CHRISTINE MIYACHI CMIYACHI@MIT.EDU

INVESTIGATION OF GATES NITONAL ROBOTS

FINAL PROJECT 6.836

INTRODUCTION

The purpose of this project was to investigate various gates using a robot built with nitinol actuators[1]. The nitinol material was studied as well. Various methods of controlling the robot were used. This report will describe this project in the following sections:

- The robot construction and nitinol

  The robot built is inexpensive and requires no soldering. Nitinol is used for the actuators of the robot. The construction of the robot will be explained as well as the properties of nitinol that give it the nickname "muscle wire".

- Controllers

  Three types of controllers were used for the robot. The first one was a manual controller with a switch. The next was a simple circuit board with a timer chip. This controller also had a tactile sensor on it, which caused the robot to increase the speed of the gate when sensed. The third controller was a micro controller, which programs the robot gates. The robot measured the speed of the gates (with a light sensor, explained below) and picks the fastest gate.

- Analysis

  A brief analysis of the project will be presented.

ROBOT CONSTRUCTION AND NITINOL

The robots built for this project are called "Stiquito" and the materials and building instructions are available through the IEEE [1]. The robot requires no soldering but good dexterity to assemble. A picture of the robot is shown in Figure 1.

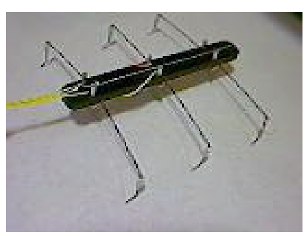


Figure 1- A Stiquito robot built for this project. This robot is wired for a tripod gate.

If you are interested in detailed instructions, please refer to the book [1]. The main points of the construction are listed here:

[1] In the process of building these robots, I experienced many of the messy real world problems discussed in class. Broken connectors were the biggest problem.

- Six legs are constructed with a stiff wire called music wire. Nitinol is attached to each of the legs.

- The Nitinol is pulled tightly to cause some tension in the music wire.

- The robot is wired so that when power is applied, current flows through the nitinol wire and causes it to contract.

Nitinol is a wire made of nickel and titanium and is called a shape-memory alloy. The wire used for this project is only 0.004" diameter. When electrically heated, the wire contracts like muscle. However, a counterforce is needed to bring the wire back into place. This is why the nitinol wire must be very tight against the music wire.

The nitinol wire contracts due to heating. The higher the current through the wire, the more heating, and the more contracting that occurs. Similarly, the wire relaxes due to cooling. For the wire I used in this project, 180mA would cause a contraction of 150g in about 1 second.

GATES AND WIRING

Four robots were built. Each one had a different gate as shown in Figure 2.

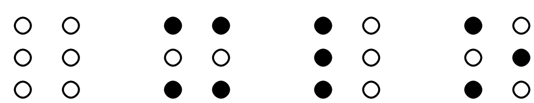


Figure 2 – Gates of the robots built. First on the right shows each leg individually controlled. This robot can be programmed to have all other gates. The second is a caterpillar gate, the third is a pace gate, and the fourth is a tripod gate.

The first three robots built were hard wired for a gate – caterpillar, tripod, and pace. The caterpillar and pace gated robots where built with manual controllers. Manual controllers are ones where a person has to touch switches to activate the gate. This was demonstrated in my presentation. The third robot uses a tripod gate and runs with a simple timer circuit to drive current to the robot legs. The last robot built wired all legs independently and is controlled by a micro controller. These controllers will be described in detail in the next section.

The fourth robot with the independent legs has been programmed to run through all the gates. The gates will be tested for speed, accuracy of walking along a line, and stability.

CONTROLLERS

Three controllers were used for this project. The simplest one is a manual controller, followed by an automatic controller, which is a simple timer circuit with a sensor. The most sophisticated is a micro controller that was used to program different gates on the robot and measure.

