Sine-Wave Locomotion in a Robotic Snake Model Form and Programming

Mark W. Sherman

Tennyson High School
20375 Whitman Street
Hayward, CA 94544
arctangent@hotmail.com

Abstract

Mathematical phenomena displayed by biological life can be useful considerations when trying to reproduce aesthetically pleasing patterns of locomotion. Specific to this project was the study, design and construction of a robotic snake while trying to preserve the mechanism of movement that allows the biological snake to move along a selected path. Mathematical expressions for the serpentine movement of a living snake clarified and formed the basis for the translation from biological to robot. With the addition of sensory abilities, the robot can interact with its environment, and finally a user interface allows interaction with the robot.

1 Introduction

The purpose of this project was to mimic the locomotive movements achieved by biological snakes. Snake locomotion is very intriguing because it does not require the use of limbs such as legs or arms to move. Although many robots use wheels for locomotion, they usually become caught on obstacles or find areas inaccessible due to their size. To eliminate the use of wheels that limit the terrain able to be covered, some robots have attempted to mimic the walking motion of animals. Walking robots can range from very simple, to very complex, but often have trouble with balance or obstacles, and their movements are not usually smooth. The wave-like motion of a snake is promising for robotic development because it allows smoother locomotion. The beneficial aspect of such a robot could include carrying hazardous materials, inspection in uncertain terrain such as the debris of a downed building after an earthquake, or assisting law enforcement personnel in surveillance maneuvers.

This paper will cover the concept, design, construction and programming considerations which governed the successful execution of this project. Included are appropriate schematic diagrams and examples of programming, a discussion of the mathematical principles which generated the sine wave locomotion, as well as specific construction techniques developed to complete this project.

2 Related Work

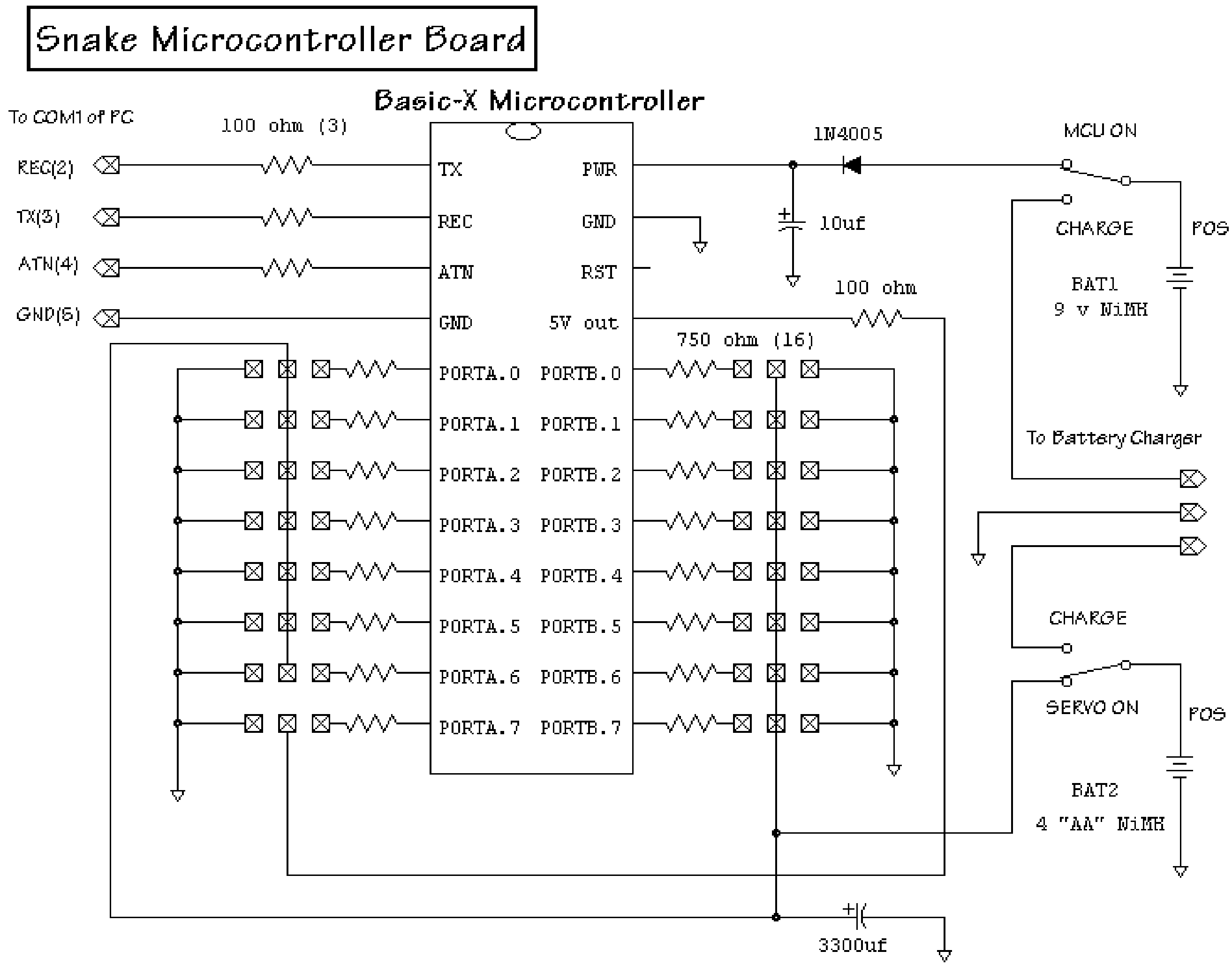
As part of the research phase, an understanding into the mechanism of movement utilized by a biological snake was imperative. It is quickly obvious upon studying biological snakes that they have a series of scales along their bellies, called scutes, which perform the function of side-to-side resistance (traction) that helps the snake pull its body along. In fact, research into experiments conducted by biologists found that without this form of traction, the serpentine movement of the snake does not take place.[1] For instance, the bad traction obtained by a snake on loose sand, will cause a side-winding trajectory rather than a forward path. For this reason, not all snakes utilize the lateral undulation method of locomotion, but rather, there are four patterns of locomotion employed by biological snakes.[2] For the purposes of this project, the mathematical model is the sine wave, and so locomotion by lateral undulation is the most appropriate description of the project goal.

