A Novel Binary Actuator Using Shape Memory Alloy

By

R. Dodge Daverman

B.S. Mechanical Engineering (2003)
B.S. Mathematics (2003)
Southern Methodist University

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2005

Author ______________________________
Department of Mechanical Engineering
May 6, 2005

Certified ____________
S. James Biggs
h Scientist
Supervisor

Certified by __ ________
A. Srinivasan
Senior Research Scie ,al Engineering
artment Reader

Accepted by ______________ __________
r Lallit Anand
Chairman, Department Committee on Graduate Students

A Novel Binary Actuator Using Shape Memory Alloy

By

R. Dodge Daverman

**Submitted to the Department of Mechanical Engineering
on May 6, 2005 in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Mechanical Engineering**

ABSTRACT

In situations that demand the use of the high-bandwidth, high-quality sense of vision for interactions with the physical world it would be beneficial to have a wearable tactile display that takes advantage of touch to communicate information to the user without causing visual distractions. This thesis presents the design and development of a novel actuator that can be configured into thin, flexible arrays to meet this need for wearable tactile displays. The actuator presented uses the strain recovery property of the martensitic transformation of Nitinol, a Shape Memory Alloy (SMA), to generate the actuation force. A compliant bistable mechanism provides the restoring force that pre-strains the martensitic Nitinol, and thus makes the actuator binary. Binary actuation alleviates some of the problems that would otherwise limit the effectiveness of Nitinol in wearable haptic systems. To increase the potential for commercial success, manufacturability issues are considered throughout the development cycle to ensure the potential for economical large scale production. The paper concludes with the presentation of three different prototypes. Their successes and failures are discussed along with recommendations for future work.

Thesis Supervisor: Dr. S. James Biggs
Title: Research Scientist, Research Laboratory of Electronics at MIT

Acknowledgements

I would like to thank the many people that I have worked with in the Touch Lab for making these past two years at MIT such a wonderful experience. In particular, I would like to thank my thesis advisor, Dr. S. James Biggs, whose creative vision inspired the work of this thesis. The amount of one-on-one time that you gave me well exceeded my expectations, and really helped to teach me how to think like an engineer. I also wish to thank Dr. Mandayam A. Srinivasan, for his role in bringing me into the Touch Lab and for serving as a departmental reader to certify this thesis.

On a more personal note, I would like to thank my parents, Jim and Wendy Daverman, for the unconditional love and support that they have given me for all these years. You have always encouraged me to pursue my interests; although it took me a long time to become passionate about engineering, the patience that you showed along the way allowed me to develop into the person that I am today.

Lastly, many thanks and much love goes to Natalie for sacrificing so much to move to Boston with me to make this all possible. I could not have done this without you.

Table of Contents

1 **Introduction** **15**

2 **Shape Memory Alloys** **19**

2.1 Background Information *19*

2.2 General Characteristics *21*

2.3 Nitinol Wire *24*

3 **Conceptual Design** **27**

3.1 Typical Restoring Mechanism *27*

3.2 Individual Actuators *29*

3.3 Actuator Arrays *31*
3.3.1 Mechanically Parallel Arrangement 32
3.3.2 Mechanically Series Arrangement 33

3.4 Target Specifications *35*

4 **General Design Approach** **36**

4.1 Problem Decomposition *36*

4.2 Design for Manufacturing *38*

4.3 Bistable Mechanism Design *39*
4.3.1 Material Selection 39
4.3.2 Mathematical Model 40

4.4 Nitinol Patch Design *43*

4.5 Drive Electronics *45*

5 **Prototype 1** **51**

5.1 Bistable Substrate Design *51*

5.2 Nitinol Patch Design *53*

5.3 Drive Electronics *58*

5.4 Testing *60*

6 **Prototype 2** **65**

6.1 Actuator Array Design *65*

6.2 Bistable Substrate Design *67*

6.3 Nitinol Patch Design *69*

6.4 Anchoring the Nitinol *76*

6.5 Testing *78*

7 **Prototype 3** **81**

7.1 Nitinol Patch Design *81*

7.2 Drive Electronics *85*

7.3 Testing 88

8 Conclusions 91

Works Cited 95

Appendix A: Nitinol Winding Station 97

Appendix B: Drive Electronics for Prototype 3 111

List of Figures

Figure 2.1 A typical plot of characteristic properties of Nitinol as a function of temperature indicates a hysteresis associated with the martensitic transformation [10]. 23

Figure 2.2 This stress-strain plot shows that winding the Nitinol at a tension of 20,000 psi will result in approximately 3.5% strain in the wire when the tension is released [16]....25

Figure 3.1 A linear spring is used as the typical restoring mechanism. When hot the Nitinol wire is stiff, but when it cools the spring will stretch it out [16]....28

Figure 3.2 A snap through buckled beam (left) and its corresponding force deflection curve (right) [19]....30

Figure 3.3 A mechanically parallel array of three actuators shown from above (a). A view from the front (b) shows the center actuator in its concave up position while the other two are in their concave down positions. A profile view (c) shows one of the concave down actuators applying a force to skin, while the concave up actuator in the middle is intentionally out of reach of the skin....32

Figure 3.4 A mechanically series array is made up of individual actuators mechanically connected to each other. In this array, adjacent actuators are in opposite binary states....33

Figure 3.5 Both stable positions of a single actuator fixed at one end (a); a second actuator connected to the end of the first actuator (b) has a different orientation and endpoint location depending on the state of the first actuator. 34

Figure 3.6 One of the 65,536 different configurations that a 16 actuator array can make....35

Figure 4.1 Functional Decomposition of the Actuator Array Design Problem....37

Figure 4.2 Relief slits in the bistable substrate (left) create tension beams that are formed (right) to produce the buckled beam....38

Figure 4.3 Screenshot of working model simulation used to design the bistable mechanism. 41

Figure 4.4 Comparison of simulation of a simply loaded bistable mechanism with experimental obtained data validates the mathematical model used for simulation..42

Figure 4.5 Nitinol fibers are wound onto a take-up spool at evenly spaced intervals (left); cutting along dotted line (right) reveals two strips containing Nitinol patches of evenly spaced parallel fibers..45

Figure 4.6 A multiplexing scheme used to limit the number of drive lines that are needed to activate actuators on the array. The dotted line shows the path of current flow when element 4T is triggered. 46

Figure 4.7 Multiplexing scheme using integrated circuits to increase the number of actuators that the BASIC Stamp can address. The configuration pictured addresses 48 resistive elements (24 actuators) with only 8 control lines. 49

Figure 5.1 Equilibrium tensions as simulated length of the Nitinol is shortened. Approximately 14 data points did not converge to steady state, and are omitted for clarity. The length of Nitinol at assembly (A) is used to compute the unstrained length (D). The plot shows that shortening the Nitinol increases the tension until the onset of snap-through (B). Eventually, the Nitinol will go into slack (C) when the beam has passed through the unstable region. The 5% strain limit (E) cannot be induced by this mechanism since the Nitinol is in slack at this length...53

Figure 5.2 A Nitinol patch consisting of one long strand of Nitinol wire wound around a PVDF frame. 57

Figure 5.3 A PVDF frame that uses a serpentine pattern to reduce its contribution to the bending stiffness of the actuator. 57

Figure 5.4 Transistor as a current source. 59

Figure 5.5 Measured resistance changes (left) due to cooling of the Nitinol after four typical pulses are superimposed (points). Exponential curves (lines) with time constants (τ) in the range of 0.06-0.20 seconds give a least-squares fit to the full data set (left). This information is mapped into a temperature curve (right) that predicts passage through the TTR in 0.05 seconds.................................61

Figure 5.6 Nitinol bending around the PVDF shows a maximum radius of curvature equal to 0.0025". 63

Figure 6.1 A picture of four actuators on a mechanically series 8 actuator array that uses an external carbon heater to induce the martensitic transformation. Both sides of the actuator are shown to illustrate the path that the electrical traces take to get the current out to the carbon heater....66

Figure 6.2 Equilibrium tensions as simulated length of the Nitinol is shortened. The length of the Nitinol at assembly (A) is used to compute the unstrained length (D). As the length is decreased, the mechanism produces a maximum tension in the Nitinol (B), before going into slack (C). Although there is tension in the wire at the unstrained length (D), the region from (B) to (C) is unstable and so the beam will snap through if the Nitinol can supply a force greater than indicated at (B). The 5% strain limit on Nitinol (E) cannot be induced by the mechanism since the curve indicates no tension in the wire. 68

Figure 6.3 Constant power curves used to determine the target resistance of the external carbon heaters. 72

Figure 6.4 Two different carbon heater designs: (left) a carbon patch covers the span of the bistable mechanism; (right) a striped pattern is used to maintain the required length to width ratio....74

Figure 6.5 Picture of Nitinol winding equipment used to make prototypes two and three....75

Figure 7.1 A Nitinol patch for Prototype 3 (left) uses silver ink to connect adjacent fibers into parallel groupings. These groupings are connected in a pattern that increases the equivalent resistance of the Nitinol patch (right). 82

Figure 7.2 The current required to supply 0.2 Joules of energy to the Nitinol in both a purely parallel configuration and a configuration of 4 groups of 3 parallel fibers 83

Figure 7.3 An electrically conductive adhesive is stenciled over a Nitinol patch with pre-cut holes (left). These holes allow for the conductive adhesive to bond to the scrap substrate. Then, peeling the carriage tape off of the scrap substrate reveals a Nitinol patch (bottom right) that can be consistently produced..85

Figure 7.4 Schematic showing how a voltage regulator can be wired as a two terminal current limiter. 86

List of Tables

Table 2.1 Manufacturer's operating guidelines for various diameters of Nitinol wire [14]. 25

Table 3.1 Target Specifications for the Individual Actuator 35

Table 5.1 Parameters that describe the bistable buckled beam of Prototype 1 as obtained from simulation. 52

Table 5.2 Properties of Nitinol patches made from different diameter wires for the bistable mechanism of Prototype 1. 54

Table 6.1 A list of parameters that describe the bistable buckled used in Prototype 68

Table 6.2 Properties of Nitinol patch with different diameters based on bistable mechanism for Prototype 2. 70

Table 6.3 Nitinol Anchoring Adhesive Selection Table 77

1 Introduction

Increasingly, portable and mobile computers are being used in the service of active individuals performing a variety of tasks in the physical world including: driving cars, flying airplanes, maintaining communications and connectivity among dispersed individuals, and performing scheduled maintenance of aircraft and vehicles. Tasks of this nature demand the use of the high-bandwidth, high-quality sense of vision for interactions with the physical world because of either the urgency of hazardous conditions or the complexity of the tasks being performed. In such cases, it would be beneficial to have a wearable display that takes advantage of other human senses, such as touch and hearing, to guide and communicate with the user in a way that does not cause visual distractions.

The US military currently makes limited use of tactile displays based on Consumer Off-The-Shelf (COTS) pneumatic actuators and electromagnetic vibrator [1, 2]. Although much can be accomplished with COTS pneumatic and electromagnetic actuators, fundamental limitations of these technologies prevent their widespread use in portable tactile displays. The main drawback of pneumatic actuators is the requirement for a compressed air supply making it impossible to implement a portable tactile display.

Electromagnetic vibrators, on the other hand, are merely pager motors repackaged. They lack the thinness and conformability that are desirable in a portable tactile display worn against the skin. Furthermore, the diffuse buzzing percept they produce lacks the spatial and temporal resolution necessary to rapidly transmit high-bandwidth information through the skin [3]. Non-vibrating actuators are also likely to be more comfortable for prolonged use. It is reported that prolonged stimulation at vibrotactile frequencies causes numbness and pain [4] that may limit user acceptance of spatial orientation displays that continually vibrate.

To overcome current limitations in tactile displays requires development of actuators that emphasize lightness, conformability, and spatial resolution.

- *Lightness* –The dismounted soldier may already be carrying 70 kg of equipment. Wearable and surface-mounted tactile displays should add as little to this burden as possible.
- *Conformability* – Wearable tactile display must be conformable enough to accommodate differences in body shape between users, and changes in conformation of the body surface during user movement. Surface-mounted tactile displays must be conformable enough to laminate to non-planar surfaces (e.g., a rifle stock, a cell phone, ... etc).
- *Spatial Resolution* – Higher spatial resolution allows for transmission of more information through the skin per unit time. The key to spatial resolution is the low-frequency response of the actuator. It is well known that the skin mechanoreceptors that sense non-vibrating skin tractions (<5 Hz) have spatial resolution 10-100 times better than those that sense vibrations (~200 Hz) in the frequency range produced by pager motors [5].

The work presented in this paper was conducted at the Laboratory for Human and Machine Haptics at the Massachusetts Institute of Technology in an ongoing project to develop tactile displays using alternative actuator technologies that have the potential to overcome these limitations. Currently there are several projects running in parallel to develop wearable tactile displays using different actuation technologies including: piezo-electric actuators, dielectric elastomer actuators and Shape Memory Alloy (SMA) actuators. The thesis focuses on the development of actuators based on Shape Memory Alloy.

Surveys of actuator materials [6-8] suggest that SMA based actuators have the conformability and energy density required for a quantum leap in tactile display technology. In addition, the cost of SMA materials is sufficiently low, that eventual competition with COTS actuators is feasible. SMA has an energy density (>5000 kJ/m^3) that is more than 1,000 times higher than conventional electromagnetic actuators (<10 kJ/m3) which implies that an SMA actuator can supply the same actuation force in a much smaller package. For this reason, it is an attractive actuator technology for wearable tactile displays [9].

The objective of the work presented in this thesis is to design and develop a low cost SMA actuator for use in wearable tactile displays. The thesis documents the iterative design process involved in realizing this objective. Chapter 2 presents a brief history of SMA followed by a general explanation of how it can be used to generate an actuation force. Chapter 3 presents the conceptual design of the actuator and how it is incorporated into a tactile display. Chapter 4 breaks the design problem down into several component problems and highlights the methods of solving these problems. Chapters 5, 6 and 7 present three different prototypes that have been developed; each discussing the degree to which they satisfy the design objectives. Finally, chapter 8 concludes the paper by commenting on the strengths and weaknesses of each prototype and by issuing advice on the direction of further research.

2 Shape Memory Alloys

The term Shape Memory Alloy (SMA) refers to a class of metals that is capable of returning to a pre-defined shape when heated through a certain temperature range [10]. Many alloys exist that exhibit the shape memory effect; however, only those capable of generating large forces during the strain recovery (Nitinol, CuZnAl, CuAlNi, and a few others) have potential for use in actuator applications [11]. The most commercially successful of these alloys is Nitinol, due to both its desirable mechanical properties, and the understanding of how processing and heat treatment affects these properties [10]. This chapter presents a brief history of the discovery of Nitinol followed by a discussion of how Nitinol can be used as the basis of an actuator. The chapter concludes by presenting the some of the properties of the commercially available brand of Nitinol that is used to develop the actuators of this thesis.

2.1 Background Information

William J. Buehler discovered Nitinol in 1959 while developing alloys of equiatomic nickel-titanium (Ni-Ti) composition for the U.S. Naval Ordinances Laboratory (NOL) [12]. The observations that led to the discovery of the shape memory property of Nitinol happened by chance while he was working on an unrelated problem. In an interview with the author of a journal article that documents his discovery, Buehler recalls the day of his first serendipitous observation [12].

> I distinctly remember my very exciting discovery of the acoustic damping change with temperature change near room temperature. This unusual event unfolded when my... assistant... and I were melting a number of [Nitinol] bars in the arc-melting furnace. On the day in question (circa 1959), six arc-cast bars were made. While cooling on the transite-topped table, the first bars arc-cast into bar form had cooled to near room

temperature, while the last bars to be cast were still too hot... to be handled with bare hands. Between the cool (first bar) and the very warm bar (last bar) were four arc-cast bars possessing a broad spectrum of temperatures.... My "hands on" approach caused me to take the cooler bar(s) to the shop grinder to manually grind away any surface irregularities that might produce a subsequent scaly or bad... surface. In going from the table to the bench grinder, I *purposely* dropped the cool (near room temperature) bar on the concrete laboratory floor [a quick test to determine roughly the damping capacity of an alloy]. It produced a very dull "thud," very much like what one would expect from a similar size and shape lead bar. My immediate concern was that the arc-casting process may have in some way produced a multitude of micro cracks within the bar – thus producing the unexpected damping phenomena. With this possibly discouraging development in mind, I decided to drop the others on the concrete floor. To my amazement, the warmer bars rang with bell-like quality. Following this I literally ran with one of the warmer bars (that rang) to the closest source of cold water – the drinking fountain – and chilled the warm bar. After thorough cooling the bar was again dropped on the floor. To my continued amazement it now exhibited the leaden-like acoustic response. To confirm this unique change, the cooled bars were heated through in boiling water – they now rang brilliantly when dropped upon the concrete floor. Subsequent discussions with my melter assistant revealed that he had in no way mixed or altered the alloy compositions during repeated melting. This immediately alerted me to the fact that the marked acoustic damping change was related to a major atomic structural change, related only to minor temperature variation.