MANUAL CONTROLLER

The schematic of the manual controller is shown in Figure 3.

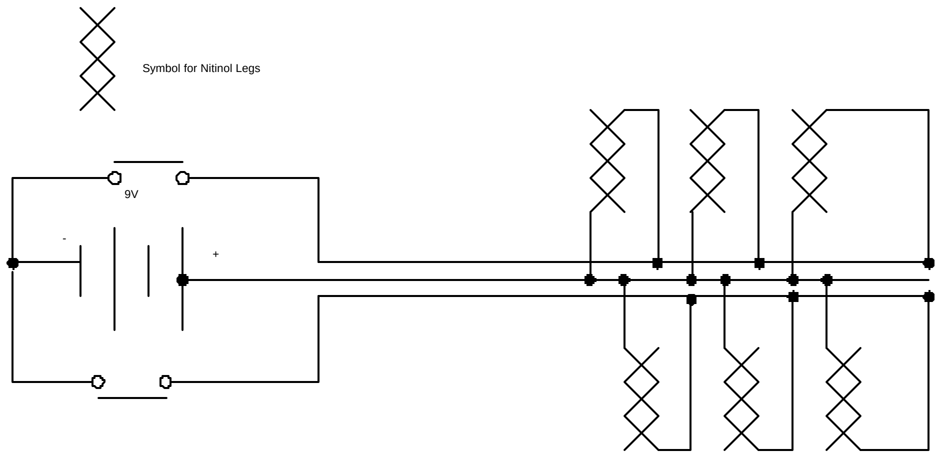


Figure 3 – The manual controller shown is wired for a pace gate.

The switches on the manual controller stay open until a human presses them closed. To move the robot, a user would push one switch on, activating three of the legs, and the turn it off and push the other switch down and activate the other set of legs.

The manual controller is used for testing and is only practical with the hard wired gates.

TIMER CONTROLLER

The circuit board for this controller was purchased through the IEEE (Institute of Electrical and Electronic Engineers http://www.ieee.org). I modified the design slightly to include a tactile sensor. The schematic is shown in Figure 4. The board is small enough to mount on top of the top of the robot as shown in the Figure 5.

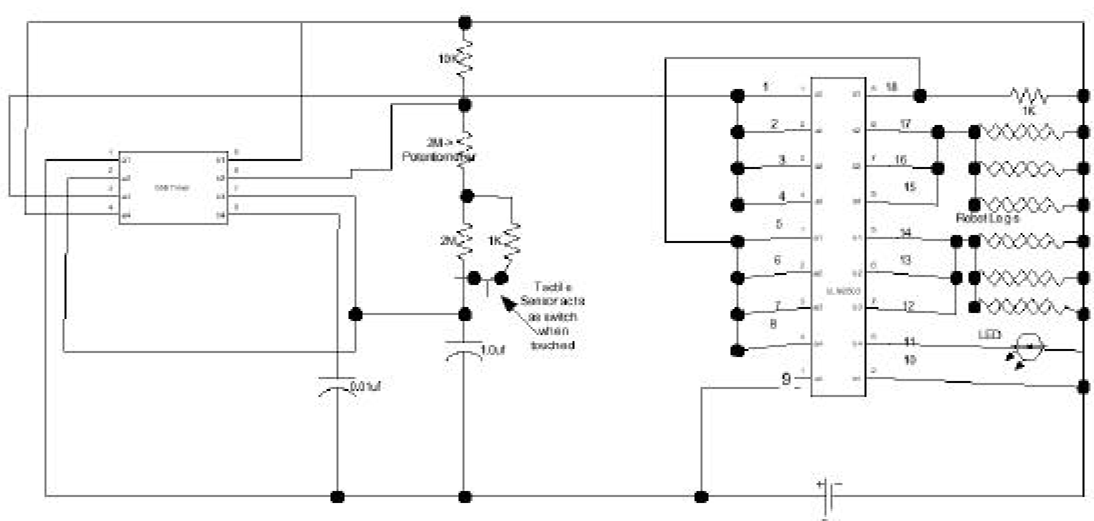


Figure 4 – Schematic for timer circuit to control the robot.