The fact that biological snakes rely on traction on the ground to be able to move forward forms the basis for the ideals which governed the design and component placement on the robot.[3] There is also a growing group of biological scientists who see robotic "specimens" as a way to test hypotheses in the laboratory without harm to a natural environment or wildlife.[4] Robotic medical instruments in the form of robotic snakes are already under development to aid doctors in the collection of data, images and tissue samples from patients' bodies with increasing precision.[5/6] Robotic snakes have numerous applications within several fields of study.

3 Mechanical Design

3.1 Purposeful Design

Since one of the goals of this project was to make the robotic snake in the likeness of a biological snake, it was decided that the "brain" should be at the head end of the robot. Further, a definite underbelly and tail section are also present. The brain connects to the servos, the



Schematic Diagram: Snake Microcontroller

muscles of the robot, through a spinal column of appropriately sized ribbon cable passed through a 1/2" hole drilled into each segment support. This approximates the neural distribution or signal delivery to each servo (muscle) required for the sine wave technique.

4 Construction

There are several important issues regarding the construction of the robot including cost versus available budget for the project, arriving at adequate construction methods without sacrificing performance, the impact upon performance after material selection and, finally, unique construction methods used in this particular robot.

4.1 Budget

The problems encountered trying to conduct research without adequate funding certainly plagued this project. This, in fact, caused potential shelving of the project at several points. In the end, a balance between function and practicality surfaced. While it is too early to determine that this had a significant impact upon the results of this research, it can be determined that considerable robotics research can be conducted despite fiscal obstacles.

Through the careful evaluation of a component's desirability when considered within the given the budget constraints, components were obtained from scrap yards, surplus electronic supply outlets, company discards and sample components obtained at trade shows. Absent a benefactor, this was the only way to continue this project. The possibility of negating progress due to budgetary constraints was a constant problem. Limits placed upon functionality became possibilities that needed to be addressed so that the integrity of the effort was not jeopardized.

