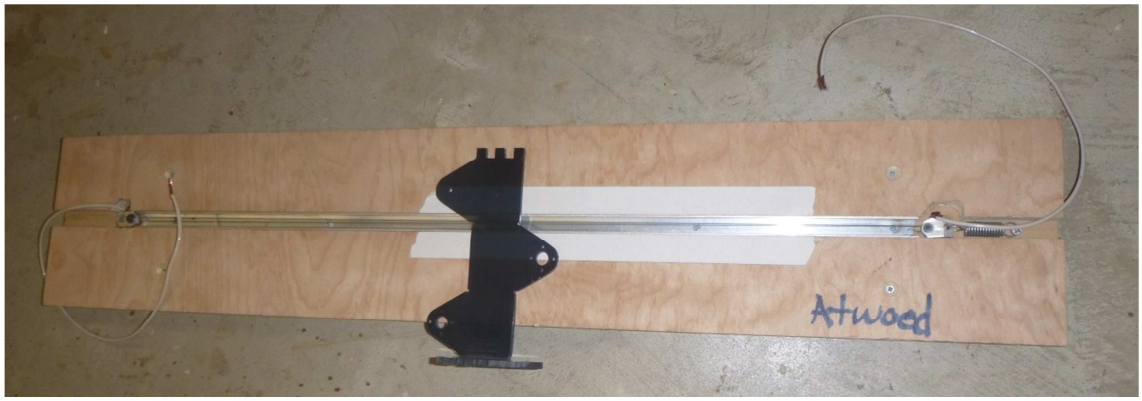

Figure 10: Nichrome Wire Powered Hot Wire Plastic Bender

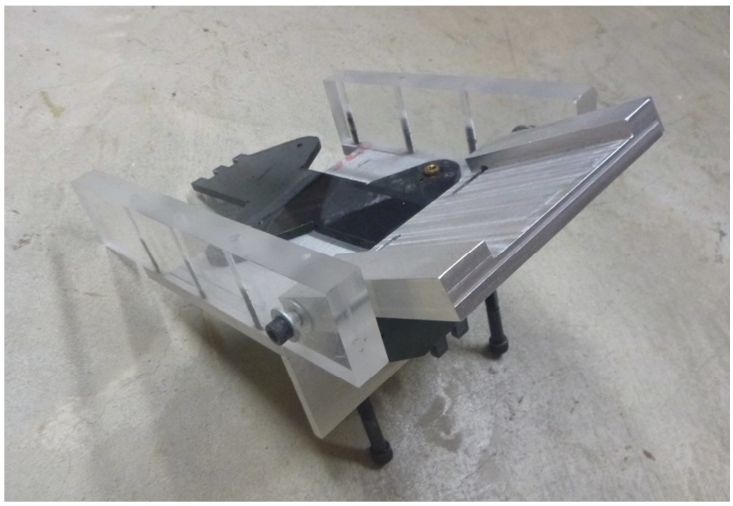

Figure 11: Segment Bending Jig

After the bending, spacers need to be added, two for the tail/head segments and four for the cargo segments. I glued little pieces of polycarbonate to the segments with solvent cement, but you could attach them the motors instead. Then the segment is wrapped with heat shrink tubing to hold the segment in compression. This holds the interlocking tabs together so the segment doesn't come apart. The glue put on the interlocking tabs earlier is not really necessary for the strength of each segment. The heat shrink tubing also acts to provide grip for the robot on smooth surfaces. Then stuff the electronics, batteries, and sensors into the segments as appropriate.

Figure 12: Completed Segments - Head, Cargo, and Tail

Motor Modules

Each of the motor segments consists of two motors held together by custom aluminum bars. The aluminum bars allow the motors to be easily changed out if one were to break, and keeps the weight of those segments low. The bars are shaped like angle stock, but at a custom size machined from slightly larger square bar on a manual milling machine. Figure 9 shows the design of the aluminum bars. Originally, the design was to put 4 bars on each motor pair, but initial testing showed that 2 bars were enough to hold the motors rigidly together.