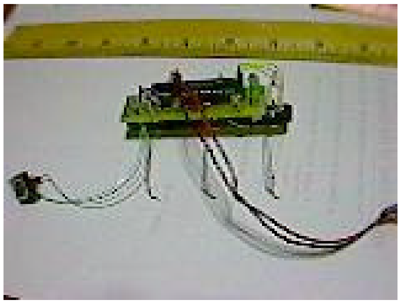


Figure 5 – Timer controller mounted on top of the robot. The tactile sensor is on the left and the lines to the battery are on the right.

The 555 Timer can be configure to generated pulses on its' own, which is how it was used here. As explained in the chip specification [2], when the input voltage on pin 6 is high and the output voltage on pin 3 is low, the 1.0 uf capacitor will charge until the voltage on pin 3 reaches 2/3 Vcc. At that time, the voltage of pin 3 reaches switches to 0, and the 555 chip will discharge the 1.0 uf capacitor. As the voltage at pin 6 drops to 1/3 Vcc, the cycle starts again. The timing diagram is shown in Figure 5.

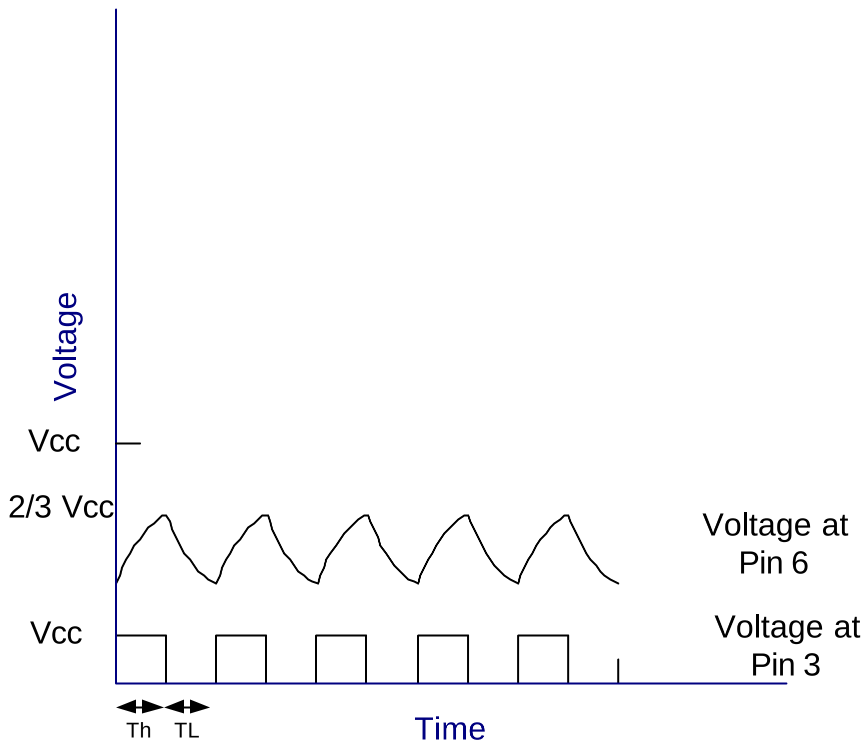


Figure 5 – The timing of the signal in the controller circuit

We can determine how fast the gate will be by looking at the following formulas.