4.2 Design

4.2.1 Biological vs. Mechanical

Upon researching biological snakes, segmentation of the body seemed the natural mode of representation. In order to achieve the multiple angles required for locomotion, the robotic snake needed to have segments articulated by servos. The segments are a mechanical "rib" system, and the hobby-grade servos operate as the "muscle" of the snake. A single degree of freedom for each segment is represented in this project with further development planned to include increasing degrees of freedom.

4.2.2 Weight and Tension Supports

Each aluminum section of the body support was logically made identical to the others in size and shape. This ensured proper fit with each servo as well as maximized the possibility that if one segment appropriately touched the floor, all others should, in theory, do the same. The materials chosen had to be lightweight enough to conserve energy, yet sturdy enough to support both the servo mounted on it, and still withstand the strains of other segments' movements. The "spine" of the snake needed to be robust enough to hold its own weight, accept opposing influence from other segments, and do so while in continuous motion.

An "S" curve was first fashioned out of stiff cardboard and later adopted as the design shape of the segment supports. Using 1/16" sheet aluminum scrap, ten identical segments were fashioned in this "S" form. (See **Diagram 1** below.) Half-inch holes were punched into one slope end of the "S" shape in order for the necessary wires to pass through. This, in effect, constituted a "spinal column" that connected the servos to the "brain." The "S" design allows each servo to securely attach to the segment in front and behind without causing interference with the degree of turn obtainable from each servo. Each of nine segments has a servo attached via the "control horn" provided by the manufacturer, while the head end carries the "brain" in the form of a circuit board.

4.2.3 Traction

Although ideally the robot should eventually maneuver without wheels, its current design utilizes wheels as traction devices in the locomotive process. Wheels that can only roll forward and back while providing side-to-side resistance (traction) appear on the underside of each segment. The position was determined by estimating the center of gravity on each segment in an effort to maximize the traction mentioned above. Surplus Lego® wheels were chosen, for their low cost, availability and their consistent quality. It was possible to give each segment a wheel assembly identical in function as all other segments, an important consideration in the sine wave technique which works best when all pivot points are equal.

4.2.4 Power

Four "AA" Nickel Metal Hydride (NiMH) batteries were distributed on four sections to supply power to the servos. These were placed on the front half of the snake. Ribbon wire was strung through section to connect the servos to the microcontroller brain.

Separate power for the microcontroller was provided by a 9 V NiMH battery. The Basic-X controller has an on-board 5 V regulator which made using a 9 V battery practical and economical, since it was a regular stock item.

The separation of power sources for the servos and the microcontroller prevents the microcontroller from resetting when the servos pulse on. Testing revealed that having shared power source caused a voltage drop high enough to reset the microprocessor. Further, using separate switches to control the power, allows programming to occur on the Basic-X without the servos running.

Each servo takes about 120 mA average current when on, and about 1 Amp when all nine servos are running. The "AA" NiMH cells have an approximate capacity of 1600 mA/hr. This translates into roughly 1 1/2 hours run time.

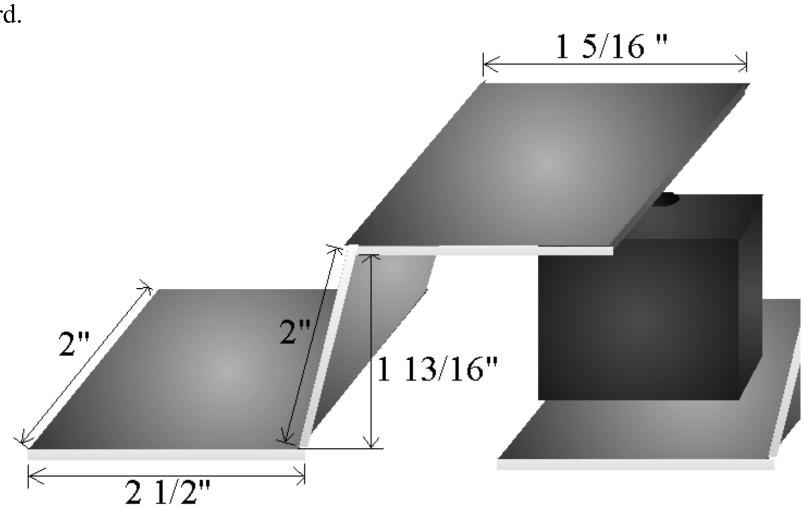


Diagram 1: Snake Segment

4.2.5 Microcontroller Assembly

The Basic-X module was mounted in a 24 pin socket on a Radio Shack #276-150 perf board trimmed to the dimensions: 2 3/8" x 1 7/8", then attached to the head end of the snake. Power switches for servos and the Basic-X were mounted on the same segment.