Figure 13: Completed Motor Module

Current Programming

There are 3 main ways I have thought about the programming so far. The first is to have the robot execute a preprogrammed sequence of motions when the routine is called. This is used in functions like left() and right(), where values are already set based on experimental trials to achieve the result. The second way is to assign values directly based on an input to the robot. In routines like freeroll() and lookout(), PWM signals from the wireless controller are captured and the values scaled into usable angular values before being directly fed to the servos. The third way is to generate values on the fly from a set of equations. The routines wave() and slither() use method by capturing the current value from a system timer, and then a sine based equation is evaluated with the captured time value in order to generate the next servo position value.

Although I have done a little bit of programming with the Propeller based tail segment, nearly all my programming so far has been for the MBED based tail segment, so I have only included that code in the documentation. It is programmed in an online C environment which makes it easy to access from any computer. This is a list of routines that I have developed so far. A lot of the programming is trial and error to figure out what works and what does not work. The functions listed below are in the same order that they appear in the attached code documentation.

- tiltcheck() checks the Parallax tilt sensor on the tail board and returns a number indicating which side the robot is currently on. This is used so the robot can output signals to the correct motors depending on its orientation.
- CapRX(DigitalIn x) checks the pin passed to it in the function call and returns the PWM value on that pin. This is used to check one channel from the R/C receiver corresponding to the user input on the controller.
- straight() sets all the servos to their neutral position to make the robot straighten out.
- freeroll() is one of the coolest functions. It sets the position of each servo in one axis of the snake proportionally to the position of one of the joysticks of the R/C remote control. The program allows the user to make the robot move real time, and usually results in the robot looking like it is breakdancing. This is only allowed for 4, 6, and 8 motor snakes because the motors can get damaged at longer lengths due to too much of the robot slamming around.
- lookout() allows motors 1 & 2 to move the head of the robot around like a pan and tilt setup, while the remainder of the robot is wrapped around an object like a tree or a post.
- wave(int wavedir) creates a crawling motion in the vertical plane similar to a caterpillar or inchworm. The values for this are generated on the fly by passing the current time from a timer into a sine function, and adjusted for each motor through the length of the snake. The wavedir argument being passed to it should be 1 for forward crawl or -1 for backwards crawling.
- Rollright8() is a routine for only an 8 motor configuration currently, and allows the robot to lift its tail in the air and slam it sideways to force the robot to roll onto the next side. This is useful when the robot is operating autonomously and needs to switch sides, such as getting a camera right side up again.
- wiggle() is a routine that runs the robots servos through their range of motion. I use it all the time to make sure that all the motors are functioning properly, but it is also enjoyable to watch it scrunch itself up.

- left() and right() are used for 6 and 8 motor robots to turn the robot left and right respectively. One instance currently turns the robot around 20 degrees.
- slither() is the routine for snake-like slithering. The values for this are also generated on the fly, using two sine based equations. This currently provides very little forward motion in the snake, due to the fact that real snakes have anisotropic friction on their undersides and most researchers model this by including wheels on the bottom of their snakes. My snake does not have wheels but I wanted to demonstrate a snake-like slithering in this platform.
- main() is where the actual program to be executed is developed referencing the above list of subroutines, as well as others that could be developed in the future or brought in from outside sources.

The Propstick USB has the capability to multitask since it has multiple processors which would make it ideal for autonomous operations where the robot needs to multitask for best performance. It would be perfect for an application like inspection of a potentially hazardous pipe, where you could devote cogs to individual tasks like monitoring chemical sensors for hazardous gases, monitoring visual sensors in the front for obstacles and turns in the pipe, and coordinating movement of the motors.

Results

The segments are assembled together into configurations based on the testing goals and the programming is adjusted to reflect the length, sensors attached, and desired operating routines. Shown in Figure 14 are several possible configurations of the snake, each with different numbers of motors and different sensor packages. The code described above has been tested on 4, 6, 8, 10, and 14 motor snakes successfully while operating in manual control mode. Another program with just the main() program section changed has been tested on an 8 segment robot to autonomously crawl forward and take a different path if obstacles are spotted ahead.