Let

T_L = the width of the of the low pulse on pin 3

T_H = the width of the high pulse on pin 3

P = potentiometer value (~0.5 Mohm)

R1 = 2.0 Mohm

R2 = 0.01 Mohm = 10 Kohm

R3 = 1 Mohm

C1 = 1.0 uf

The formulas for the time then are:

T_L= (P + R1)*C1*ln(2) = (0.5 + 2.0)Mohms 1.0 uf*0.693 = 1.733 seconds

T_H= (P + R1+R2)*C1*ln(2) = (0.5 + 2.0 + 0.01) Mohms 1.0 uf *0.693 = 1.739 seconds

If the tactile sensor is set off, we can replace the value of R1 with (1/R1 + 1/R3) = 1.5 Mohms

T_L= 1.386 seconds

T_H= 1.4 seconds

So the robot will move ½ second faster for each pulse of the gate when the tactile sensor is fired.

CURRENT TO THE ROBOT LEGS

So as the voltage from pin 3 goes up and down, this voltage is sent to the ULN2803A chip [2], which is used to control the current to the robot legs. As seen in the circuit diagram in Figure 4, the voltage is the only input from the timer into this part of the circuit.

The video in Appendix C will show a robot with this circuit working.

MICRO CONTROLLER

To program and analyze the gates, a micro controller was attached to a robot with independently wired gates. Using this setup, various gates could be tested with one robot, which removed the variations if testing with robots.[2]

I discovered the Basic Stamp [3] micro controller on the robotics store web site (http://www.robotstore.com) and ordered one. The chip came embedded with a BASIC interpreter, RAM for user data, and EEPROM for the program, a 5-volt regulator, and lots of general-purpose I/O pins, and built in commands for math and I/O pin operations. It runs a few thousand instructions a second and it is easily programmed using the BASIC language[3]. I also used a small

[2] Even though these robots are simple, they are very small and difficult to build exactly the same. I built over five of them (one was ruined mid-stream) and my last one was significantly better than my first. The point is that there would have been a huge variation between two of the robots I built if I were going to test gates with a hardwired robot.

[3] I know, *real* programmers don't use BASIC.

breadboard to add some of my own circuitry to the design. A picture of the setup is shown in Figure 6.

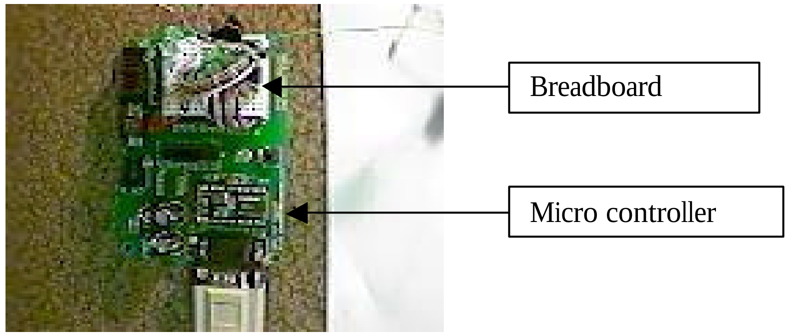


Figure 6 – Micro Controller with breadboard and ULN2803A, LEDs, wires leading to robot.

The package with the micro controller came with some software to be installed on a PC. With this software and a serial cable, you could edit BASIC programs and download them to the micro controller.

The circuit used with the micro controller is shown in Figure 7. Pins 0 – 5 of the micro controller are used to control each of the six legs. If the pin is set high, current flows through that leg. And similarly, when the pin is set low, the current is turned off. The ULN2803A chip is used to drive the current to the legs as it was in the timer circuit.