While this observation led to the discovery of some interesting properties of this equiatomic nickel-titanium alloy in this same temperature range, it would take another act of chance before the shape memory effect would first be observed. To demonstrate Nitinol's ability to resist fatigue, Buehler had prepared a strip of Nitinol by bending it in a zigzag fashion so that the strip could be compressed and stretched like a spring over a large number of cycles without failing from fatigue. During this demonstration, the strip of Nitinol was passed around the room. For whatever reason, one of the men in the room, Dr. David S. Muzzey, used his pipe lighter to apply heat to the Nitinol strip. To everyone's astonishment, the Nitinol strip stretched out longitudinally and thus the property of heat-induced strain recovery in Nitinol was discovered [12]. Then, in 1962, Dr. Frederick E. Wang, a Ph.D. in physical chemistry, brought his expertise in crystal physics to Buehler's research group at NOL and was responsible for explaining how the shape-memory property of Nitinol actually works [12].

2.2 General Characteristics

An in depth description of the shape memory effect of Nitinol is beyond the scope of this paper; however, this section presents a brief review of how the shape memory effect can be used as the basis for a new actuator.

The shape memory effect in Nitinol is the result of a solid state phase transformation from martensite to austenite which is commonly referred to as a martensitic transformation. Austenite, the parent phase, exists at higher temperatures and has an atomically ordered body-centered cubic crystal structure [10]. The lower temperature phase, martensite, has a more complex crystal structure that is capable of shear on the atomic level [12].

The shape that the Nitinol will revert to upon the martensitic transformation is set by heating the Nitinol to 500°C and constraining it to the desired position. This process restructures the body centered cubic crystal structure of the austenite phase [12]. Then, the Nitinol is allowed to cool through the transition temperature causing it to transform into the martensite phase. This reordering of the atomic structure due to the transformation from austenite to martensite produces no macroscopic changes in the Nitinol [11]. In other words, no visible strain recovery occurs when the Nitinol transforms from the high temperature phase to the low temperature phase.

When a load is applied to the Nitinol in the martensite phase, its ability to shear on the atomic level allows it to deform through a twinning mechanism at relatively low stress levels. This twinning mechanism is reversible and can accommodate strains up to 8% of the austenite length. Strains larger than 8% are generated through an irreversible mechanism that creates dislocations in the Nitinol [12]. When the Nitinol is heated through its transition temperature it undergoes the martensitic transformation and reverts back to the austenite crystal structure, ~~recovering~~ any strain due to twinning. During this transformation, any strain due to twinning is recovered. This strain recovery is capable of producing large forces and will return the Nitinol to its original unstrained shape [11]. Of course, any strains due to dislocations are not recovered.

A plot of length versus temperature for a typical Nitinol specimen, under a constant load, is shown in Figure 2.1 in which the length of the specimen is measured as it is heated and then cooled [10]. This plot is useful in determining the transition temperature of the Nitinol. The transition does not happen at a single temperature but rather over a range of temperatures known as the transition temperature range (TTR). While all Nitinol is composed of nearly equal parts nickel and titanium, the TTR can be controlled by adjusting the exact composition of the alloy [10]. However, there is always a hysteresis associated with the martensitic transformation (see Figure 2.1). In other words, the TTR for heating and cooling do not overlap.

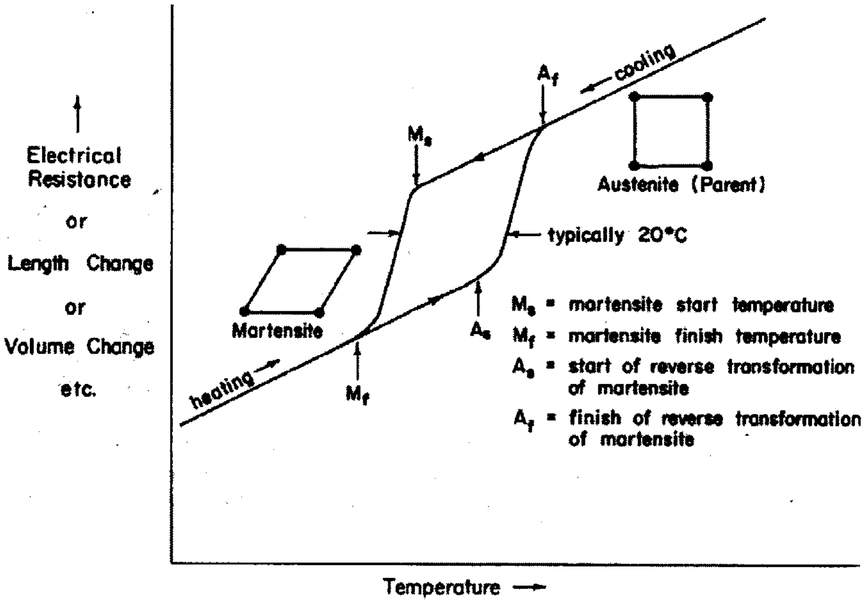


Figure 2.1 A typical plot of characteristic properties of Nitinol as a function of temperature indicates a hysteresis associated with the martensitic transformation [10].

Nitinol recovers strain when it goes through a heat-induced martensitic transformation; the method used to heat or cool the Nitinol has no effect on its ability to recover strain. This gives the designer freedom in selecting appropriate methods of heating and cooling for a specific application. In the past, solid state engines have been made from Nitinol that rely on hot and cold water baths to heat and cool the Nitinol [13]. One of the first successful commercial applications of Nitinol was a pipe coupler for the F-14 fighter jet. In this application, the coupler was radially expanded at extremely low temperatures in a liquid nitrogen bath. At this low temperature the pipes to be joined are fitted into the coupling. Then the coupling is removed from the liquid Nitrogen bath and allowed to warm up to room temperature. When the temperature of the coupler passes through the TTR (-180 °C) it generates a compressive force that holds the two pipes together. The low TTR for this application was designed so that the coupler would remain in the austenite phase at all operating temperatures of the aircraft, thus ensuring a continuously sealed joint [12]. Other actuator applications in which hot and cold water baths are not desirable have demonstrated success heating the Nitinol by passing an electrical current (DC, AC, or PWM) through the Nitinol [14, 15].

2.3 Nitinol Wire

While the previous section described the behavior of Nitinol in general, this section focuses on the properties of the commercially available Nitinol wire used throughout the remainder of this paper. In wire form, Nitinol uses the shape memory effect to convert thermal energy into mechanical energy in the form of a linear contraction. To accomplish this, a longitudinal force is first applied to the Nitinol wire in the martensite phase to induce an initial longitudinal deformation. Then, when the Nitinol is heated through the TTR the strain recovery generates a contracting force. Because linear contraction is useful in many different applications, this is the form in which it is typically mass produced and sold at relatively low cost.

The Nitinol wire used in this thesis is produced by Dynalloy Inc., and sold under the trade name Flexinol™ in a variety of diameters and transition temperatures [14]. This wire is wound onto spools at the factory under a tension of 20,000 psi which means that it is pre-strained approximately 3.5% out of the box (Figure 2.2) [16]. Keeping the stress levels in the wire below 15,000 psi ensures that the permanent strain remains less than 0.5% and that the fatigue life is in the tens of millions of cycles. Table 2.1 shows the recommended operating conditions of a variety of Flexinol™ wires [14].

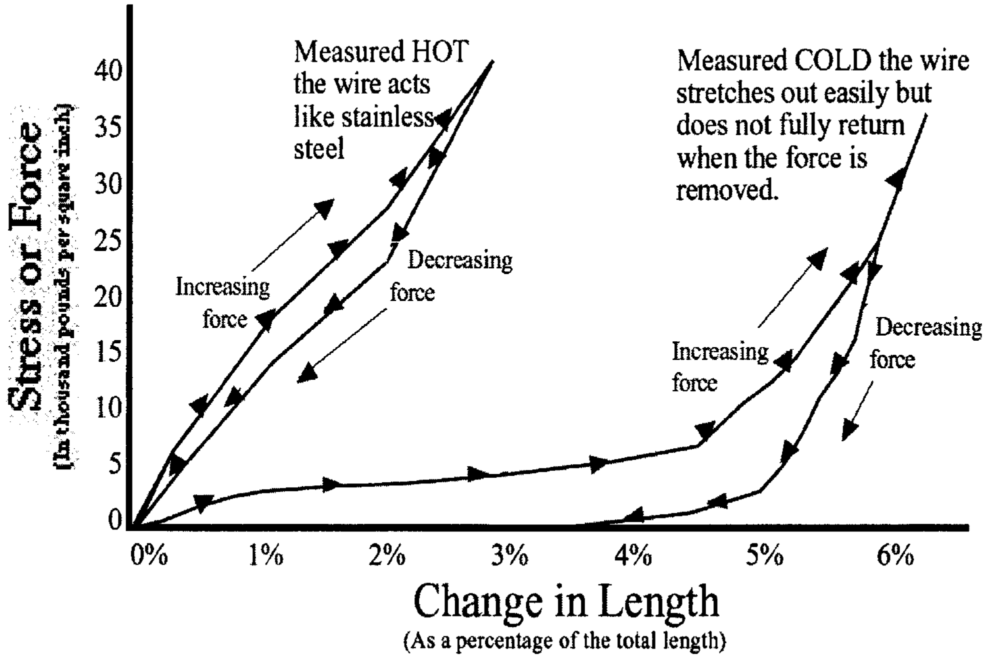
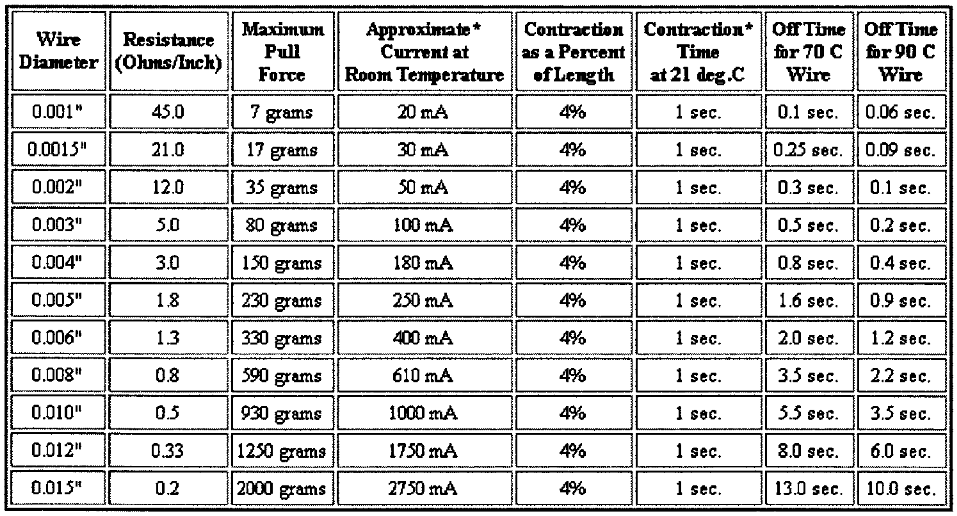


Figure 2.2 This stress-strain plot shows that winding the Nitinol at a tension of 20,000 psi will result in approximately 3.5% strain in the wire when the tension is released [16]



Wire Diameter	Resistance (Ohms/Inch)	Maximum Pull Force	Approximate* Current at Room Temperature	Contraction as a Percent of Length	Contraction* Time at 21 deg.C	Off Time for 70 C Wire	Off Time for 90 C Wire
0.001"	45.0	7 grams	20 mA	4%	1 sec.	0.1 sec.	0.06 sec.
0.0015"	21.0	17 grams	30 mA	4%	1 sec.	0.25 sec.	0.09 sec.
0.002"	12.0	35 grams	50 mA	4%	1 sec.	0.3 sec.	0.1 sec.
0.003"	5.0	80 grams	100 mA	4%	1 sec.	0.5 sec.	0.2 sec.
0.004"	3.0	150 grams	180 mA	4%	1 sec.	0.8 sec.	0.4 sec.
0.005"	1.8	230 grams	250 mA	4%	1 sec.	1.6 sec.	0.9 sec.
0.006"	1.3	330 grams	400 mA	4%	1 sec.	2.0 sec.	1.2 sec.
0.008"	0.8	590 grams	610 mA	4%	1 sec.	3.5 sec.	2.2 sec.
0.010"	0.5	930 grams	1000 mA	4%	1 sec.	5.5 sec.	3.5 sec.
0.012"	0.33	1250 grams	1750 mA	4%	1 sec.	8.0 sec.	6.0 sec.
0.015"	0.2	2000 grams	2750 mA	4%	1 sec.	13.0 sec.	10.0 sec.

Table 2.1 Manufacturer's operating guidelines for various diameters of Nitinol wire [14].

3 Conceptual Design

The design objective of this thesis is to create an actuator that is based on SMA technology that can be used in a wearable tactile display. Because the force that Nitinol is capable of exerting is the result of it "recovering" strain, any Nitinol based actuator needs a mechanism that provides martensitic pre-strain.

This chapter presents a typical pre-strain mechanism found in Nitinol based actuators and discusses why it is not well suited for wearable haptic applications. A conceptual design is then presented using a bistable mechanism to overcome the limitations of the typical pre-strain mechanism. Two different concepts are then presented, and potential applications are discussed for configuring multiple actuators into large arrays controlled by a central processor. Finally, the chapter concludes with a set of target specifications for the design.

3.1 Typical Restoring Mechanism

A typical mechanism used to provide the restoring force that pre-strains the Nitinol is a simple linear spring (Figure 3.1) [14-16]. In this case, when the power is off, the Nitinol is in the martensite phase, and the spring applies a force that is strong enough to induce a 4% stain. When the power is on, the Nitinol is heated above the TTR causing it to undergo a martensitic transformation into the austenite parent phase. As discussed in chapter 2, this martensitic transformation generates large forces as the Nitinol recovers the strain that was present in the low temperature phase. These forces are capable of overcoming the force of the linear spring; thus, returning the Nitinol to its original unstrained length. When the power is turned off, the Nitinol cools below the transition temperature and the crystal structure transforms back into the weaker martensite. Again, in the martensite phase, the force of the spring is strong enough to induce a 4% strain in the Nitinol.

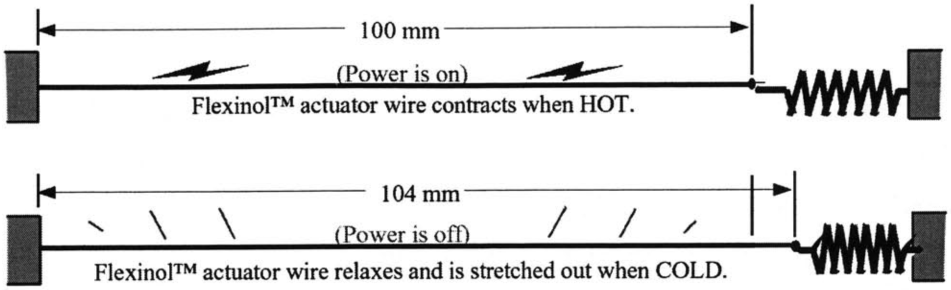

100 mm
(Power is on)
Flexinol™ actuator wire contracts when HOT.
104 mm
(Power is off)
Flexinol™ actuator wire relaxes and is stretched out when COLD.

Figure 3.1 A linear spring is used as the typical restoring mechanism. When hot the Nitinol wire is stiff, but when it cools the spring will stretch it out [16].

This typical set up can be effective in certain applications; however, it is far from ideal for portable tactile displays. In the example shown in Figure 3.1, the Nitinol must stay above the transition temperature in order to maintain the 4mm displacement. Since heat is required to keep the Nitinol in the austenite phase, power must be supplied continuously to the Nitinol. The power required is sustained and therefore battery life is diminished. For example, a standard 9V alkaline battery (500 mAhr) stores 16,200J of energy. If this battery is used to pass a 30 mA current directly through a Nitinol fiber (0.0015" dia., 180 mm long) to keep it above its transition temperature it will be exhausted in approximately 22 hours. Therefore, this design consumes too much power to be used as a portable device.

Safety is another problem with continuously powering the Nitinol. In a tactile display, the force output of the actuator is transmitted to the skin. This is most effectively accomplished by placing the actuator directly against the skin, which presents a safety issue since the typical transition temperature of Nitinol is 90°C. This is far too hot for prolonged skin exposure as tissue damage begins at 45°C [17]. To use a device of this nature as a tactile display would require an insulating layer between the skin and the Nitinol. This would decrease the cooling rate of the Nitinol resulting in a reduced peak frequency.