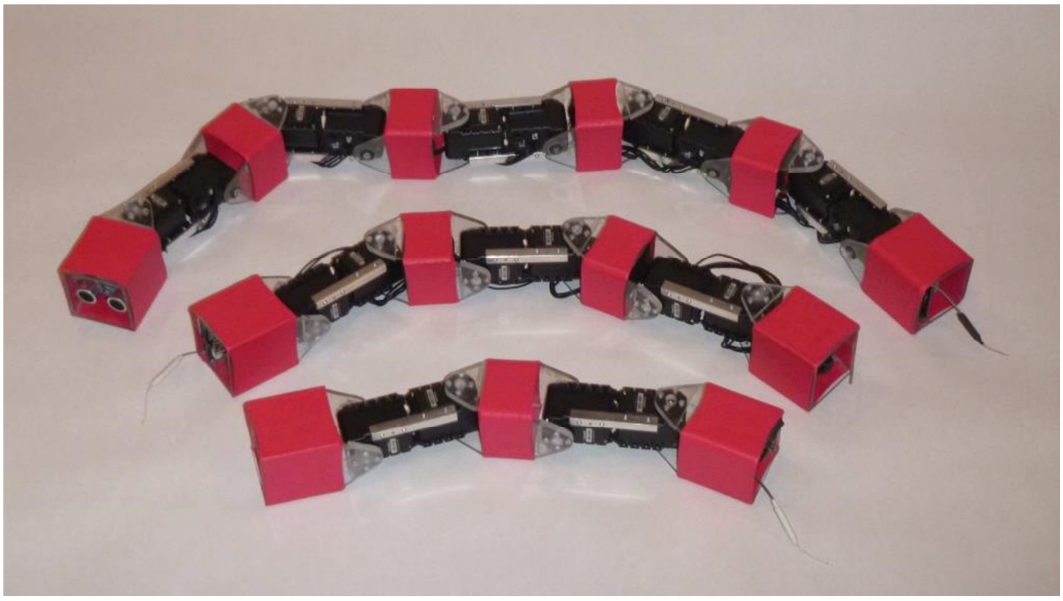

Figure 14: Several Assembled Configurations of the Snake Robot

The snake is able to grasp onto a smooth tree or a wooden post and not fall without any branches helping to hold it up. It can also be perched amongst branches to be able to use a larger portion of the snake to look around to inspect an area. The snake could be used like this to provide a lookout over an area during a mission to help a tactical team watch their escape route, help medics to find the casualties, or set up ahead of time to provide long term surveillance of an area.

Figure 15: SnakeRobot Holding onto a Tree, and Perched in a Tree

The snake was tested to be able to crawl through a 4” diameter PVC drainage pipe. This would be useful for inspection operations where soldiers wanted to verify that the pipes were clean of hazardous chemicals or within radiation limits. It could also be used during a tactical raid on a compound to get a visual on the inside by driving the snake through a hole in a wall, under a gate, or through an air vent.

Figure 16: Snake Crawling through Drainage Pipe

Summary

This project has demonstrated a snake robot that could be deployed for reconnaissance in the field by a medic or another soldier. Although this is just a first prototype, the robot already has amazing capability. It can be deployed in a stationary position, such as a tree trunk or post, to provide a lookout in a location where leaving a soldier to hold watch would be dangerous. It can be used as a reconnaissance robot to scout an area before medics move in, or even before a tactical team moves in. The robot can crawl through pipes and other small entrances to look for hazardous materials with a camera, or sniff for chemicals with a chemical sensor package. The robot can be commanded via remote control, or can run autonomously to take care of a task and then report back. Perhaps most importantly, the robot can be customized before deployment in many ways including changing the length, adding more sensor packages, and reprogramming the software to create custom programs based on the upcoming mission.

A thorough bill of materials for each type of segment is below. A 6 motor configuration for the robot consists of 1 tail, 1 head, 2 cargo, and 3 motor segments and costs around $430. A 10 motor configuration of the robot costs around $630 as built. Additional costs are battery chargers and a television or laptop to view the video feed. There are some aspects of the project that could be refined, but as a prototype this project definitely shows that a modular biologically inspired snake robot could be a useful tool for soldiers and medics on the battlefield in a wide variety of missions.