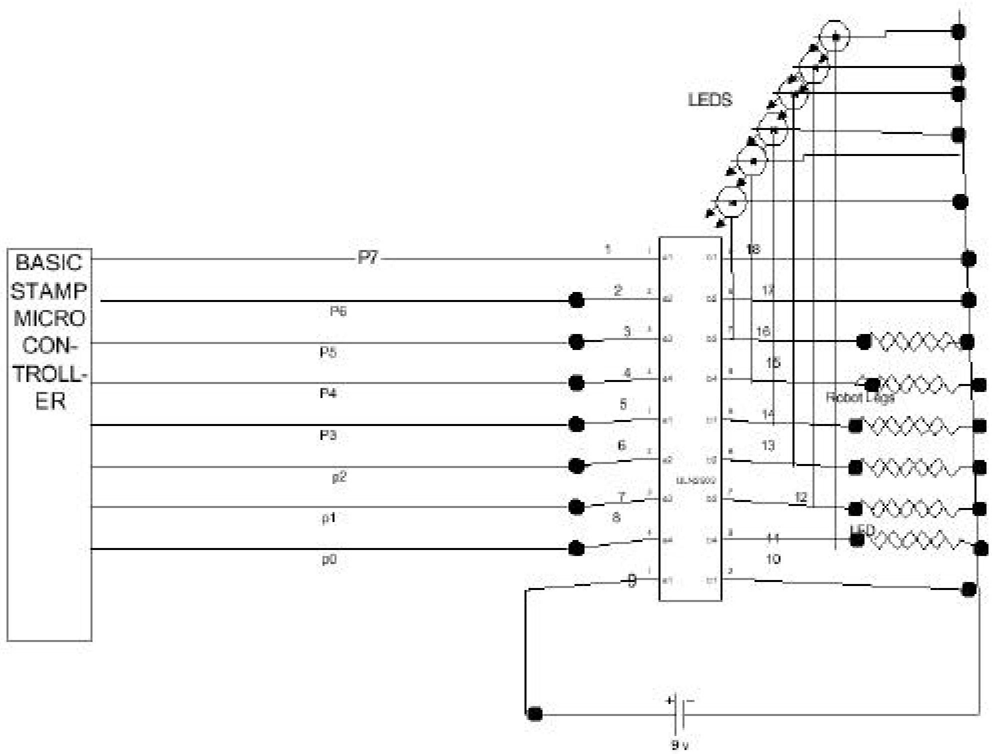


Figure 7 – The schematic of the micro controller. The ULN2803A chip is used again to drive the current to the robot legs. LEDs light up as the current is sent to the legs so we can observe the gate that the robot should be using by observing the LEDs. The program used to set the gates is listed in Appendix A.

CONTROLLING NITINOL

It was late one night and I was confused as I tested the robot. The circuit worked but the robot legs wouldn't move. I touched the nitinol wires to test their tautness against the music wire. They were taut but they were very hot. Tired and puzzled I went to bed. When I woke up the next morning, I realized that the legs were staying contracted because the heat was not able to dissipate after each pulse of current. The wire had not cooled off and the next pulse of current was sent, causing the nitinol wire to stay contracted.

To solve this problem I considered using both pulses of current of various widths and at various frequencies. These techniques are called pulse width modulation (PWM) and pulse frequency modulation (PFM) respectively. The robot nitinol legs only needed enough current to replace the heat lost. So the legs would need an initial burst of current and then less as time went on. This would replace the pulse of continuous current for ½ second in the program.

To test my ideas I wrote two programs. The concept is shown in Figure 8.