3.2 Individual Actuators

Despite its limitations, the typical mechanism possesses the desirable properties of being passive. It uses energy stored in strain to provide the restoring force needed to pre-strain the Nitinol in its martensitic phase. The ability to store energy as strain energy is a defining property of compliant mechanisms [18]. The linear spring of Figure 3.1 is nothing more than a simple compliant mechanism. While this particular mechanism is not well suited for use in portable tactile displays, it does reveal the advantages of using a compliant structure to supply the restoring force. Therefore, a more complex compliant mechanism is needed to overcome the limitations of the typical mechanism while remaining passive.

The compliant mechanism that is used to provide the restoring force is the well known snap-through buckled beam. It is a slender, flexible beam with both ends constrained by a compressive force such that the beam buckles into the configuration shown in Figure 3.2. The generalized force-deflection curve for this mechanism indicates that no force is required to maintain zero deflection at the center point (3); however, this point is unstable [19]. Small perturbations in either direction require an opposing force to maintain the deflection. If no such force is present, the beam will revert to its maximum deflection (1 or 2 depending on direction of perturbation). These two maximum deflection positions are stable and so the beam is said to be bistable.

If the beam is initially in the fully deflected position (1), and a small vertical force is applied in the direction of the arrow (Figure 3.2), the beam will deflect into position (2). Releasing the force in this position would result in the beam returning to the stable concave down position. However, if the force is increased, the beams center point deflection continues to decrease. At the apex of the curve, the applied force balances with the restoring force of the beam to maintain a certain deflection. At this point, reducing the center point deflection requires less force than maintaining it. Therefore, a slight increase in the force will cause the beam to rapidly deflect into the other stable position (4). This is the snap-through phenomenon and is aptly named for the sound that it produces.

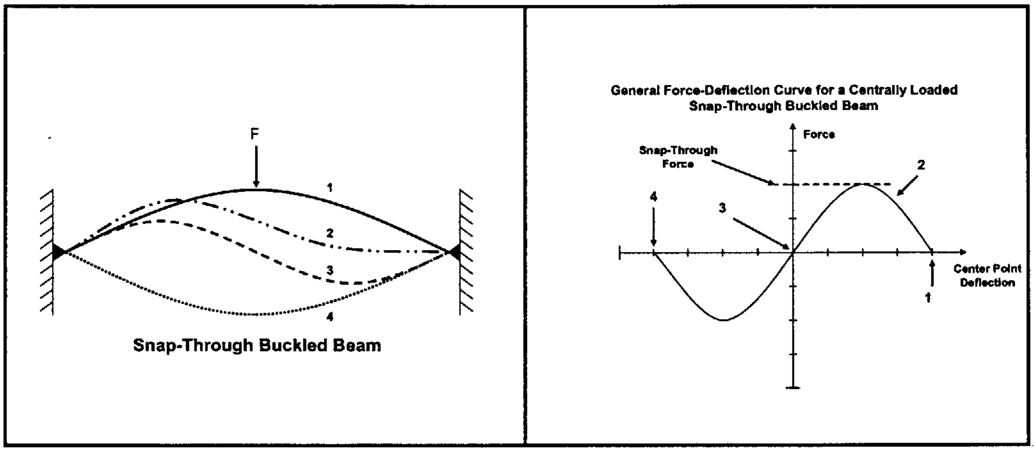

F
1
2
3
4
Snap-Through Buckled Beam
General Force-Deflection Curve for a Centrally Loaded Snap-Through Buckled Beam
Force
Snap-Through Force
2
4
3
Center Point Deflection
1

Figure 3.2 A snap through buckled beam (left) and its corresponding force deflection curve (right) [19].

Unlike the typical mechanism of 3.1, the bistable mechanism requires two separate Nitinol patches, one on either side of the beam. When the beam is in the concave down position, and the Nitinol on the top is heated through its transition temperature, the force generated from the martensitic transformation will push the beam through to the other stable position. After passing through the unstable position, the restoring force of the beam switches directions and provides the force to pre-strain the Nitinol patch on the bottom of the beam. Heating the Nitinol on the bottom then returns the beam to the original concave down position.

It is the bistable property of the snap-through buckled beam that makes it possible to use Nitinol in wearable haptic displays. The ability of the mechanism to hold one of two stable states means that sustained power is not needed. Instead, power is only required to heat the Nitinol when the actuator needs to change states. This creates significant power savings compared to the typical restoring mechanism. Remember, the typical mechanism has a battery life of only 22 hours on a single 9V battery. Making the restoring mechanism bistable produces approximately 81,000 transitions out of the same 9V battery using a 0.2J per transition. Because the mechanism can hold two stable positions, it is unlikely that any application would require an average frequency over the lifetime of the battery to be greater than 0.5 Hz. At this frequency, the 81,000 transitions will last approximately 45 hours.

Solving the power consumption problem ends up solving the safety issue as well. In the case of the typical restoring mechanism, where the Nitinol produces a force that holds the state of the actuator, the Nitinol needs to maintain a temperature that is above 90 °C. The bistable property of the substrate removes this problem by requiring the Nitinol to be hot only for brief durations (~100ms or less) as it pushes the beam through the unstable region.

Another benefit of making the restoring mechanism bistable is that the actuator becomes binary. Controlling a binary actuator is simple and inexpensive. Because there are only two states, control becomes a matter of triggering the mechanism to change states, and storing the state of the actuator. This can be done open loop, thus simplifying the design, and eliminating the need for expensive sensors.

3.3 Actuator Arrays

The concept of the individual actuator described above is a novel approach to SMA actuator design. To apply this actuator to tactile displays, the idea is extended by interconnecting individual actuators to create arrays consisting of many actuators. This section presents two different conceptual array configurations and discusses some of their potential applications.

3.3.1 Mechanically Parallel Arrangement

A simple extension of the individual actuator involves lining up multiple actuators side by side in a mechanically parallel arrangement as shown in Figure 3.3. In this array of actuators, nothing is mechanically different from the individual actuator. The only thing that separates this idea form the individual actuator of section 3.2 is that the triggering of actuators is handled by a central processor. Therefore, the only added complexity associated with a mechanically parallel array is found in the design of the drive electronics. For the array of three actuators pictured in Figure 3.3, this added complexity is trivial; however, an array of 300 actuators would require a clever wiring solution.

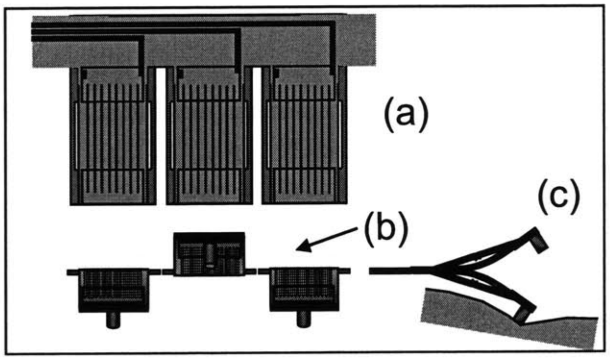


Figure 3.3 A mechanically parallel array of three actuators shown from above (a). A view from the front (b) shows the center actuator in its concave up position while the other two are in their concave down positions. A profile view (c) shows one of the concave down actuators applying a force to skin, while the concave up actuator in the middle is intentionally out of reach of the skin.

An array of this type naturally lends itself to tactile applications. One potential application is to mount it against the skin in such a way that when it is in the concave up position, the end floats freely in air, but when it is in the concave down position, it contacts the skin as pictured in Figure 3.3c. This type of an array could be embedded into clothing, for example, to make garments capable of transmitting data to the user through haptic feedback.

3.3.2 Mechanically Series Arrangement

A second conceptual design for arrays of actuators is based on a mechanically series configuration. In this arrangement, individual actuators connect to each other in an end to end fashion as shown in Figure 3.4. These connections are made so that there is a zero curvature boundary condition where the two actuators are joined. This boundary condition ensures a mechanical coupling so that the state of one actuator affects the position and orientation of the other actuators on the array.

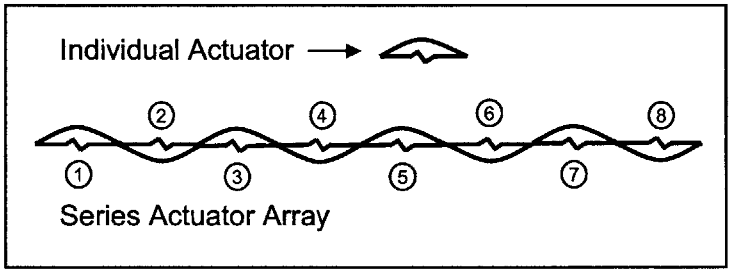


Figure 3.4 A mechanically series array is made up of individual actuators mechanically connected to each other. In this array, adjacent actuators are in opposite binary states.

When the individual actuator is viewed from the side, the two stable states form two distinct shapes: concave down or concave up (Figure 3.4 actuators 1 and 2 respectively). If one end of the actuator is fixed and the other end is free, as pictured in Figure 3.5a, then as it changes states, the free end of the actuator swings through an angle (snap-through angle), as it moves from concave down (solid) to concave up (dashed).

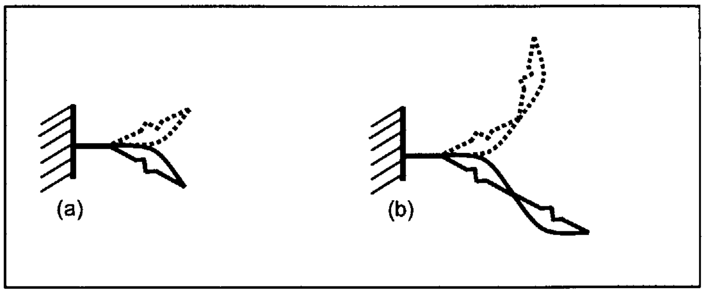


Figure 3.5 Both stable positions of a single actuator fixed at one end (a); a second actuator connected to the end of the first actuator (b) has a different orientation and endpoint location depending on the state of the first actuator.

When a second actuator is connected to the end of the first, its position and orientation relative to a global coordinate system are dependent on the state of the first array. This can be seen in Figure 3.5b, where the second actuator is attached to the first actuator in a concave up position and the state of the first actuator has a large effect on the position and orientation of the second actuator even when the second actuator remains in the same state. Figure 3.5b shows two of the four possible configurations of the two actuator array. The number of different configurations that a single array containing N actuators can make is 2^N. This means that a 16 element array has 65,536 different configurations. Figure 3.6 shows s 3D sketch of one of these configurations. In this view, it becomes clear that the different configurations the array takes on can be interpreted as different shapes. As the number of elements in the array increases, and the size of the actuators decreases, the array does a better job of approximating a continuous range of shapes.

The ability of the array to represent shapes makes it quite interesting and implies certain potential applications. In the field of haptics, such a device could make it possible to feel shapes in virtual environments. Applications in fields other than haptics would include: three dimensional pop-up books for children, dynamic sculptures, flexible robots … etc.

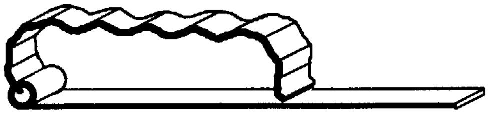


Figure 3.6 One of the 65,536 different configurations that a 16 actuator array can make.

3.4 Target Specifications

This chapter presented a novel concept for an actuator that relies on Nitinol to provide the actuation force. Ideas were then presented for how arrays of these individual actuators might be constructed for use in both tactile and shape displays. With a concept in place, and applications in mind, target specifications are generated to guide the design. A representative list of these specifications chosen to ensure actuator manufacturability, actuator performance, and user safety is shown in Table 3.1.

Target Specifications			
Metric	**Unit**	**Value**	**Rationale Criteria**
Length of Buckled Beam	mm	5-20	Manufacturability & Performance
Width of Buckled Beam	mm	< 10	Manufacturability & Performance
Overall Thickness of Actuator	in	< 0.100	Performance
Snap-Through Angle	Degrees	~ 90	Performance
Peak Frequency	Hz	> 3	Performance
Supply Voltage	Volts	< 36	Safety
Fatigue Life	Cycles	> 106	Performance

Table 3.1 Target Specifications for the Individual Actuator

4 General Design Approach

After generating a conceptual design, the next step in the design process is to formulate a plan for implementing the concept. This chapter begins by decomposing the problem into smaller more manageable component problems in an attempt to simplify the design task. These component problems are then discussed in the order in which they are solved in practice.

4.1 Problem Decomposition

Decomposing the design problem into functional sub-units is the first step in designing the actuator array. As shown, (Figure 4.1) the overall problem is initially broken down into two parts: (1) development of a unit actuator, and (2) development of a wiring system connecting the actuators to a central processor. The functions of the unit actuator are subdivided into three parts: (1) provision of the actuation force, (2) provision of the restoring force to the Nitinol, and (3) heating of the Nitinol.

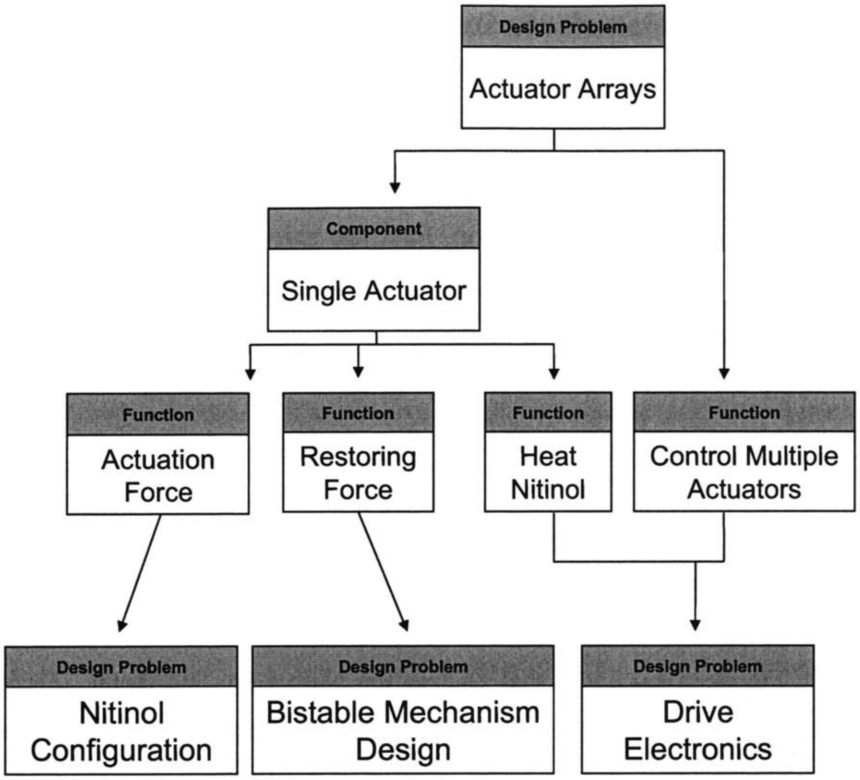


Figure 4.1 Functional Decomposition of the Actuator Array Design Problem

In an ideal world, these component problems would be independent, and solving them in any order would always produce the same solution. In practice, however, the solution of one sub-problem depends on the solution of others. Some components of the problem are especially critical in the sense that many subcomponents depend on how they are solved. It is desirable to identify these critical problem components, and to formulate design assumptions about them. Although not identified by the functional decomposition, the "economical mass production" is critical component of the problem because it is a criterion that has doomed many prior tactile displays to failure, particularly those developed in academic laboratories. Accordingly, manufacturability is added to the list of component problems and is the topic that is addressed first.

4.2 Design for Manufacturing

In order to design a product that can be mass produced economically, manufacturing costs must be addressed early in the design process. In this design two manufacturing costs are relevant: (1) parts and (2) labor. First, the individual parts (actuators) must be inexpensive to produce, and second, the cost of the time required to assemble the individual actuators (labor) must be minimal. At the outset, it was decided that the goal of minimizing the labor cost of assembly would drive the design of the individual actuator components.

The scheme for rapidly assembling the actuator arrays provides a framework for the design of the different components. The scheme starts with a thin, flexible substrate of compliant material. The substrate is cut to shape and provided with relief slits (Figure 4.2). Next, the electrical traces that address the Nitinol are laid down. Prefabricated Nitinol patches, (produced in a separate process described later), are then laminated to the top of the substrate. The substrate is then flipped over, and Nitinol patches are laminated to the bottom surface. After all of the Nitinol patches are secured to the substrate, the electrical connections between the Nitinol patches and the electrical traces are made, and then the multiplexing diodes are attached to the substrate. Once everything is assembled on the flat substrate, buckled beams are formed in the device by stamping bends into the tension beams as shown in Figure 4.2.