Access to all 16 ports of the Basic-X was achieved using female headers on the board. Each port made available power and common so each servo could directly connect to the Basic-X. A 750 ohm series resistor on each port connection to the Basic-X provided protection against any direct shorts due to wiring errors. A 3300 uF capacitor was connected across the servo power supply to help smooth out surges caused by the servos powering on.

Seven of the sixteen ports provide 5 volts (from the Basic-X regulator) and common connection to external sensors, switches, or other devices. The remaining nine ports supplied separate power to the servos from the four NiMH "AA" cells.

Connection to the computer's serial com port through a male header on the circuit board made quick and easy changing of the program without the necessity of removing the microprocessor from the robot. When power switches are off, the batteries can be checked and charged easily through a cable connecting to a male header mounted on the control circuit board.

5 Theory of Operation

During locomotion, the servos move in the pattern close to a sine wave. The underside of the robot has wheels, which have little friction forward and back, but much more friction from left to right. As the segments move in a sine pattern, the robot is forced to move forward, since each segment can only move forward or backward. The wheels resist side to side motion. Unlike a biological snake which cannot slither in reverse, our robot achieves backwards locomotion by reversing the direction of the wave. Turning left or right is accomplished by adding a curve to the sine, producing a sine wave that curves in one direction.

5.1 Sine Wave Controls Servo Angle

In the program, the subroutine which controls servo movement, takes an operand from 0 to 1 indicating servo angle. The servos have approximately 180° degrees of movement. To produce a smooth oscillating pattern for a single servo, the equation $y=(\sin(x)/4.0)+.5$ is used. As x is incremented, the servo angle will smoothly increase and decrease repeatedly.

5.2 Phase Shift

For the robot to move effectively, all the servos must move together to produce a pattern. If an image of the snake were taken as it was moving, the snake would have a sine-looking pattern to it. This is accomplished by adding a phase shift to each servo segment. This results in each servo behind the head moving 0.8 radians out of phase with the servo in front of it. This is accomplished by the following code:

```
For segment = 1 to 9
y= ( sin (x - 0.8 * csng(segment)  )     /4.0 + 0.5
MoveServo segment + 11, y  + toff*csng( 10-segment)
Next
```

The division by 4 scales the amplitude of the sine down to 25%, and the addition of 0.5 makes 0.5 the center location instead of 0. The MoveServo subroutine takes the pin# of the microprocessor the the servo is connected to as the first argument, and then angle as the second. The y value calculated before is added to an offset (toff) value multiplied by the segment number. This allows the robot to turn.

5.3 Making a Turn

If x were graphed against the turning offset (toff* 10-segment), it would be a line. How does adding a line make the robot curve in one direction? It does because these y values are not the y values you would expect on a cartesian x-y plane. The y values are actually angles. So if toff was .05, the first servo turns an angle of 0.05, the second turn 0.10, and the third 0.15. Assume the head of the snake were at a fixed angle and position. The actual angles positions observed for each segment would be the sum of that segment and all the servo angles for the segments before it. So, the first segment would have an angle of 0.05, the second would have (0.05+0.10)=0.15 , and the third would have (0.05+0.10+0.15)=.30. The curve's angles are added to the sine angles calculated earlier. When the snake is turning, toff is either positive and negative. This changes the sine wave pattern of the robot to a curved sine.