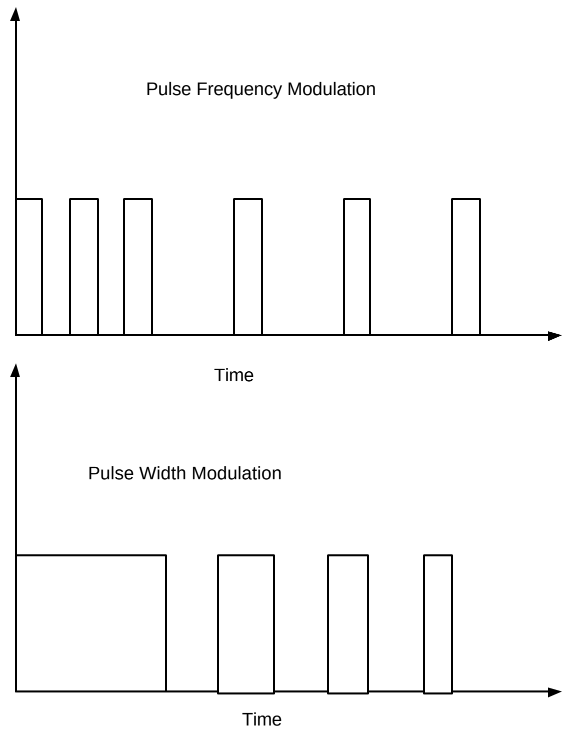


Figure 6 – The idea of PFM and PWM. PFM keeps the pulse width constant and varies the time between pulses and PWM keeps the time between pulses constant and varies the pulse width.

If the PFM program would control the leg effectively, this would be the one I would use, since it has the current on less time overall. I ran each program 100 times on the robot. The PWM would eventually have cause the legs to stay contracted with the PFM moved the whole time with out generating too much heat. Thus, for the experiment, I used the PFM technique. The basic program is shown in Appendix B.

ANALYSIS

Each gate was analyzed for straightness (walking along a straight line). Speed and stability were not factors to compare for this robot (see below). Since the gates were all demonstrated on one robot, the variation between robots was eliminated.

Although the legs of this robot are much simpler than insects, the gates of insects can still be demonstrated effectively. Insect gates are generated from a pattern generator in the central nervous system and vary from a caterpillar gate or a wave of leg motion, to the pace gate where each side of the legs move at once, to the tripod gate. Some gates are inherently instable, like the pace gate because the insect would be standing on one leg or all the legs on one side and could easily tip. This gate is useful for maneuvering on unstable ground or irregular ground. The caterpillar gait is the most stable since the most legs are on the ground at least for a portion of the gate. It is also the slowest gate studied. The tripod gate is the fastest gate studied and is stable with three legs on the ground at all times.

For this robot, speed was not compared because the speed of the gates was almost identical. This was probably because there were six legs. The three gates studied generated about the same amount of forward propulsion.

Stability was also not an issue with this robot since the legs did not move off the ground. The pace gate though is the least stable for insects though, since they have all legs on one side off the ground.

The method to measure the straightness of the gate was rather crude. A small piece of led was taped to the bottom of the robot. For each gate, a line of walk taken was drawn for four steps on each side.

Not surprisingly, the pace gate was the most zigzagged, followed by the tripod gate. The caterpillar gate was the most straight. Figure 9 mimics the results.

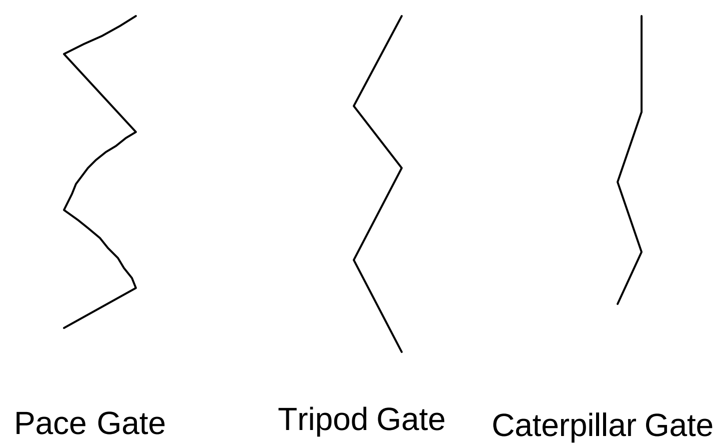


Figure 9 – Blow Up of the pencil tracings for each of the robot gates tested.

PROBLEMS WITH NITINOL

Nitinol is an unusual material. It is as thin as a strand of hair but very strong and hard. Yet when sanded and bent it can break easily. It also gets very hot when contracting.