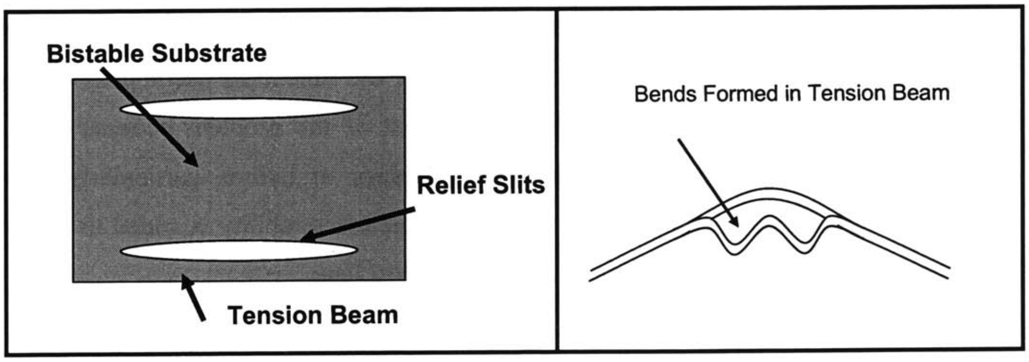


Figure 4.2 Relief slits in the bistable substrate (left) create tension beams that are formed (right) to produce the buckled beam.

4.3 Bistable Mechanism Design

The objective is to design a bistable buckled beam that snaps through in response to the martensitic transformation of the Nitinol patch. A set of parameters that fully describes a mechanism is said to be valid if it satisfies two conditions that make it compatible with the capabilities of Nitinol: (1) the beam snaps through before the Nitinol fully contracts, and (2) the Nitinol is not strained more than 5% by the beam. Meeting these two constraints will ensure that the actuator will work for millions of cycles. In addition to designing a valid mechanism, the snap-through force for the mechanism needs to be determined so it can be used later in designing the Nitinol patch. This section then covers the material selection of the bistable substrate, and the mathematical model used to predict the behavior of the mechanism.

4.3.1 Material Selection

The bistable mechanism is the foundation of the actuator, and it requires special consideration during material selection. The curvature of the beam in its stable position imposes stresses that make it susceptible to creep and stress relaxation. Fatigue is also a factor since the actuator is expected to last for millions of cycles. The last consideration in material selection is formability. The assembly scheme presented in section 4.2 calls for the buckled beam to be formed in the substrate after the Nitinol patches have been laminated to it. Therefore, the material must be formable at a temperature below the TTR if it is to be compatible with the general assembly scheme. Despite their widespread use in compliant mechanisms, this low temperature formability constraint excludes most plastics from consideration. On the other hand, most metals are easily formed at low temperatures. Metals such as steel, aluminum, and titanium generally have low susceptibility to creep and predictable fatigue lives. Because it is the least expensive, and is widely available in thicknesses ranging from 0.001" and up, steel is chosen as the material for the bistable mechanism. Specifically, a 1010 alloy of blue tempered spring steel (McMaster-Carr), with high yield strength is chosen because of its ability to maintain the formed shape even at large stresses.

4.3.2 Mathematical Model

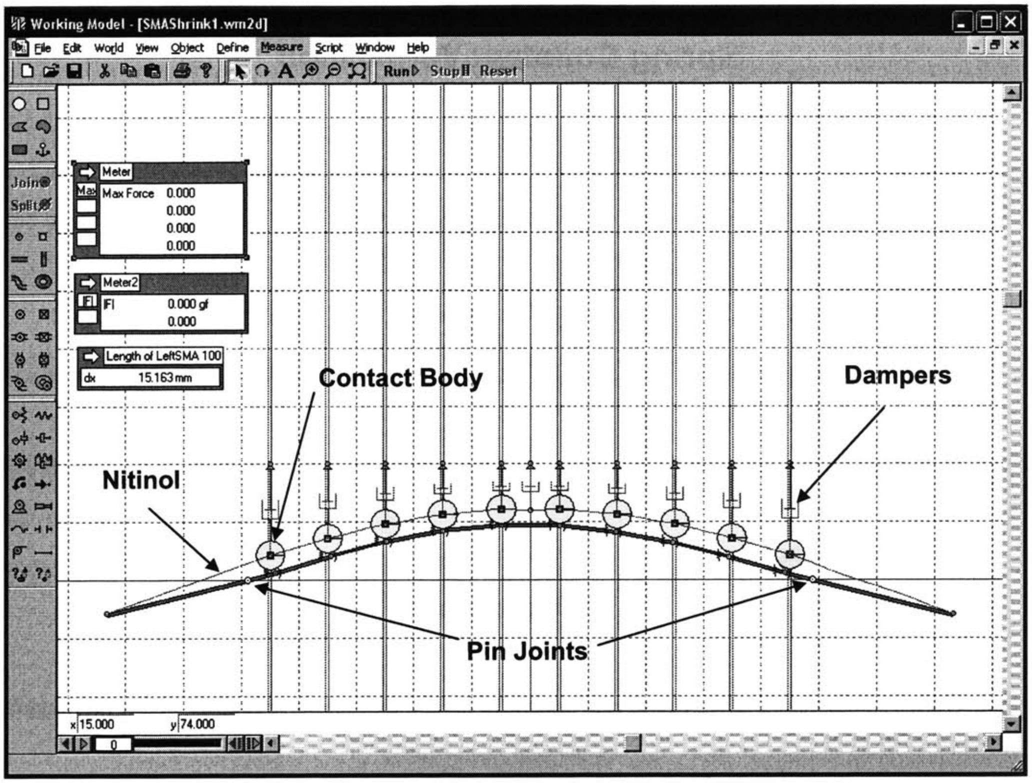
Compliant mechanisms typically experience large deflections that introduce geometric nonlinearities adding complexity to their analysis [18]. It is possible to derive a closed form analytical solution to a large deflection problem for simple geometries and loading conditions such as the centrally loaded fixed-fixed buckled beam [19]. However, increasing the complexity of geometry or loading conditions makes closed form solutions unwieldy.

The snap-through force is delivered to the bistable mechanism by the Nitinol and so it is necessary that the capabilities of the martensitic transformation are accurately reflected in the mathematical model. To ensure that the Nitinol meets the fatigue life specifications, it is critical that the bistable mechanism changes states in response to a contraction that is less than 5% of the Nitinol's original length [14]. Therefore, the length of the Nitinol must be accurately represented in the mathematical model. Nonlinearity in the problem is significant because the contact region where the Nitinol loads the beam changes as the beam deflects.

A number of commercially available software packages exist that can simulate this loading condition numerically. Working Model 2D is a two-dimensional dynamic rigid body motion simulator, and is used in this paper to handle the simulation. For this particular software package, it is convenient to model bending in a flexible beam as a series of rigid bodies connected by linear rotational springs. The value of the spring constant is determined by

$$K = \frac{EI}{L} \tag{4.3.1}$$

where E is the Young's modulus, I is the moment of inertia and L is the length of the rigid bars [20]. The boundary conditions where the mechanism is supported are modeled as pin joints as shown in Figure 4.3.


Working Model - [SMAShrink1.wm2d]
File Edit World View Object Define Measure Script Window Help
Run▷ Stop‖ Reset
Join
Split
Contact Body
Dampers
Nitinol
Pin Joints

Meter
Max Force 0.000
0.000
0.000
0.000

Meter2
|F| 0.000 gf
0.000

Length of LeftSMA 100
dx 15.163 mm

x 15.000 y 74.000

0

Figure 4.3 Screenshot of working model simulation used to design the bistable mechanism.

A script can be written in Working Model 2D to automate the task of searching the parameter space for a solution; however, it is important to validate the mathematical model before committing time to the automated search. Validating the model is accomplished by comparing the simulation results of a simply loaded buckled beam with experimentally obtained force deflection data (Figure 4.4). A sample mechanism was tested by imposing a vertical displacement to the center of the beam with a linear micropositioner. The structure that supported the beam rested on an electronic scale that measured the reaction forces. The curve corresponding to the numerical simulation falls within the scatter of experimental data, gathered from 4 trials of the same mechanism, indicating that the spring constants are accurate and the mathematical model is valid.

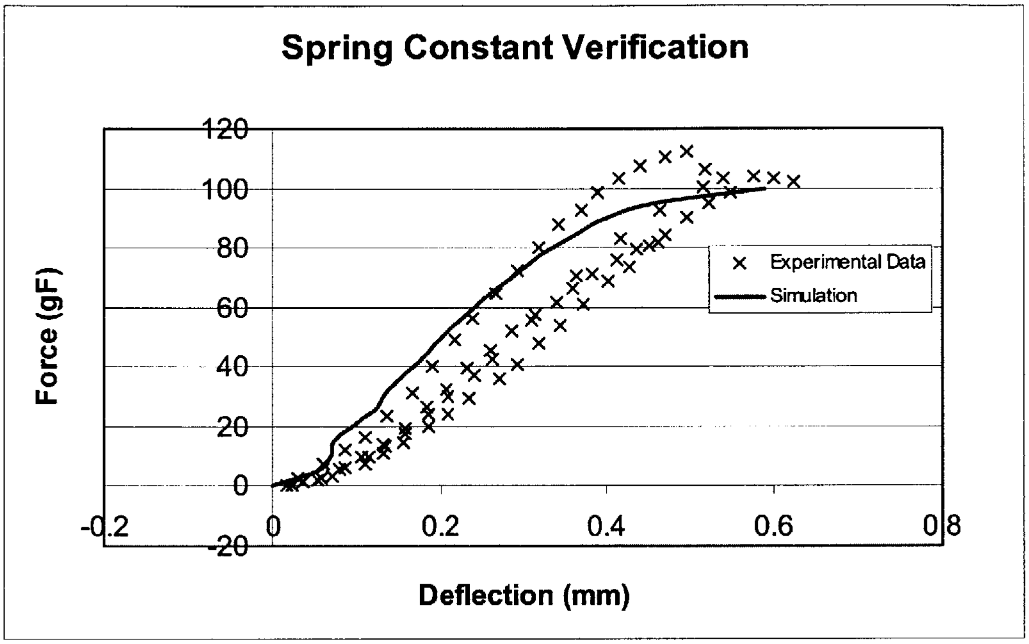


Figure 4.4 Comparison of simulation of a simply loaded bistable mechanism with experimental obtained data validates the mathematical model used for simulation.

To accurately reflect the tension in the Nitinol as a function of length, it is modeled as a rope connected to pulleys in Working Model 2D. A displacement load is applied by incrementally decreasing the length of the Nitinol. At each length, the simulation is run until equilibrium is established, at which point the tension in the rope is measured and recorded. This approach is easy to implement and can be automated with a simple script that can output the results to a file and then change the parameters of the beam and simulate again. The output data can then be used to produce plots of tension versus length that can be used to decipher whether or not a particular set of beam parameters is valid for the actuator design.

4.4 Nitinol Patch Design

The Nitinol patch must provide the actuation force that causes the bistable mechanism to change states. As such, the snap-through force determined in the bistable mechanism design is used as an assumption when designing the Nitinol patch. Therefore, it is necessary to complete the design of the bistable mechanism prior to designing the Nitinol patch. This section broadly discusses some of the general issues involved in designing the Nitinol patch. More specific details of how these issues are handled in implementation are discussed in the context of their respective prototypes in chapters 5, 6 and 7.

The primary issue in meeting the functional requirements of the Nitinol patch is to match the force output of the patch to the snap-through force of the bistable mechanism. The Flexinol™ wire used in this paper has a recommended pull force that varies with diameter (see Table 2.1). The equivalent force output of fibers laid in parallel is simply the sum of the individual pull forces. Therefore, determining the appropriate number of fibers is a simple matter of dividing the snap-through force by the recommended pull force.

There are a variety of wire diameters that are commercially available; each of which is capable of delivering the actuation force when configured in parallel as described above. Selecting the fiber diameter allows the designer to optimize another property of the wire. For this application, the diameter is selected to find a balance between maximizing peak operating frequency and minimizing the cost of the actuator.

The peak operating frequency of any SMA based actuator is governed by its ability to transfer heat. Heating the wire can be accomplished rapidly by resistive heating [14] which typically makes the cooling rate the limiting factor. Forced convection is often used to increase the cooling rate; however, it is not applicable to wearable displays. Therefore, the only way to increase the cooling rate is through the geometry of the material. This is accomplished by increasing the surface area to volume ratio in the

Nitinol. Hence, smaller diameter wires, which have larger surface area to volume ratios, cool faster, thus indicating that minimizing the diameter of the wire maximizes peak operating frequency.

Maximizing the peak frequency comes at the expense of increased per actuator cost. Smaller diameter wires are more expensive than larger diameter wires and produce smaller contraction forces. Therefore, more fibers are needed per patch to match the snap-through force, thus compounding the cost problem. Therefore, the diameter selection involves a design tradeoff based on a compromise between peak frequency and actuator cost.

In an effort to keep the actuator cost to a minimum, a scalable manufacturing scheme is developed to address the issue of arranging Nitinol fibers in parallel. In this scheme, a carriage tape backed with adhesive is mounted onto a large spool. Nitinol is then wound around the spool so that the fibers are evenly spaced on the carriage tape as shown in Figure 4.5. The carriage tape is then removed from the take-up spool by cutting (Figure 4.5). The adhesive on the carriage tape holds the Nitinol wires in place once the carriage tape is removed from the spool. This produces the long tape with parallel Nitinol fibers shown in Figure 4.5; cutting along the dotted line produces two strips of evenly spaced Nitinol patches. These patches are then ready to be assembled onto the actuator.

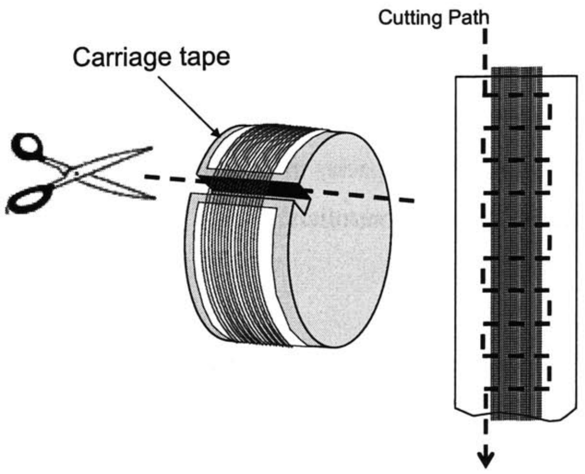


Figure 4.5 Nitinol fibers are wound onto a take-up spool at evenly spaced intervals (left); cutting along dotted line (right) reveals two strips containing Nitinol patches of evenly spaced parallel fibers.

4.5 Drive Electronics

Designing the drive electronics is a two part problem: (1) deliver power to the heating element, and (2) minimize the control lines on the actuator. Providing the power to heat the Nitinol depends on the Nitinol patch design, and the method of heating; therefore, the solution to this problem is presented later in the context of the different prototypes. This section focuses on the design of a circuit that is capable of controlling multiple actuators on a single array with a minimal number of control lines.

Microcontrollers are useful in portable applications for sending, receiving, and manipulating data; however, the number of output pins that they possess limits the number of peripheral devices with which they can communicate. Minimizing the number of control lines is equivalent to increasing the capacity of the microcontroller. This is critical in maintaining a low unit cost for large arrays of actuators because it reduces the number of relatively expensive microcontrollers needed.

Chosen for its simplicity, the BASIC Stamp 2 (BS2), which has 16 output pins, is the microcontroller used in work presented. Because the actuators are binary, and require two Nitinol patches each, direct wiring to the BS2 would occupy 2 output pins per actuator. This equates to a BS2 capacity of 8 actuators. If an array of 120 actuators is desired, it would require 15 microcontrollers.

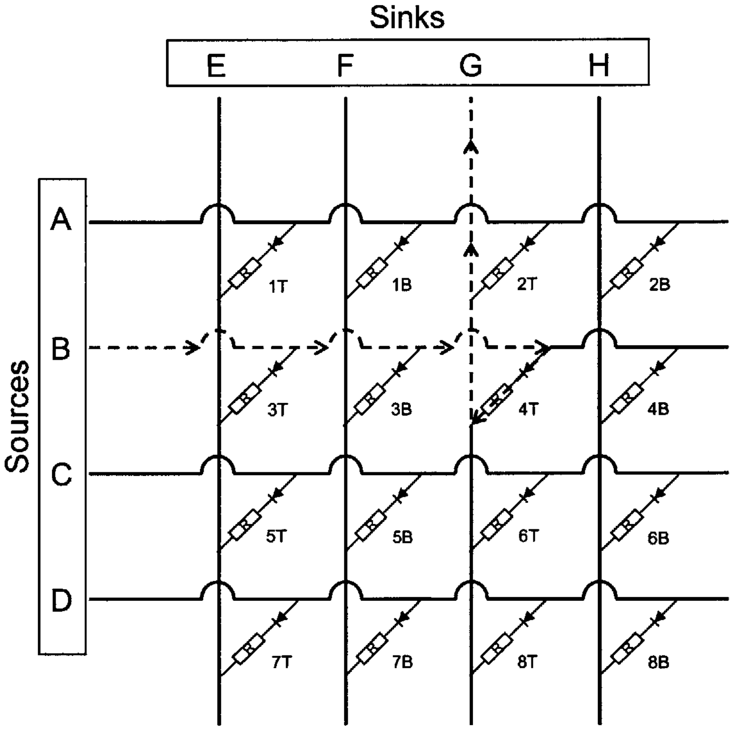


Figure 4.6 A multiplexing scheme used to limit the number of drive lines that are needed to activate actuators on the array. The dotted line shows the path of current flow when element 4T is triggered.