Electrical schematics, mechanical design drawings, and commented code are available in attached documents. Demonstrations in this youtube video: http://youtu.be/aIagGPOaIo4

Modular Robotic Snake Detailed Bill of Materials by Segment Type					
Item Purchased	**Use in Robot**	**Source Information**	**Cost When Purchased**	**Quantity Needed Per Segment**	**Cost Per Segment**
Motor Modules					
Dynamixel AX-12A Servos	Robot Motors	Trossen Robotics RO-902-0010-001	$225 per 6	2 Motors	$75.00
60mm 3 wire cable	Motor Connection Cables	Trossen Robotics CBL-BIO60	$10.90 per 10	2 Cables	$2.18
3/8"x3/8" Aluminum Bar	Motor Mounting Bars	Mcmaster 9008K21	$8 per 6 feet	2, 2.5 inch bars	$0.50
M2x8mm Screws	Mounting Motors to Bars	Mcmaster 92832A111	$5.96 per 100	8	$0.48
M2 Nuts	Mounting Motors to Bars	Mcmaster 90591A111	$1.39 per 100	8	$0.11
M2x10mm Flat Head	Attaching to next segment	Mcmaster 91420A005	$3.20 per 100	8	$0.26
M3x16mm Flat Head	Attaching to next segment	Mcmaster 92125A134	$7.33 per 100	2	$0.15
Tail Segments					
MBED LPC1768 OR	Microcontroller Option	mbed LPC1768	$50	One or the other	$50.00
Parallax Propstick USB	Microcontroller Option	Parallax 32210	$50		
MBED or Propeller Based Board	Tail Electronics	Custom	$7 per board	1	$6.20
Tilt Sensor	Robot Orientation	Parallax 28036	$10	1	$10.00
Power Switch		Mouser 633-MS13ASW30-RO	$4.72 each	1	$4.72
6 Pin SPOX Board Connector	Main Robot Connectors	Mouser 538-22-05-7065	$0.71 each	1	$0.71
2 Pin SPOX Board Connector	Charging Plug	Mouser 538-22-05-7025	$0.44 each	1	$0.44
R/C Receiver and Transmitter	Wireless Control if Desired	Hobbyking HK-6DF-M1	$27 per set	1	$27.00
Cargo Segments					
Cargo Segment Electronics Board		Custom	$3 per board	1	$3.00
Li-Ion Round Cell 650mAh	6 cells per battery pack	Hobbyking 9210000031	$1.96 per cell	6	$11.75
6 Pin SPOX Board Connector	Main Robot Connectors	Mouser 538-22-05-7065	$0.71 each	2	$1.42
2 Pin SPOX Board Connector	Battery Plug	Mouser 538-22-05-7025	$0.44 each	1	$0.44
3 Pin SPOX Board Connector	Local Servo Motor Hookup	Mouser 538-22-05-7035	$0.51 each	1	$0.51
Head Segments					
Parallax Ping)))	Sensors	Parallax 28015	$30	1	$30.00
Wireless Camera	Sensors	Amazon	$20	1	$20.00
Sensor/Camera Board		Custom	$4 per board	1	$4.00
Voltage Regulators		Depends on what voltage you need	$1	1	$1.00
Other					
Red 3" Heat Shrink Tubing	Cargo, Head, Tail Segments	Mouser 5174-13004	$22 per 4 foot	2 inches	$1.00
1/8" Polycarbonate Sheet	Plastic for Segments	Bought Local	$50 for 4'x4'		$2.00
Segment To Segment Cables					
6 Pin SPOX Connector	Main Robot Connectors	Mouser 538-50-37-5063	$0.39 each	2	$0.78
SPOX Crimp Pins	Main Robot Connectors	Mouser 538-08-70-1040	$0.26 each	12	$3.12
Wire, 18-22 gauge				6, 200mm pieces	