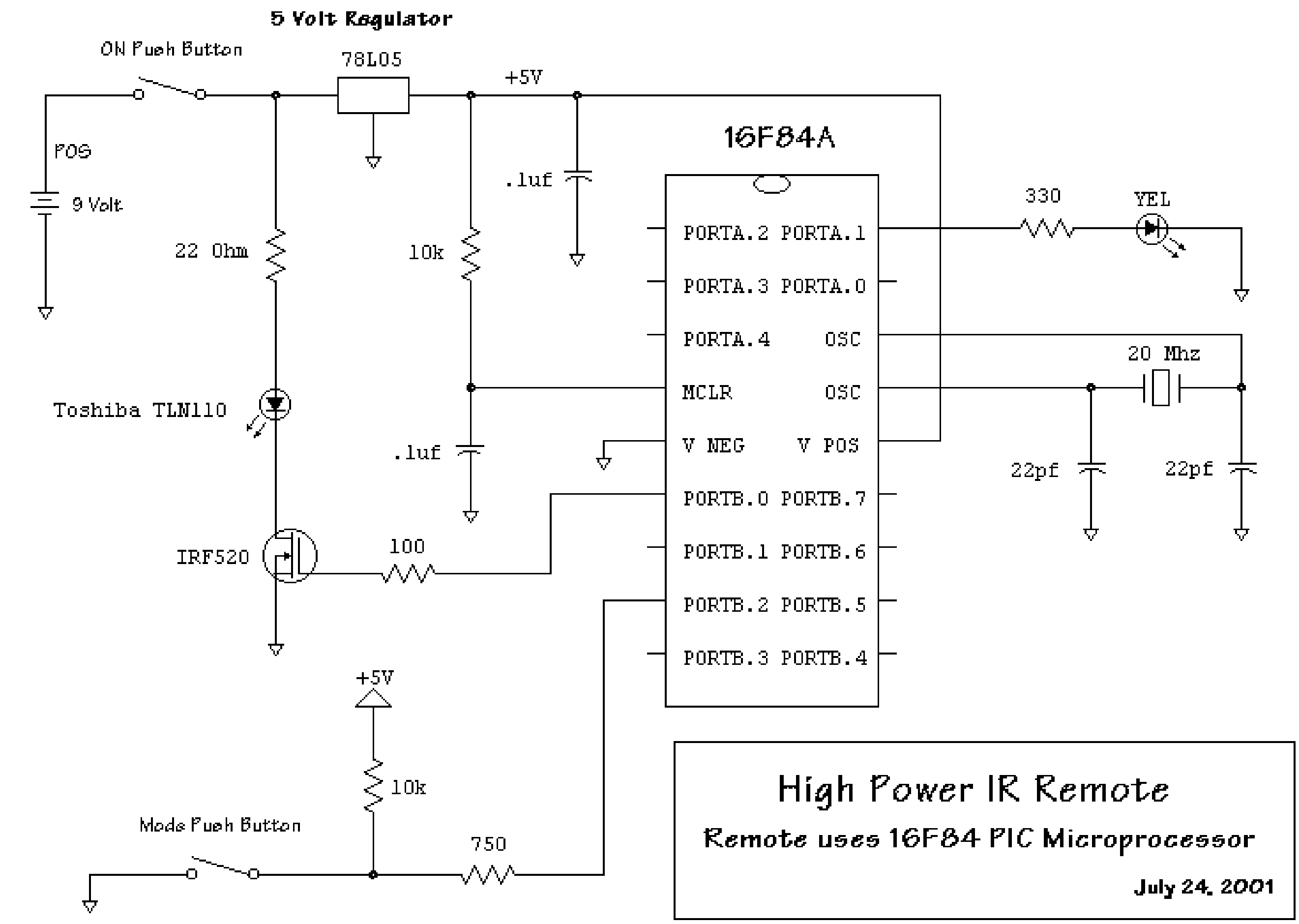
5.4 Program

Below is the program used to achieve sine-wave locomotion with the snake model described:

```
'Robotic Snake Model Controller Program for Basic X
'Written in Visual Basic
'Mark W. Sherman

  dim mode as byte   'snake mode 0=stop 1=forward
                     '2=left 3=right 4=backward
  dim xinc as single  'x increment
  dim toff as single  'turning offset (for turns)
dim speed as single 'snake speed
Sub Main()
  dim segment as byte 'snake segment counter
  dim x as single     'x coordinate independent
                        'variable
  dim y as single     'y coordinate dependant variable
  dim countdown as byte  'time delay counter
       speed=0.3    'set a medium speed
```

```
'Flash colored lights
putpin 25,0
delay 0.5
putpin 25,1
putpin 26,0
delay 0.5
putpin 25 ,1
delay 0.5     ' 1/2 second delay
x=0.0         'set wave to initial position
toff=0.00     ' range :-.02 ro +.02: 0.0 is straight, - is
              'right, + is left
xinc=0.0
mode=0  'start robot in a stopped state
countdown=0  'no countdown
do
  countdown=countdown-1  'deincrement countdown
  if (countdown<0) then
    countdown = 0
  end if
  if countdown=1 then
    mode=1        'set mode to forward
    call updatemode()   'update movement variables
 end if
  if countdown=15 then  'when there are 15 cycles left
                        'in the countdown ...
    xinc=-xinc  '...reverse the direction of the robot's
                'motion
 end if
 if getpin(10)=0 then  'if the right nump sensor is
  'touched
    mode= 2  'left countdown =30  '30  cycles moving to
             'the left
  call updatemode()
  xinc=-xinc     'reverse robots direction
end if
if getpin(9)=0 then   'if the left touch sensor is touched
  mode= 3     'right
  countdown =30 '30 cycles moving in the reverse dir
  call updatemode()
  xinc=-xinc    'reverse robot's direction
end if
if getpin(5)=0 then   'if the speed select key is pressed
  speed=speed+0.1  'increment robot's speed
  if speed > 0.6 then   'and wrap around if too high
    speed= 0.1
  end if
  delay 0.25
  call updatemode()
end if
  if getpin(11) = 0 then   'if in infrared signal is pressed
    putpin 25,0     'blink red & green lights to indicate
                ' reception
    putpin 26,0
    mode=mode+1     'increment to the next mode
    if mode > 4 then 'wrap around if necessary
      mode=0
    end if
    call updatemode()
```

```
      delay 0.25