A multiplexing array similar to the one shown in Figure 4.6 is used to reduce the number of control lines needed. In this example, four control lines are used to source current and four more are used to sink current. A diode precedes each resistive element at the intersection of these control lines. These diodes restrict the direction of current flow so that if only one source and one sink are simultaneously on, the current can only flow through one resistive element. Therefore, triggering a specific actuator is a simple matter of turning on the appropriate source and sink. For example, if it is desired to activate the top of the fourth actuator, 4T, source line B and sink line G both need to be on, and every other control line off would need to be off. The current would then flow from B to G along the dotted line shown in Figure 4.6.

In the schematic array pictured above, the source and sink lines form a 4x4 grid capable of addressing 16 resistive elements (8 actuators). Directly wiring the eight source and sink lines to the output pins on the BS2 only uses up half of its output pins. Adding additional source and sink lines to the multiplexing array until all of the output pins are occupied creates an 8x8 grid capable of addressing 64 resistive elements (32 actuators). Therefore, this multiplexing scheme increases the BS2's capacity by a factor of four. However, an array consisting of 120 actuators would still require 4 microcontrollers.

The idea of the multiplexing circuit is based on the ability to source and sink current. The *LMD18400 Quad High-Side Driver* from National Semiconductor is an integrated circuit that contains four DMOS N-channel power transistors that are used to gate current. Applying a high digital logic to any of the four input pins opens their respective output gates, and allows current to flow through them; holding the input pins low closes the gates and blocks current from flowing [21]. This device takes up four of the BASIC Stamps output pins to provide four source lines.

The current sinks are implemented using a *TPIC2603 6-Channel Serial Interface Low-Side Driver* from Texas Instruments. This device works in much the same way as the high-side driver, with the exception that the gates drain the current directly to ground. Control information is transmitted to the TPIC2603 through a synchronous serial communication interface which requires three communication lines: (1) a data line, (2) a clock line, and (3) a chip select line [22]. This means that the BASIC Stamp uses 7 output pins to control one high-side and one low-side driver, translating into a 4x6 grid that can control 24 resistive elements (12 actuators).

At first these numbers may appear discouraging; however, the serial interface between the BS2 and the low-side driver means that an additional low-side driver only requires one additional BASIC Stamp output pin. Because low-side driver only listens to the input data signal if its chip select pin is pulled low, the data and clock lines can be common to all the low-side drivers in the multiplexing array. Therefore, each additional low-side driver, which adds 6 drain lines to the array, only occupies one additional BS2 output pin. In the example shown in Figure 4.7, one high-side and two low-side drivers occupy 8 output pins and address 48 resistive elements (24 actuators). Using up the remaining output pins adds 8 low-side drivers to the array and increases the capacity of the BS2 to 240 resistive elements (120 actuators). This is almost 4 times as many as would be possible with a simple 8x8 multiplexing grid, and is 15 times larger than using no multiplexing circuit.

It is important to note that implementing the multiplexing circuit in this way imposes a constraint on the amount of current that can be used to heat the Nitinol. The low-side driver, which is responsible for the success of the multiplexing scheme, has a maximum continuous current rating of 0.8 A [22]. Therefore, this is the absolute maximum current that can be passed through the system. The multiplexing diodes also present a possible constraint. Surface mount common anode diode arrays (two diodes per IC) suitable for mounting to the bistable substrate are available for continuous currents less than 250mA [23]. Larger current rated diodes are more expensive per IC and only contain one diode. Therefore, it is desirable to keep the current levels below 250mA to minimize the per actuator cost of the array.

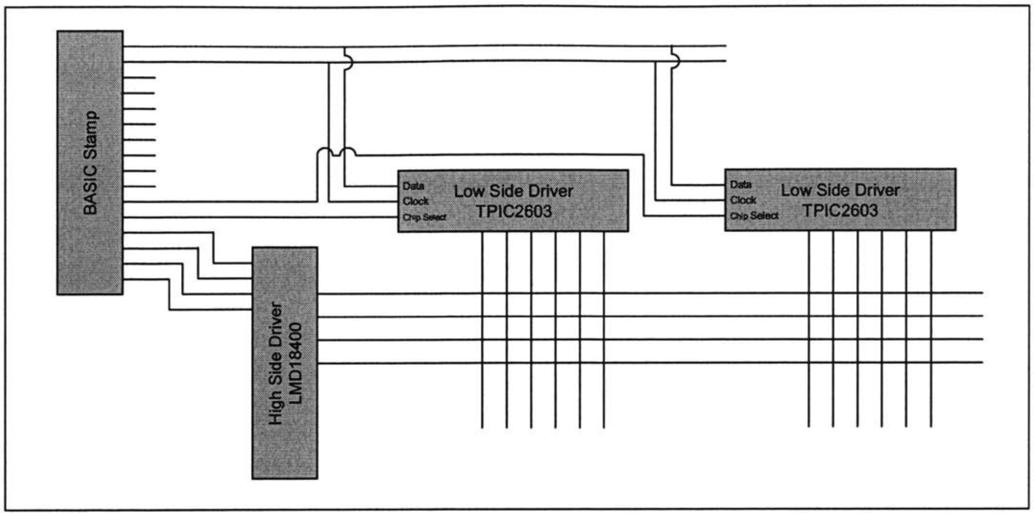


Figure 4.7 Multiplexing scheme using integrated circuits to increase the number of actuators that the BASIC Stamp can address. The configuration pictured addresses 48 resistive elements (24 actuators) with only 8 control lines.

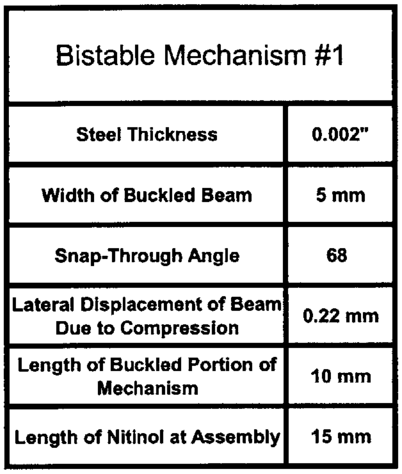
5 Prototype 1

At this point in the design process, a conceptual design has been established that utilizes a bistable mechanism to pre-strain the Nitinol. A functional decomposition has subdivided the design problems into manageable component problems, and methods for solving these component problems have been presented. The next step is to test the legitimacy of the concept through implementation. Therefore, the objective of this first prototype is to prove the validity of the individual actuator concept before committing time to developing the actuator arrays.

5.1 Bistable Substrate Design

In mass production the shape of the bistable substrate will be fabricated with progressive die stamping [24]. This procedure requires expensive tooling and is not practical when dealing with prototyping quantities. Therefore, a prototyping technique was developed using a 35W CO_2 laser to quickly cut the bistable substrate out of thin (less than 0.003") sheets of spring steel. A fundamental problem with this technique is that the heat generated by the laser alters the mechanical properties of the steel in the vicinity of the cutting path [24]. This brittle heat-affected zone limits the bending angles that can be stamped into the tension beams of the bistable substrate before they fracture. Therefore, this technique limits the snap-through angle of the actuator.

This technique is used for this prototype in an attempt to prove the validity of the concept before committing time to developing better manufacturing techniques. Therefore, a range of snap-through angles that can be physically prototyped with this technique is determined experimentally and is used to constrain the parameter space that is searched with simulation. With these constraints in place, the simulation is run and a valid geometry if found (Table 5.1). The plot of predicted tension vs. length of the Nitinol fibers for this valid configuration is shown in Figure 5.1.



Bistable Mechanism #1	
Steel Thickness	0.002"
Width of Buckled Beam	5 mm
Snap-Through Angle	68
Lateral Displacement of Beam Due to Compression	0.22 mm
Length of Buckled Portion of Mechanism	10 mm
Length of Nitinol at Assembly	15 mm

Table 5.1 Parameters that describe the bistable buckled beam of Prototype 1 as obtained from simulation.

Flexinol™ comes from the manufacturer with a 3.5% strain already induced in the wire. Therefore, the unstrained length of the wire for the bistable mechanism of Table 5.1 can be computed as

$$l_0 = \frac{l_{3.5\%}}{1+0.035} = \frac{15mm}{1.035} = 14.49mm \qquad (5.1.1)$$

where $l_{3.5}$ is the length of the Nitinol wire that makes up the Nitinol patch. At this length there is no tension in the wire which indicates that the mechanism has changed states before the Nitinol has fully contracted. Therefore, the first condition on validity is satisfied (Figure 5.1).

For the second condition to be satisfied, the Nitinol must not experience a strain of more than 5%. With the unstrained length known, the 5% strain length can be computed as

$$l_{5\%} = (1+\varepsilon)l_0 = (1.05)(14.49mm) = 15.21mm \qquad (5.1.2)$$

Figure 5.1 shows that at this length the Nitinol would have zero tension. Therefore the bistable mechanism cannot induce a 5% strain length in the Nitinol and the second condition is satisfied.

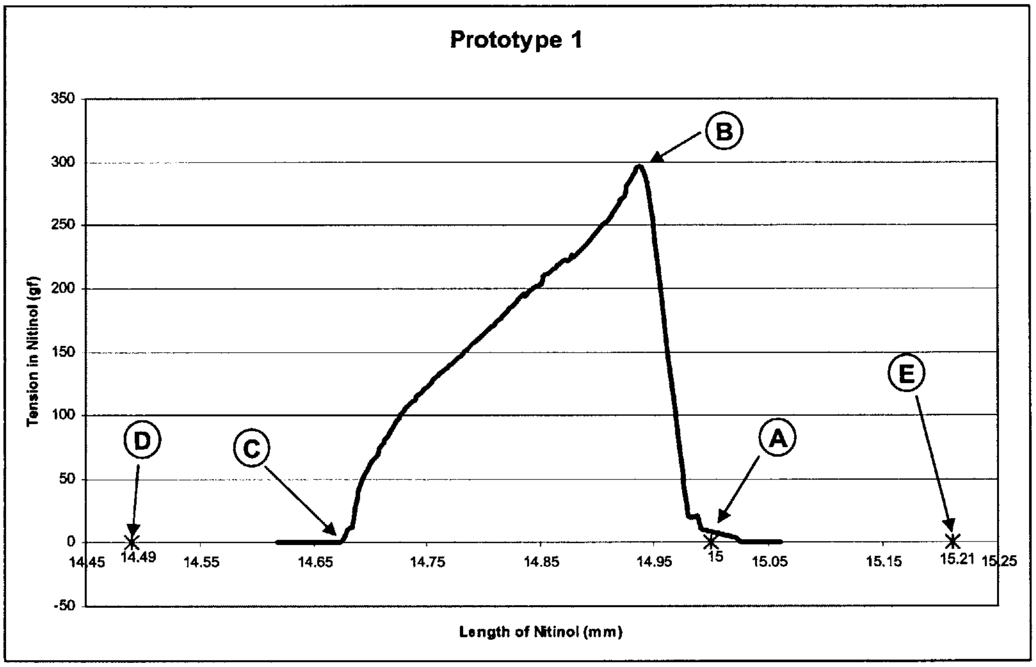


Figure 5.1 Equilibrium tensions as simulated length of the Nitinol is shortened. Approximately 14 data points did not converge to steady state, and are omitted for clarity. The length of Nitinol at assembly (A) is used to compute the unstrained length (D). The plot shows that shortening the Nitinol increases the tension until the onset of snap-through (B). Eventually, the Nitinol will go into slack (C) when the beam has passed through the unstable region. The 5% strain limit (E) cannot be induced by this mechanism since the Nitinol is in slack at this length.

5.2 Nitinol Patch Design

The snap-through force for this mechanism, as determined in simulation, is 296 grams. This is used to determine the number of fibers needed per Nitinol patch for various commercially available diameters. The results of these calculations, along with some other properties of the Nitinol patches, are shown in Table 5.2.

Nitinol Wire Diameters		Number of Fibers Needed	Parallel Equivalent Resistance	Single Wire 1 Second Heating Current	Parallel Configuration 1 Second Heating Current	Estimated Heating Time with a 0.8A Current	Series Equivalent Resistance	Prototyping Cost of Nitinol per Actuator	Bulk Cost of Nitinol per Actuator
	0.001"	38	0.7 Ω	0.02 mA	0.76 mA	0.135 sec	1010 Ω	\$5.70	\$2.71
	0.0015"	18	0.69 Ω	0.03 mA	0.54 mA	0.137 sec	223 Ω	\$1.21	\$0.43
	0.002"	9	0.79 Ω	0.05 mA	0.45 mA	0.120 sec	64 Ω	\$0.61	\$0.21
	0.003"	4	0.74 Ω	0.10 mA	0.40 mA	0.128 sec	12 Ω	\$0.27	\$0.09

Table 5.2 Properties of Nitinol patches made from different diameter wires for the bistable mechanism of Prototype 1.

A simple and common method of heating the Nitinol is to pass an electrical current directly through it. The increase in temperature in the Nitinol is then a function of the power that it dissipates. Because Nitinol behaves like a resistor, the power dissipated in the Nitinol is governed by the equation

$$P = i^2 R_{Nitinol} \tag{5.2.1}$$

The equivalent resistance of the required number of fibers for this bistable mechanism connected in parallel is less than one ohm for each of the four diameters listed in Figure 5.3. At less than 1 ohm, the equivalent resistance of the Nitinol is of the same order as the resistance of the electrical leads. Therefore, roughly the same amount of power is dissipated in both the leads and the Nitinol which makes for a bad design for two reasons. First, dissipating large amounts of power in the leads makes the circuit inefficient and significantly reduces battery life. Secondly, dissipating large amounts of power in the leads requires that they be wide enough so that they do not get damaged. This makes building arrays of actuators difficult as the amount of space needed for the leads gets too large.

A strictly parallel arrangement requires excessively large currents to heat the Nitinol. Table 5.2 shows the recommended value of current that will heat the Nitinol through its TTR in one second. Multiplying this value by the number of fibers required yields the current that will heat the Nitinol patch in one second. Increasing the peak frequency is accomplished by pulsing larger currents through the Nitinol for shorter durations. The maximum current that can be pulsed through the circuit is limited to 0.8 A by the low-side driver in chapter 4. A rough estimate of the time it would take to heat the Nitinol with a 0.8A pulse can be determined by using the recommended one second heating current as a reference.

The 0.0015" diameter wire is considered as an example of how to estimate the 0.8A pulse duration. The recommended one second heating current for Nitinol patch consisting 0.0015" diameter fibers in parallel is 540mA. The energy it takes to heat the Nitinol can then be found as follows

$$E = P\Delta t = i^2 R_{eq} \Delta t = \left(0.540^2\right)(0.69)(1) = 0.201 \text{ Joules} \qquad (5.2.2)$$

For the moment, assume that none of this energy escapes into the air. Then this energy becomes a reference for calculating the pulse length required for the maximum allowable current of 0.8A. This is done as follows

$$\Delta t = \frac{P}{i^2 R} = \frac{0.201}{\left(0.8^2\right)(0.69)} = 0.46 \text{ seconds} \qquad (5.2.3)$$

Obviously the assumption that none of the energy escapes into the air is not completely accurate; the time that it will take to heat the Nitinol will be less than this value. Even if the time it takes is 30% of this value, it will still take 0.136 seconds to heat the Nitinol. This is greater than the suggested cooling time of wire indicating that the peak frequency of this design would be limited by the rate of heating. The estimated heating times for the other Nitinol patches are shown in Table 5.2. None of the designs have acceptable heating times in a parallel configuration and so a different approach needs to be taken.

The problems of the parallel configuration stem from the fact that the equivalent resistance is too low; thus a higher resistance implementation is explored. Electrically speaking, the Nitinol fiber is nothing more than a resistor; therefore, maximizing the equivalent resistance of the Nitinol patch is accomplished by connecting the fibers electrically in series. For a Nitinol patch consisting of n fibers the ratio of series resistance to parallel resistance is

$$\frac{R_{Series}}{R_{Parallel}} = \frac{nR_{fiber}}{R_{fiber}/n} = n^2 \tag{5.2.4}$$

Therefore, the effect of using a series configuration significantly increases equivalent resistance, especially as the number of fibers increases. The series equivalent resistances for the different diameter Nitinol patches can be seen in Table 5.2; each one is much larger than the resistance of the leads resulting in most of the power being dissipated in the Nitinol. Also, all of the recommended one second heating currents are low enough that the cooling time will be the limiting factor on peak frequency.