If you have every observed a muscle fiber, it behaves like nitinol (with less heat generated). In humanoid robots, nitinol could be used to create striated muscle very similar to humans. Facial muscles are very subtle and are used for communication. There are not many actuators that can simulate this. However, given the heating problems of nitinol and the problems of controlling thousands of these wires, this idea may not be practical.

FUTURE IDEAS

- I had a line-tracking sensor that I was unable to assemble because I ran out of time. I will add this circuit to the micro controller system and see if I can get the robot to track a line.

- I would like to get some strain gages and program the micro controller to be a neural network or pattern generator for the gates. This would mimic more closely the sensor-motor loop of an animal.

CONCLUSION

This project investigated gates of nitinol robots. The tripod gate was found to be the best for walking in a straight line.

Two methods of controlling the robot were used: a simple timing circuit with a sensor for feedback and a micro controller. The micro controller was small enough to mount on the robot's back and would have had enough power to handle sensors and a wide variety of programs for the robot.

The nitinol material looks very much like muscle in its behavior but its heating properties make it difficult to work with. If the wires were bundled together to make a muscle fiber, the heat problem would become worse. There would be no place for the heat to escape and the muscle would stay contracted.

The PFM technique was one way to contract the wire with the least heat necessary.

Building a simulation of this robot would have not allowed me to experience the flames of nitinol and the mechanical difficulties of building a real world robot.

APPENDIX A – BASIC PROGRAM TO SET THE GATES

```
'{$STAMP BS2sx}
'  This program walks the robot through all the gates
'  Stability of gates will be checked manually
'
'   Author: Christine Miyachi
'   Final Project 6.836
'
' Tripod Gate
    Reps VAR BYTE
    FOR Reps = 1 TO 4
        HIGH 0
        HIGH 4
        HIGH 2
        PAUSE 1000 ' wait 1 sec
        LOW 0
        LOW 4
        LOW 2
        PAUSE 1000
        HIGH 3
        HIGH 1
        HIGH 5
        PAUSE 1000
        LOW 3
        LOW 1
        LOW 5
        PAUSE 1000
    NEXT
' Pace Gate
    FOR Reps = 1 TO 4
        HIGH 0
        HIGH 1
        HIGH 2
        PAUSE 1000 ' wait 1 sec
        LOW 0
        LOW 1
        LOW 2
        PAUSE 1000
        HIGH 3
        HIGH 4
        HIGH 5
        PAUSE 1000
        LOW 3
        LOW 4
        LOW 5
        PAUSE 1000
    NEXT
' Caterpiller Gate
    FOR Reps = 1 TO 4
        HIGH 0
        HIGH 3
        HIGH 2
        HIGH 5
        PAUSE 1000 ' wait 1 sec
        LOW 0
        LOW 3
        LOW 2
        LOW 5
        PAUSE 1000
        HIGH 1
        HIGH 4
        PAUSE 1000
        LOW 1
        LOW 4
        PAUSE 1000
    NEXT
```

APPENDIX B – PWM AND PFM PROGRAMS

Pulse Width Modulation Program

```
'{$STAMP BS2sx}
'   Author: Christine Miyachi
'   Final Project 6.836
'
' Contract with higher width at first
    ' first pulse
Reps VAR BYTE
FOR Reps = 1 TO 100
    HIGH 3
    HIGH 4
    HIGH 5
    PAUSE 200
    LOW 3
    LOW 4
    LOW 5
    PAUSE 50
    ' second pulse
    HIGH 3
    HIGH 4
    HIGH 5
    PAUSE 100
    LOW 3
    LOW 4
    LOW 5
    PAUSE 50
    ' third pulse
    HIGH 3
    HIGH 4
    HIGH 5
    PAUSE 25
    LOW 3
    LOW 4
    LOW 5
    PAUSE 2000
NEXT
```