      putpin 25 ,1    'turn red light off
    end if
    for segment = 1 to 9     'for each of the nine segments
      y= ( sin (x - 0.8 * csng(segment)  )      /4.0 ) + 0.5
      MoveServo segment + 11, y  + toff*csng(10_ ---------
      segment )
      'move to servo to that angle + and turning offset
    next
    Delay 0.020  'delay 1/50 second
    x=x+xinc     ' range :-0.4 to +0.4: 0.0 is stop, + is
                 'forward, - is backwards
                  'make sure that  -2pi<x<2pi
    if x > 6.28 then
      x=x-6.28
    end if
    if x < -6.28 then
      x=x+6.28
    end if
  loop
End Sub

Sub updatemode()
  if mode=0 then  'stop
    xinc=0.0
    toff=0.0
  end if
  if mode=1 then  'forward
    xinc=speed
    toff=0.0
  end if
  if mode=2 then 'left
    toff=0.04
    xinc=speed
  end if
  if mode=3 then 'right
    toff=-0.04
    xinc=speed
  end if
  if mode=4 then 'backward
    toff=0.0
    xinc=-speed
  end if
end sub

Public Sub MoveServo( ByVal ServoPin As Byte,  _
ByVal Position As Single)  'Basic X Servo Routine
 ' Moves a servo by sending a single pulse. The position is
 'a nondimensional value in range 0.0 to 1.0.
 Dim PulseWidth As Single
 'Translate position to pulse width. Resulting range is 1.0
 'to 2.0 ms,
 ' centered at 1.5 ms.
  PulseWidth = 0.001 + (0.001 * Position)
  'Generate a high-going pulse on the servo pin.
  Call PulseOut(ServoPin, PulseWidth, 1)
End Sub
```



Schematic 2: Infrared Remote Control (IRC)

6 User Interface

6.1 Infrared Remote Control (IRC)

The robot is controlled more directly by the user through the use of the project-designed IRC. The infrared signals are picked up by the onboard infrared receiver contained on the main "brain" circuit board. If the robot receives a signal from the remote control, it will cycle through various modes of operation with each press: forward, left, right, reverse, and frozen. The command hierarchy programmed into the robot uses stop as the default, and on subsequent signals, begins to cycle through the above list of simple commands. The whisker sensors have the highest priority. This position in the hierarchy will circumvent the remote control signals if the user tries to command the robot to run into an obstacle.

6.2 Necessary Interpretations/Assumptions

It is necessary to assume that the user, at this stage of project development, is an integral part of the experiment. Therefore, user interface is provided by the use of the IRC.