Selecting the fiber diameter for the Nitinol patch then becomes a decision between balancing the peak frequency of the actuator and the cost of the Nitinol. The 0.001" diameter wire Nitinol patch costs more than 6 times as much as the next highest priced Nitinol patch. Therefore the 0.001" diameter wire is not considered. The other three wire diameters are more closely priced, and so the peak frequency considerations are more important. The 0.0015" diameter wire has the highest series equivalent resistance of the remaining three diameters and as such will require the smallest current to heat it in a given amount of time. Therefore, Nitinol patches for this prototype are made using the 0.0015" diameter wire.

The implementation of a Nitinol patch that addresses the fibers in series is shown in Figure 5.2. This design consists of a single strand of Nitinol and two insulating frames made from PVDF. The Nitinol wire is wound around the first PVDF frame (Figure 5.3) so that this one wire traverses the frame 18 times. This Nitinol wound PVDF frame is then laminated to a second PVDF frame for electrical insulation. This assembly of PVDF and Nitinol is then attached to both sides of the bistable substrate with an epoxy based structural adhesive and connected to the drive electronics with mechanical clips.

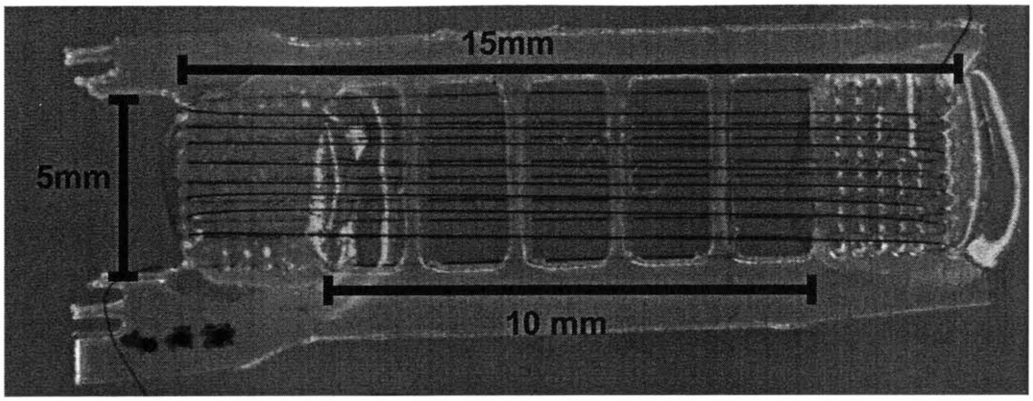


Figure 5.2 **A Nitinol patch consisting of one long strand of Nitinol wire wound around a PVDF frame.**

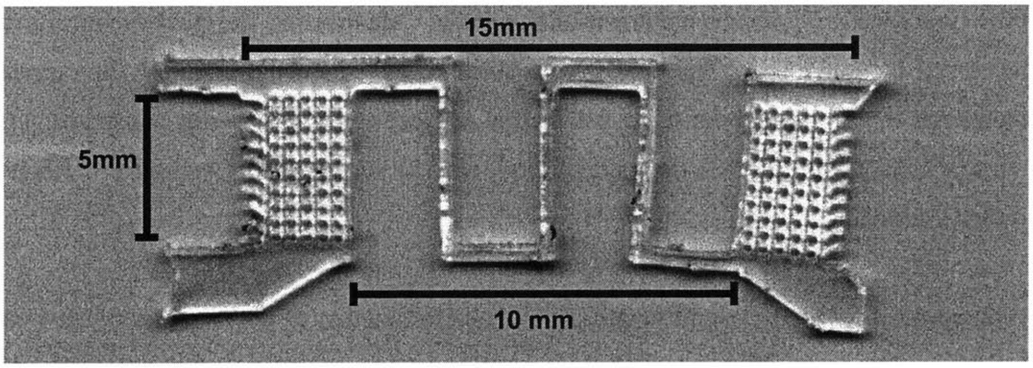


Figure 5.3 **A PVDF frame that uses a serpentine pattern to reduce its contribution to the bending stiffness of the actuator.**

5.3 Drive Electronics

The main objective of this prototype is to prove the validity of the single actuator concept. At this point, only the individual actuator is of concern, and so the multiplexing circuit is omitted from drive electronics of this prototype. However, the drive circuit is still constrained by the 0.8A maximum current rating of the low-side driver [22]. Therefore, the primary functionality of the circuit is to supply a pulse of current to the Nitinol to heat it through its transition temperature.

The appropriate amount of current and the corresponding pulse duration are unknown at this time, and will remain unknown until testing. Maximizing the peak frequency is achieved by minimizing the time that it takes to heat the Nitinol; thus, the current delivered to the Nitinol must be maximized subject to the following constrains: (1) current cannot exceed 0.8 A (low-side driver), and (2) voltage cannot exceed 36V (target specification for safety).

The power dissipated per unit length of Nitinol is a function of the diameter of the wire and the current that passes through it. Therefore, a current controlled circuit will yield more accurate power dissipation than a voltage controlled circuit. A simple way to create a constant current source is to use a transistor wired as shown in Figure 5.4 [25].

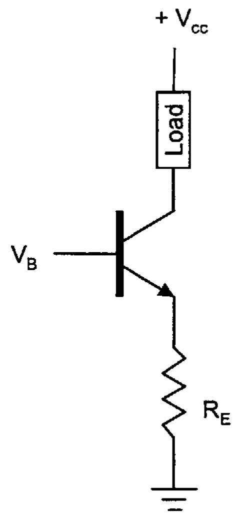


Figure 5.4 Transistor as a current source.

The peak frequency of the actuator, which is limited by the cooling rate of the Nitinol, is an important aspect of the design that is still largely unknown. Rough estimates have been made about the peak frequency based on the manufacturer's recommended cooling time, but a more thorough understanding of the cooling rate for this particular configuration of Nitinol wires is desired.

The resistance of the Nitinol wire is a function of the strain in the wire, the temperature of the wire, and the solid state phase that the Nitinol is in. Measuring resistance changes in the Nitinol at times when the strain is unchanging yields data that can be directly correlated with temperature. From this correlation, it is expected that the resistance curve of the Nitinol will exhibit an exponential decay as the Nitinol cools. Determining the exponential time constant of this resistance curve and making some simple assumptions about initial and final temperatures allows the resistance curve to be mapped into a temperature curve.

Implementing a circuit capable of measuring the resistance changes in the Nitinol is fairly straight forward. A known constant current can be passed through the Nitinol without heating it if it is sufficiently small (lmA for this circuit). This small current can be implemented with a separate transistor wired like before. Then measuring the voltage drop across the Nitinol and substituting into Ohm's law along with the known current yields the resistance of the Nitinol. The circuit is implemented so that this measured voltage across the Nitinol is output to a computer where it is then translated into resistance data.

5.4 Testing

This prototype was tested with the drive electronics described in the previous section. The pulse duration was slowly increased until the mechanism successfully snapped through from both sides with 70 ms pulses. This was an important milestone in the development of the actuator as it validated the conceptual design.

The resistance measurements taken during these experiments proved to exhibit an exponential decay as expected. Analyzing these curves resulted in time constants ranging from 0.06-0.20 seconds with an average of 0.15 seconds. Because the pulse length was slowly increased until the mechanism snapped through, it is safe to assume that the initial temperature for the cooling curve is only slightly above the TTR. A small buffer is added to this value because the Martensitic transformation really occurs over a range of temperatures. Therefore, the initial temperature is assumed to be 100°C. The final temperature is assumed to be room temperature, or 23°C; therefore, the cooling curve shown in Figure 5.5 is defined by the following equation

$$T(t) = (T_0 - T_\infty)e^{\frac{-t}{\tau}} + T(\infty) = 77e^{\frac{-t}{0.15}} + 23 \qquad (5.4.1)$$

The time it takes for the Nitinol to cool through the TTR is roughly 0.05 seconds. Assuming instantaneous heating is possible, the peak frequency limited only by the time it takes to cool the Nitinol is calculated as

$$T_{\min} = 2t_c = (2)(0.05s) = 0.1s \;\Rightarrow f_{\max} = \frac{1}{T_{\min}} = \frac{1}{0.1s} = 10Hz \qquad (5.4.2)$$

where t_c is the cooling time. From the experimentally determined heating time and the cooling data, the expected peak frequency is roughly 4Hz. Experimentally, the highest frequency that was sustained for more than 30 seconds was 1 Hz. Anything faster than this would stop working after only a few cycles. The Nitinol would continue to twitch, but the actuator would not change states.

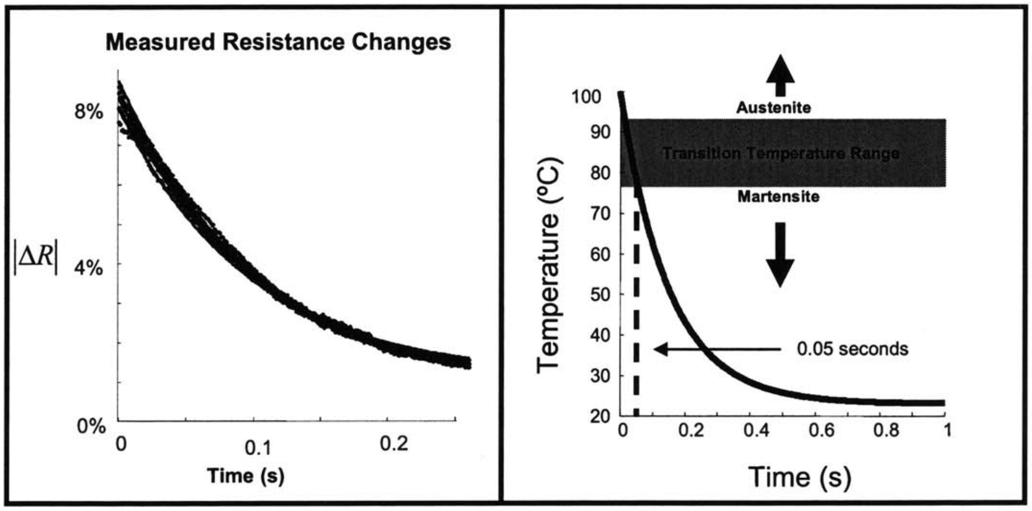


Figure 5.5 Measured resistance changes (left) due to cooling of the Nitinol after four typical pulses are superimposed (points). Exponential curves (lines) with time constants (τ) in the range of 0.06-0.20 seconds give a least-squares fit to the full data set (left). This information is mapped into a temperature curve (right) that predicts passage through the TTR in 0.05 seconds.

An explanation for the discrepancy between the computed peak frequency and the largest frequency experimentally obtained can be found in the assumptions that were made in mapping resistance to temperature. The assumption that the ambient temperature is 23°C is not exactly accurate when operating at frequencies larger than 1Hz.

When Nitinol cools, it transfers its heat into the surrounding air causing the ambient temperature to increase. When this happens, the Nitinol cools to a temperature above 23 °C. Then with the next pulse, the Nitinol is heated to a temperature above 100 °C. This then increases the ambient even more. The net effect of this cycle is an upward shift in the cooling curve. Therefore, the gap between the ambient temperature and the TTR decreases. The end result is that both sides of the actuator remain in the austenite phase, and the actuator refuses to transition between states. Because it does not account for this rise in ambient temperature, the calculated absolute peak frequency is not accurate. A more realistic estimate on the absolute peak frequency for this actuator is 5Hz and can is made by cutting the computed value in half.

Several endurance tests were conducted to gather information of the fatigue life of the actuators. In these tests, actuators were excited at various frequencies until failure. The number of cycles at failure is then easily computed from the frequency and time before failure. The longest lifetime of any of the mechanisms tested was roughly 10,000 cycles; however, the average lifetime was closer to 5,000 cycles. This is well below the target specification.

Two distinct modes of failure were observed in this prototype. The first mode occurred because Nitinol wires would occasionally get crossed creating a short path to ground. When this happens, not all of the Nitinol sees the current, and thus, not all of the Nitinol is heated above the transition temperature. Therefore, the actuation force generated by the Nitinol is drastically reduced below the level necessary to cause snap-through.

This is a minor fault in the design of the PVDF frame and could easily be fixed in later prototypes. It is mentioned because it presents an interesting feature of the drive electronics. Because the circuit is current limited, the change in resistance of the Nitinol patch due to the short has no effect on the amount of current that flows through it. Therefore, the Nitinol is not overheated or damaged by this short circuit condition. In fact, untangling the wires manually, returns the actuator to working order.

The other mode of failure observed during testing is catastrophic and prevents the further development of this prototype. In this mode, the Nitinol wire fractures due to fatigue at the tight radius where it wraps around the PVDF frame. Theoretically, the strain in the outer wall of the Nitinol fiber due to an imposed curvature can be computed as

$$\varepsilon_o = \frac{1}{(2R/d)+1} \tag{5.4.3}$$

where R is the radius of curvature, and d is the diameter of the Nitinol wire [24]. Figure 5.6 indicates that the PVDF allowed for a maximum radius of curvature of 0.0025". Substituting this value and the diameter of the wire into equation (5.4.3) reveals that the strains in Nitinol imposed by winding around the PVDF were

$$\varepsilon_o = \frac{1}{\left(2\left(0.0025/0.002\right)+1\right)} = 0.286 = 28.6\% \tag{5.4.4}$$

This value is well above the recommended limit of 5% for cyclic loading and thus explains the failure mode.

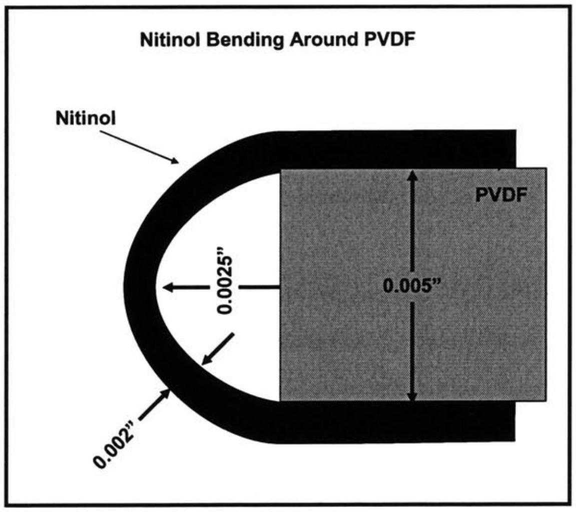


Figure 5.6 Nitinol bending around the PVDF shows a maximum radius of curvature equal to 0.0025".

6 Prototype 2

In the previous prototype, the Nitinol was wound around a PVDF frame as a means of increasing the resistance of the Nitinol patch. The tight radius of curvature where the Nitinol bent around the PVDF induced large strains that caused the Nitinol to fail in fatigue. This mode of failure can be traced back to the decision to use resistive heating to induce the martensitic transformation. If an alternative heating method is used, the Nitinol can be laid in parallel without imposing these large strains, thus eliminating the failure mode observed in Prototype 1. Because Prototype 1 proved the validity of the concept, this prototype has a two part objective: (1) develop a method of using an external mechanism to efficiently heat the Nitinol, and (2) develop manufacturing techniques for implementing actuator arrays.

6.1 Actuator Array Design

The array design developed in this paper is the series configuration, although most of the ideas would easily transfer to a parallel configuration as well. As suggested in chapter 3, the difficulty in designing an array of actuators lies in determining how to get power to the Nitinol with as few drive lines as possible. While the multiplexing circuit solves these problems schematically, implementing this circuit on the array is a task that requires some consideration.

The number of actuators per array is arbitrarily chosen for this prototype to be eight. This number is small enough to be convenient to work with during prototyping, but is also large enough to create an interesting demo. An eight actuator array requires 16 Nitinol patches and can be driven by four sources and four sinks. Therefore, eight physical source and sink lines need to run the length of the array as shown in Figure 6.1. These drive lines are made of copper, and are situated next to the steel substrate on the neutral axis of the array to minimize their contribution to the overall stiffness of the beam.