Pulse Frequency Modulation Program

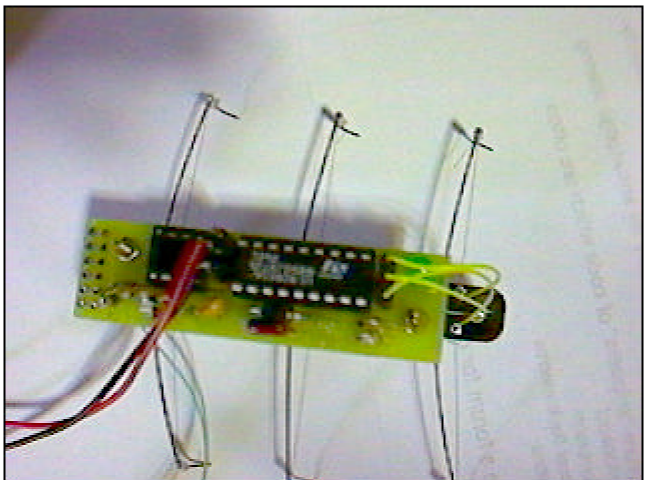
```
'{$STAMP BS2sx}
'   Author: Christine Miyachi
'   Final Project 6.836
'
' Contract with higher frequency at first
Reps VAR BYTE
Loops VAR BYTE
FOR Loops = 1 TO 100
    FOR Reps = 1 to 4
        HIGH 3
        HIGH 4
        HIGH 5
        PAUSE 50
        LOW 3
        LOW 4
        LOW 5
        PAUSE 50
    NEXT
    ' lower frequency pulses
    FOR Reps = 1 to 2
        HIGH 3
        HIGH 4
        HIGH 5
```

```
        PAUSE 100
        LOW 3
        LOW 4
        LOW 5
        PAUSE 100
    NEXT
    PAUSE 2000
NEXT
```

APPENDIX C

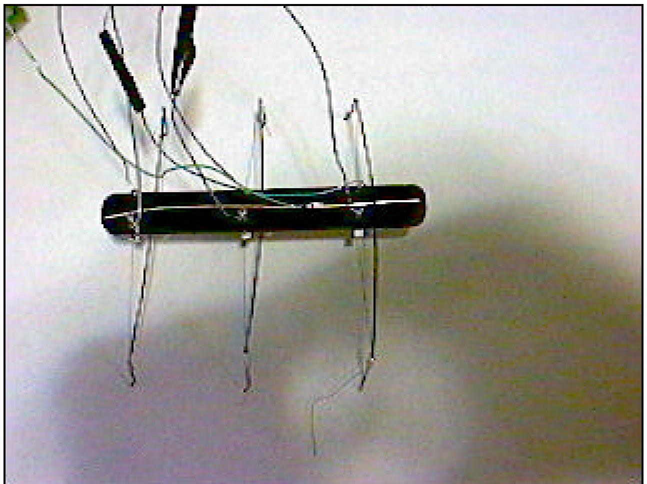
There are the video clips of the robot. They are *.avi files. If you double click on them, they should play.

The first clip displays the timer controller controlling a robot hard-wired with a tripod gate. A tactile sensor is wired into the circuit as described into the report, which changes the timing of the gate. When the sensor is fired, the robot moves faster.



robot-timer

This next video shows the some of the gates that were programmed. It first shows the pace gate, where three legs move at once on each side. The caterpillar gate is shown next. This one moves the front and back legs together and then the middle legs together. The camera focuses on one side so you can see the gates clearly. The tripod gate is also programmed but was left out of the video because it is demonstrated in the manual controller video.



robot-micro

BIBLIOGRAPHY

1. Conrad, James, Mills, Jonathan, *Stiquito: Advanced Experiments with a Simple and Inexpensive Robot*, IEEE Computer Society, 1997

2. http://eu.st.com/stonline/index.shtml - The ST Microelectronics online store; chip specifications are here.

3. Basic Stamp Manual, Version 2.s0, http://www.parallelaxinc.com