6.3 Construction

The IRC is housed inside a plastic project box obtained as a trade show sample. The box measures 4 1/8" x 2 3/8" x 13/16" and houses the circuit board and also a 9 V NiMH battery as a power source. Activation is via pushbutton control by the user. See **Schematic 2**, "High Power IR Remote" above.

6.4 Program

Following is the program used in the infrared transmitting device. While an ordinary 38 khz TV remote control works, part of this project was to design all components possible for the project.

```
    'Program written using Pic Basic Pro from
    'Micro Engineering Labs
define osc 20
trisa=%11111101
trisb=%11111110
n var byte
func var byte
```

```
push var portb.2
detect:             'determine mode
read 1,func         'read eeprom
if (func<>0) and (func <> 1) then func=1
write 1,func
endif
if push = 0 then func = func xor 1
write 1,func
endif
indicate:           'indicate mode to user
high porta.1        'ON indicator led goes high
if func = 0 then goto cont
if func = 1 then goto pulseme
pulseme:
for n= 1 to 250     'start burst of 38 kc
high portb.0        'led ON
pauseus 5           'led ON for 5 usec
low portb.0         'led OFF
pauseus 17
asm                 'fine tune delay
nop
nop
nop
nop
endasm
next n              'repeat to complete burst
delay:
low porta.1
pause 400           'delay and do another burst
goto indicate
cont:               'routine for continuous burst
for n= 1 to 250     'start burst
high portb.0 'led on
pauseus 5           'led ON for 5 more usec
low portb.0         'led OFF
pauseus 17          'off time for led
asm                 'fine tune delay
nop
nop
nop
nop
endasm
next n
goto cont           'continue burst
flip1: write 1,1    'write a 1 to address 1
return
end
```

7 Preliminary Results

Under remote control, the snake robot was able to move forward, reverse, and move to the left or right. After adding two touch sensors to the front of the robot, a little bit of autonomy was achieved. When the robot makes contact with an obstacle, it backs up and turns away from it. It then continues forward.

When the user sets a mode from remote control, (such as forward, turn, etc.) the robot will obey unless it receives touch sensor input. The user cannot force the robot to move into an obstacle.

Even with simple obstacle detection, the robot can often get in to trouble. If it gets trapped in a corner, it will repeatedly turn left and right until it is rescued. There is also no way to detect obstacles it touches as it backs up. In the future, use of infrared for obstacle detection and having the ability to sense obstacles all around are desirable additions.

8 Conclusions and Future Work

In this paper, we have pointed out that robotic research can be done within limited budgets and still achieve some degree of success. We have also discussed the advantages to using mathematical phenomena found in biological species to create an aesthetically pleasing locomotion pattern in a robotic machine.

Design changes will take place to incorporate additional degrees of freedom. This will enable the snake to maneuver over uneven terrain, which will bring this project one step closer to achieving one of the project goals: designing a robot that could assist in surveillance maneuvers over terrain with debris in the path. Plans to provide additional sensory capacity for heat, light and motion are under development, as well as the delivery to a remote location of audio/visual capacity so information can be sent back to the user remotely. Experiments into finding materials suitable to achieve the traction goals of the original wheels used, will be conducted to eliminate the use of the wheel as a traction device. There is also the evolution of intelligence to make the snake more independent of the user. Last, but always important, new ways to fund robotic research and encourage more high school and college students to prepare for robotics related occupations needs to receive significant attention.

References

[1]Docent 2001 Snake Facts: Snakes of Indiana Homepage http://herps.ipfw.edu/center/accounts/snakes/INSnakes.htm

[2]Vertebrate Natural History 2001: Vertebrate Locomotion http://muskingum.edu/~ingold/lectures/232lec10.htm

[3]Owen, Scott 1997. The Motion Dynamics of Snakes and Worms http://www.css.tayloru.edu/instrmat/graphics/hypgraph/animation/art_life/snakes.htm

[4]Current Opinion in Neurobiology 8:777-782 http://vorlon.cwru.edu/~beer/Papers/BioRobotics.pdf

[5]Chen 1996 Snake-like Robot Endoscopes http://robby.caltech.edu/~chen/res-medical.html

[6]Murray 2000. Geometric Phases, Control Theory, and Robotics. pp 33-34. National Academy Press. http://books.nap.edu/books/030905785X/html/34.html#pagetop