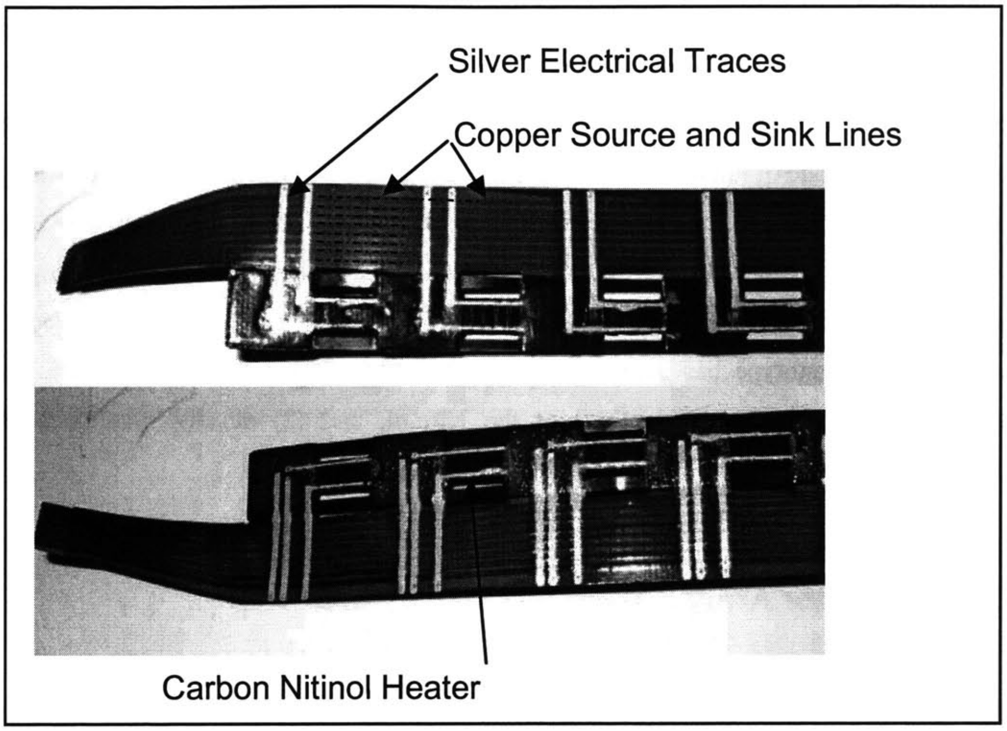


Figure 6.1 A picture of four actuators on a mechanically series 8 actuator array that uses an external carbon heater to induce the martensitic transformation. Both sides of the actuator are shown to illustrate the path that the electrical traces take to get the current out to the carbon heater.

The copper traces and steel substrate are laminated between two insulating layers of flexible polyimide. Appropriately placed vias connect the copper source lines to silver traces on the surface of the array. These silver traces route the current through a multiplexing diode and continue on to the heating elements before reaching another via that connects to the copper drain lines.

A pad printer (Imprintor; Part #B101; Oglesby, IL) is used to lay down traces of a conductive silver ink (CMI; Exp 2601-14A/B; Woburn, MA) onto the surface of the array. These traces join the copper drive lines to the individual Nitinol patches and were shown in Figure 6.1. Pad printing is low cost technique that is widely used in applications that involve printing onto non-planar surfaces.

After the silver traces are printed on both sides of the actuator, the Nitinol patches are attached to the substrate with a structural acrylic adhesive. The details of this process are discussed later. After the Nitinol patches are attached to the substrate, pads of silver ink are stenciled onto the substrate to make the electrical connections to the multiplexing diodes and to connect the electrical traces on the substrate to the electrical traces on the Nitinol patch.

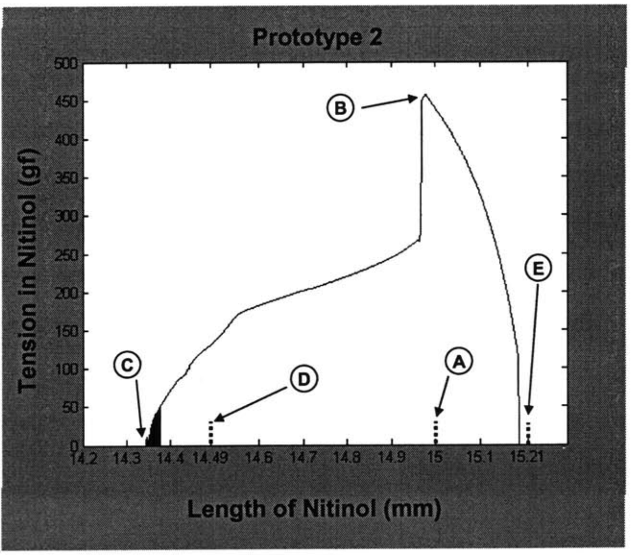
6.2 Bistable Substrate Design

In prototype 1 the technique used to cut the shape of the bistable substrate limited the amount of snap-through angle in the bistable mechanism. This was acceptable at the time because of the specific objectives of that prototype. Since the eventual mass production of the actuators will not use this limiting technique and the target specifications call for a larger snap-through angle, a new bistable mechanism is designed for prototype 2 without this constraint. Therefore a new chemical etching process is developed to produce substrates of the desired shape. Although the edges produced by this technique are somewhat jagged, the material properties are unaltered and consequently the range of snap-through angles is not limited.

With the constraint on the range of snap-through angles lifted, a new simulation is carried out to identify another set of valid parameters with a larger snap-through angle. The results of this simulation indicate a valid geometry to exist defined by the parameters in Table 6.1. The corresponding plot of tension vs. length, shown in Figure 6.2, indicates a snap-through force of 457 grams.

Bistable Mechanism #2	
Steel Thickness	0.002"
Width of Buckled Beam	5 mm
Snap-Through Angle	108
Lateral Displacement of the Beam Due to Compression	0.47 mm
Length of Buckled Portion of Mechanism	10 mm
Length of Nitinol at Assembly	15 mm

Table 6.1 A list of parameters that describe the bistable buckled used in Prototype



2.

Figure 6.2 Equilibrium tensions as simulated length of the Nitinol is shortened. The length of the Nitinol at assembly (A) is used to compute the unstrained length (D). As the length is decreased, the mechanism produces a maximum tension in the Nitinol (B), before going into slack (C). Although there is tension in the wire at the unstrained length (D), the region from (B) to (C) is unstable and so the beam will snap through if the Nitinol can supply a force greater than indicated at (B). The 5% strain limit on Nitinol (E) cannot be induced by the mechanism since the curve indicates no tension in the wire.

6.3 Nitinol Patch Design

In this design, a Nitinol is patch is made up of individual Nitinol fibers laid in parallel on a small patch of material. The fibers are then heated by some external source through their transition temperature causing them to contract. A good thermal coupling between the Nitinol fibers and the heat source is required this idea to work. The design problem is then a two part problem involving the design of a small external heater, and figuring out how to achieve a good thermal coupling between the heater and the Nitinol.

In theory, the external heater makes for a robust design. Adding an external heater means that even if a single Nitinol fiber breaks, the heater is unaffected, and all of the Nitinol will still contract. Of course, the force generated by the broken fiber will not be able to contribute to the actuation force, but the force from all of the other fibers will contribute. To ensure that the remaining fibers provide an adequate actuation force, the Nitinol patch is designed with more fibers than is necessary. This introduces a factor of safety in terms of actuation force without affecting the drive electronics.

As with the previous prototype, the first step in designing the Nitinol patch is to determine the number of fibers required to match the snap-through force of the bistable mechanism. The results of these computations are shown in Table 6.2. Once again, choosing a fiber diameter is a balance between cost and peak frequency. However, this time the peak frequency is not dependent on the electrical properties of the wire diameters. Because the Nitinol will be heated by an external heater, the rate at which heat is put into the Nitinol is only a function of the temperature of the heater and the thermal coupling between the heater and the Nitinol. Therefore, the decision of which Nitinol diameter to use reduces to a balance between cost and cooling rate. Table 6.2 shows that 0.002" diameter wire is a reasonable choice. To introduce a factor of safety of 1.5 on the actuation force, 20 fibers of this diameter are used to make up the Nitinol patch instead of the minimum value of 15 indicated in Table 6.2.

		Number of Fibers Needed	Parallel Equivalent Resistance	Single Wire 1 Second Heating Current	Parallel Configuration 1 Second Heating Current	Series Equivalent Resistance	Prototyping Cost of Nitinol per Actuator	Bulk Cost of Nitinol per Actuator	Cooling Time
Nitinol Wire Diameters	0.001"	65	0.41 Ω	0.02 A	1.3 A	1727 Ω	$9.76	$4.21	0.06 sec
	0.0015"	30	0.41 Ω	0.03 A	0.9 A	372 Ω	$2.02	$0.64	0.09 sec
	0.002"	15	0.47 Ω	0.05 A	0.75 A	106 Ω	$1.01	$0.33	0.10 sec
	0.003"	7	0.42 Ω	0.10 A	0.7 A	21 Ω	$0.47	$0.14	0.20 sec

Table 6.2 Properties of Nitinol patch with different diameters based on bistable mechanism for Prototype 2.

As we saw in prototype 1, passing a current through a resistor is a good method of generating heat. Since this prototype uses an external heater, it is not constrained by the electrical properties of the Nitinol, and so resistive heating is still a good choice. In this case, the resistance value of the heater can be constructed by design to be compatible with the drive electronics.

The first step in designing the carbon patch is to determine the target resistance. This target resistance is affected by the constraints on the system. To reduce the cost of the multiplexing diodes, common anode diode arrays are used; this imposes a maximum current constraint of 250 mA on the circuit. The maximum voltage rating of 25V for the low-side driver is the source of the other constraint that influences the target resistance.

In the prototype of chapter 5, the average power required to heat the Nitinol through its transition temperature was

$$P = i^2 R_{Nitinol} = \left(0.100^2\right)\left(220\right) = 2.2 \text{ Watts} \tag{6.3.1}$$

This number can then be used as a reference in determining the target resistance of the carbon patch. The middle curves in both plots in Figure 6.3 are constant power curves representing the 2.2W reference power used to heat the Nitinol of prototype 1. The bottom plot can be used to determine the resistance needed to dissipate the 2.2W for a given voltage.

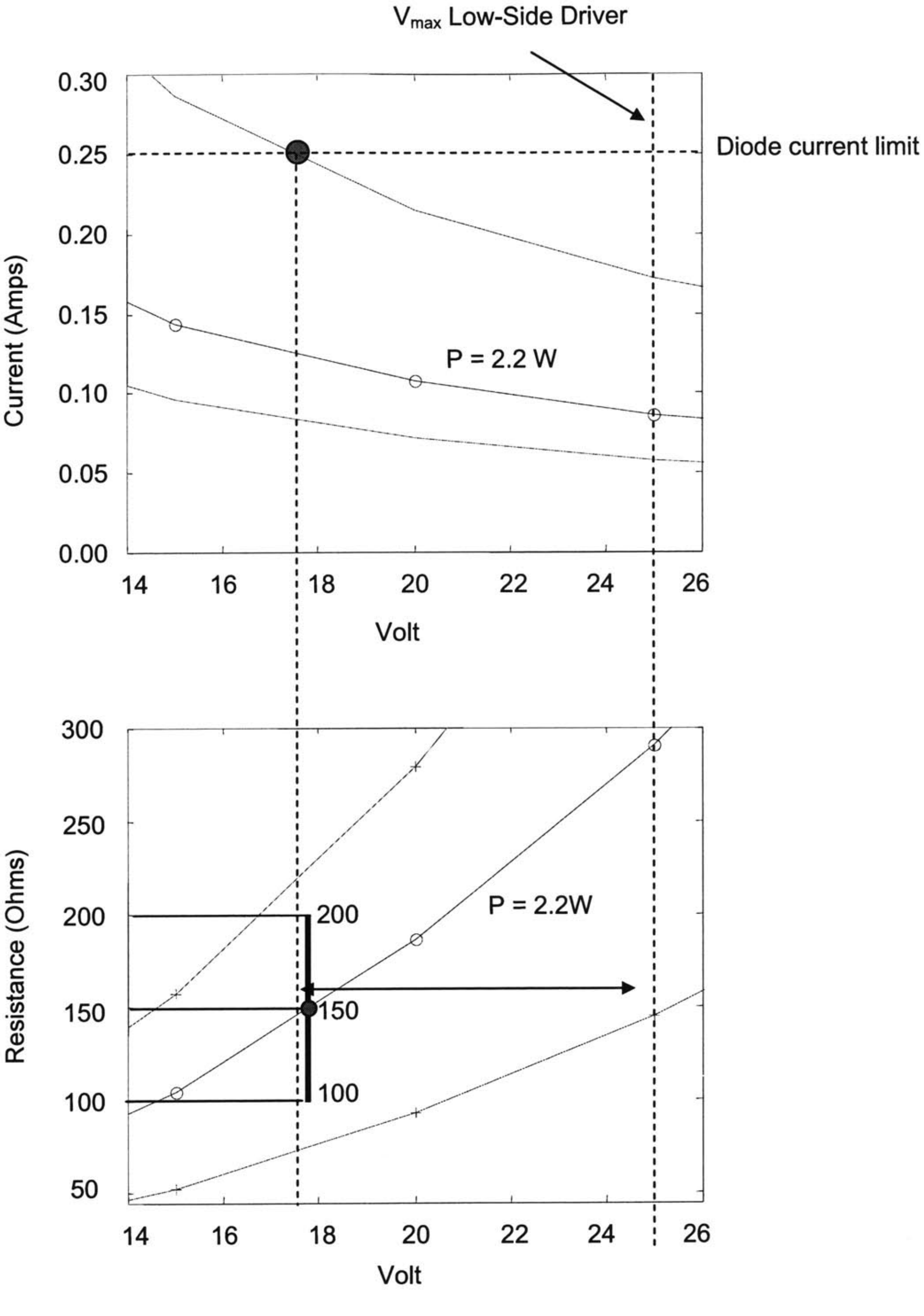


Figure 6.3 Constant power curves used to determine the target resistance of the external carbon heaters.

The supplier of the carbon ink that is used to make the carbon patch advises that the ink is reliable to +/- 50% of the target resistance (CMI; EXP 2599-41; Woburn, MA). The other two curves in each plot represent this variance in the manufacturing of the carbon heaters. The top plot then shows that the curve corresponding to the -50% variation intersects the maximum current limit of the multiplexing diode at approximately 18V. Following the dotted vertical line down to the point where it intersects the 2.2 W curve in the bottom plot reveals that a target resistance of 150Ω. The bar in the lower plot indicates a range of +/- 50 ohms which is the target range for the carbon patches. Although this is a smaller range than the supplier advertises, it should be attainable with refined stenciling techniques.

Looking back to the top plot in Figure 6.3, it is clear that staying inside the +/- 50% curves and above 18V will ensure that the multiplexing diodes are safe. This can be accomplished with the target range of 100-200 ohms by staying between 18-21V. Of course, if more power needs to be delivered to the heater, there is some room in the upper corner of the current plot that is outside the envelope of the variation in the resistance. If this is the case, these calculations will need to be revisited to ensure that the current does not exceed the 250mA limit.

With the target resistance in mind, designing the carbon patch to match this target resistance is a simple matter solving a few equations. The sheet resistivity of the carbon ink is 150 Ω/sq./mil., which means that a square patch of carbon 1 mil thick has a resistance of 150 Ω. The area spanned by the Nitinol is a rectangle that has a length of 4 mm and a width of 10 mm in the direction that the current will be passed. Therefore, a carbon patch 1 mil thick that covers this area has a resistance of

$$R = R_S\left(\frac{L}{W}\right) = 150\left(\frac{4}{10}\right) = 60\Omega \qquad (6.3.2)$$

where R_S is the sheet resistivity and L and W are the length and width respectively. This resistance is lower than the target value and would result in current that exceeds the 250 mA limit; hence, a design correction must be made.

One method of correcting the design is to realize that the carbon ink does not need to cover the entire Nitinol footprint, but instead can be spread out in stripes as shown in Figure 6.4. With this type of design, the Nitinol is heated where it contacts the carbon and relies on conduction distribute the heat throughout wire. A successful design then constructs the ratio of length to width to produce the target resistance. This ratio is found to be

$$\frac{L}{W}=\frac{R}{R_S}=\frac{150}{150}=1 \tag{6.3.3}$$

The other approach is to modify the thickness of the carbon patch. Reducing the thickness to 0.4 mils increases the resistance to 150 Ω. This can be accomplished by changing the thickness of the stencil.

The carbon patch is made from a compliant carbon ink and is stenciled onto a polyimide tape to make the heater. The carbon ink needs to cure before it is effective. During this curing process, all of the solvent that is in the ink is removed through evaporation. This evaporation causes a significant reduction in volume of about 40%. The thickness of the resistor is reduced accordingly, which leads to a higher resistance than the stencil height alone would predict. Therefore, the exact thickness of the stencil needed should be determined experimentally.

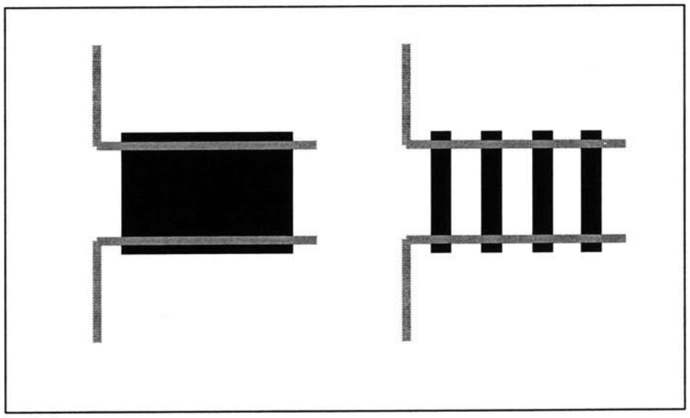


Figure 6.4 Two different carbon heater designs: (left) a carbon patch covers the span of the bistable mechanism; (right) a striped pattern is used to maintain the required length to width ratio.

In chapter 4, a general scheme is presented for creating Nitinol patches consisting of parallel Nitinol fibers. For this prototype, a winding station was built based on this scheme for producing Nitinol patches and is pictured in Figure 6.5. The pulleys are positioned to help guide the Nitinol from the stock spool to the take-up spool. One of the pulleys sits on top of a linear stage that is used to position the fiber on the spool. By controlling the speed of the linear stage relative to the take-up spool, the spacing between fibers is controlled. This results in a take-up spool with equally spaced parallel Nitinol fibers.

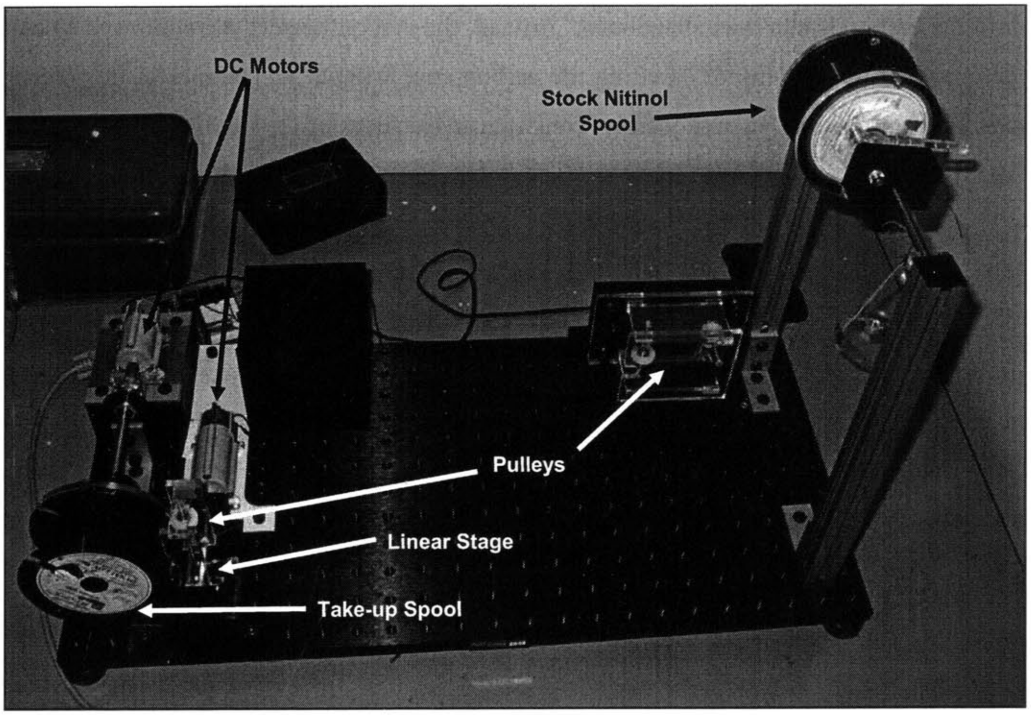


Figure 6.5 Picture of Nitinol winding equipment used to make prototypes two and three.

The preferred embodiment of the design is to embed the fibers into the carbon heater to ensure a good thermal coupling between the external carbon heater and the Nitinol fibers. This is done by winding the Nitinol onto the take-up spool while the carbon on the carriage tape is still wet. The take-up spool is then removed from the winding station and placed in an oven to cure at 70 °C.

In practice, embedding the fibers into the carbon is not as straightforward as described above. The reduction in volume of the carbon ink as it cures causes the ink to pull away from the Nitinol as the carbon hardens. This causes small cracks to form in the carbon heater which significantly increases the resistance of the carbon patch as well as the manufacturing variability. This procedure for embedding the Nitinol does not produce consistent resistances and therefore would result in inconsistent heating.

Because of the material problems mentioned above, the idea of embedding the Nitinol into the carbon heater was abandoned. Instead, the carbon heaters were allowed to cure before winding the Nitinol. Letting the carbon cure without the Nitinol in it, prevents cracks from forming. Consequently, consistent resistances of the predicted value can be attained.

The idea is then to embed the Nitinol in a thin layer of silicone adhesive stenciled on top of the carbon patch. The purpose of the silicone is two fold. First, it is used to secure the fibers in place until they are attached to the bistable substrate. Second, the silicone is used to improve the thermal coupling between the Nitinol and the carbon heater. It is important to keep the silicone layer thin so as not to increase the thermal mass of the actuator.

6.4 Anchoring the Nitinol

In the first prototype the issue of attaching the Nitinol patch to the bistable substrate was fairly straight forward. The tiny holes at either end of the PVDF frame gave the epoxy a large surface for adhesion. Since this prototype consists of 18 individual fibers instead of only one long strand wrapped around a frame, care must be taken to ensure that all of the fibers get anchored to the substrate with a structural adhesive.

This problem is one of selecting a material and a footprint so that the adhesive does not fail under the anticipated loading conditions. The loading conditions imply three different failure modes. These modes are: failure in shear along individual Nitinol wires, failure in shear along the bistable substrate, and failure due to peeling near the polyimide patch.

During assembly, the Nitinol must stay below the TTR; therefore, adhesives that require curing at elevated temperatures are excluded from consideration. The choices then reduce to either epoxy or acrylic based structural adhesives. The polyimide insulation layer that makes up the surface of the bistable substrate has poor adhesion characteristics. Therefore, a hole is cut in the polyimide layer to expose a patch of steel on which the adhesive can bond.

The load is determined by the number of fibers in the actuator and the recommended pull force of each fiber. This load is then used with the footprint dimensions of the interface to calculate the stresses in the adhesive. The results of these calculations are shown in Table 6.3 along with the material properties of the acrylic (3M, DP-805) and epoxy (3M, DP-110) adhesives. Both systems have similar tensile-shear strengths; however, the peel strength of acrylic is 10 fold better than that of epoxy. The factor of safety for the peel load when using epoxy is less than 1 therefore the acrylic adhesive is selected.

Loads		
Peel	8.9	lb/in
Shear Stress at Nitinol Interface	59	psi
Shear Stress at Steel Interface	44.5	psi

	Peel		Shear at Nitinol Fiber		Shear at Steel	
	Peel Strength (lb/in)	Factor of Safety	Shear Strength (psi)	Factor of Safety	Shear Strength (psi)	Factor of Safety
Acrylic	35	3.9	3,700	62.7	3,700	83.1
Epoxy	3	0.3	2,200	37.3	2,200	49.4

Table 6.3 Nitinol Anchoring Adhesive Selection Table

6.5 Testing

Because the resistance of the external heater is different than the series resistance of Prototype 1, a new drive circuit was built to test Prototype 2. Although the design of arrays of actuators was addressed in this prototype, the drive electronics were not yet built to handle multiplexing. Instead, the test circuit is a simple RC circuit in which a relay, in response to a computer controlled input signal, controls when the capacitors charge or discharge. This simple circuit allows for the voltage to be adjusted as well as the pulse duration for individual test actuators.

Testing of this prototype had limited success. The length of the pulse used to heat the Nitinol through its TTR for this actuator was typically close to 0.5 seconds. Obviously this is not good in terms of peak frequency, and it indicates that the thermal coupling between the heater and the Nitinol was not efficient.

Another problem with this actuator was that it failed to work from both sides. Although the Test actuators that were made with Nitinol patches on only one side would snap through without hesitation, two sided actuators would not work both ways. There are two noticeably different things between prototype 1 and prototype 2, each of which could be responsible for this behavior.

The obvious difference is the external heat source. This dramatically increased the time it took to heat the Nitinol through the TTR. Consequently, a possible explanation for the failure is that the increased time it took to heat the Nitinol allowed for heat to be transferred to the other side of the actuator as well. This would then cause the Nitinol on both sides to be in the austenite phase and would prevent the mechanism from reaching the other stable position. Therefore, it was hypothesized that increasing the rate at which the Nitinol is heated would allow the actuator to work both ways. A simple experiment was conceived to test this explanation.

In this experiment, all of the constraints imposed by the electrical circuits are thrown away. A 20 fiber Nitinol patch is then constructed, and the fibers are connected at each end with an electrically conductive ink to address them in parallel. Then the RC pulsing circuit can be used to deliver large currents in short amounts of time. This theory was quickly proven false when three different actuators of this construction all failed to work in both directions when tested.

The other difference between Prototype 1 and Prototype 2 is that more Nitinol fibers were used than were absolutely necessary in Prototype 2. This was done to ensure that the actuation force would be greater than the snap-through force. A second failure theory is then formulated based on the idea that the extra Nitinol fibers make the Nitinol patch stiffer. It is then hypothesized that the bistable mechanism is not strong enough to provide the restoring force needed to pre-strain this stiffer Nitinol patch. Therefore, if the exact number of fibers necessary to overcome the bistable mechanism, as computed in the simulation, is used instead of the number that includes a factor of safety, the bistable mechanism will be able to provide the appropriate restoring force.

Another simple experiment was conceived to test this theory. In this experiment actuators with 15 fibers addressed in parallel were pulsed with the RC drive circuit at large currents. Due to an accidental manufacturing error, the Nitinol patches were constructed with only 12 fibers each. Nonetheless these actuators were tested to see if they would snap through. Surprisingly enough they snapped through both ways with no problems. This confirmed the suspicion that the extra Nitinol had been too stiff for the bistable mechanism to stretch out. Because the restoring force of the Nitinol reaches zero before the Nitinol reaches its 5% strain length, using 12 fibers instead of 15 does not pose a problem for fatigue life.

7 Prototype 3

Although prototype 2 did not function as well as prototype 1, manufacturing techniques were successfully developed for assembling the series array. In developing these techniques, a method for increasing the resistance of the Nitinol patch was developed that does not induce any strains in the Nitinol. The objective of prototype 3 is to develop functioning actuator arrays based on this new manufacturing method that can be controlled by a portable drive circuit.

7.1 Nitinol Patch Design

In the first prototype it was noted that the equivalent resistance obtained by wiring all of the fibers in parallel was much too low to be implemented. This motivated the use of a series arrangement in which the resistance of the Nitinol was maximized. This worked for heating the Nitinol, but the only way to implement it introduced excessive strains in the Nitinol. These strains eventually caused catastrophic failure due to fatigue.

The second prototype tried to get around this problem by introducing an external heater. This way the Nitinol patches could be made without introducing the large strains that caused failure in the first prototype. This version had problems with the thermal coupling that kept the peak frequency of the design under 1Hz.

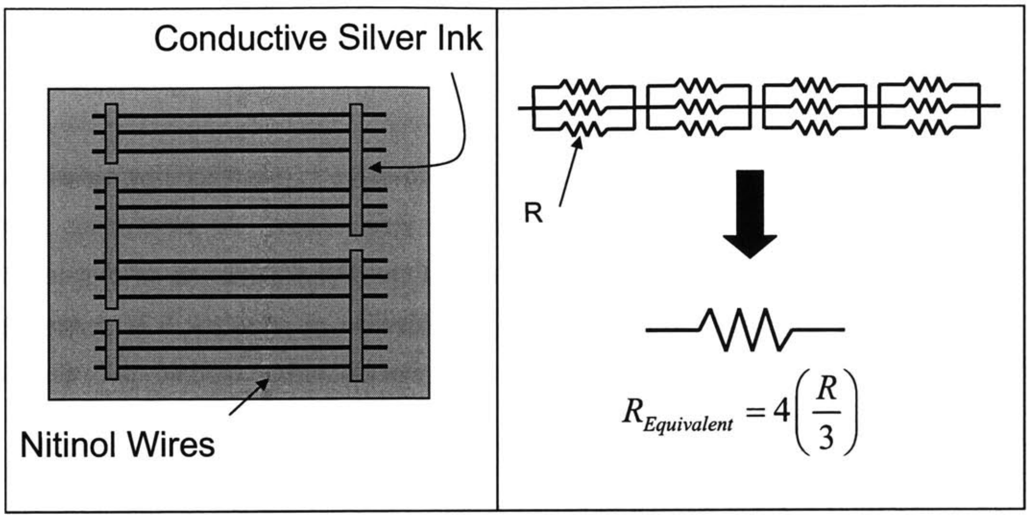


Figure 7.1 A Nitinol patch for Prototype 3 (left) uses silver ink to connect adjacent fibers into parallel groupings. These groupings are connected in a pattern that increases the equivalent resistance of the Nitinol patch (right).

The idea behind the third prototype is to keep the good aspects of the first two prototypes while eliminating the problems that they had. Heating the Nitinol by passing a current through it proved to be a successful method in prototype 1. Therefore prototype 3 returns to this method. The pitfalls of introducing large strains in the Nitinol by winding it around the PVDF frame are avoided by constructing the Nitinol patch as shown schematically in Figure 7.1. This configuration consists of groups of three Nitinol fibers addressed in parallel with a pad of silver ink. These groups of parallel fibers are then connected in series to their adjacent groups of parallel fibers. Since each fiber can be thought of as a 7.1 Ω resistor, this configuration can be represented by the resistor network shown on the right in Figure 7.1 and thus has an equivalent resistance of 9.4Ω.

In this configuration the fibers all lie flat like they did in prototype 2, but they will be heated by passing a direct current through them like they were in prototype 1. The energy required to heat the Nitinol in prototype 1 can be calculated as

$$E = P\Delta t = i^2 R_{Nitinol}\Delta t = (0.100A)^2 (223\Omega)(0.070s) = 0.189 Joules \qquad (7.1.1)$$

Figure 7.2 uses this energy as a reference to estimate the amount of current required to heat the Nitinol as a function of the pulse duration. The totally parallel configuration requires more than the limit of 0.8A, imposed by the multiplexing circuit in chapter 4, to heat the Nitinol in less than 300ms. However, the curve corresponding to the configuration shown in Figure 7.1 with an equivalent resistance of 9.4Ω shows that it can be heated with a current of 0.8A in 50ms. Allowing the Nitinol to cool for the full 0.1s recommended cooling time before actuating the other side translates into an estimated peak frequency of 3.3 Hz.

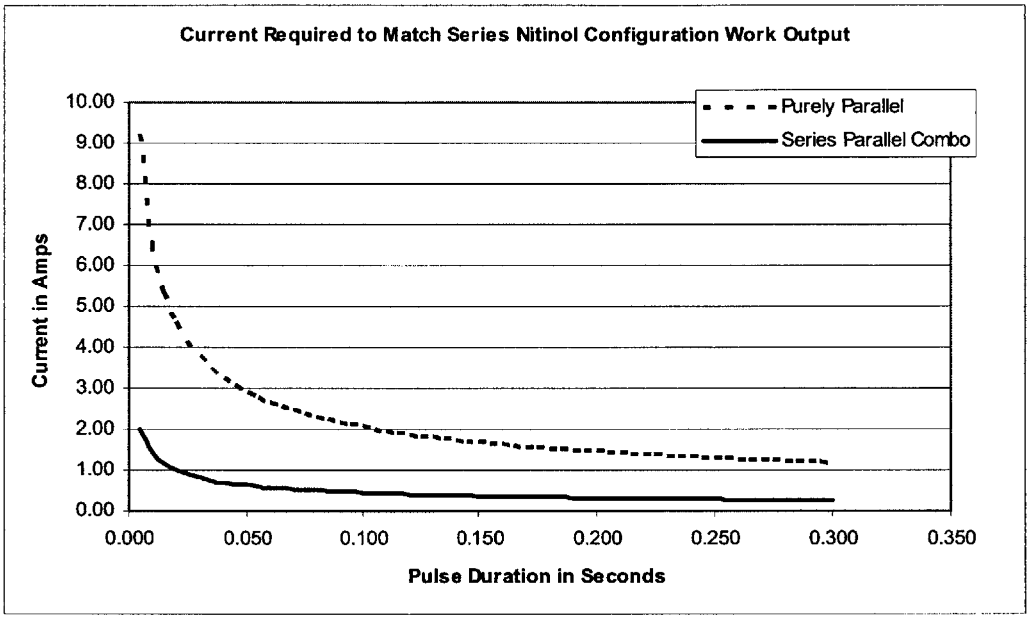


Figure 7.2 The current required to supply 0.2 Joules of energy to the Nitinol in both a purely parallel configuration and a configuration of 4 groups of 3 parallel fibers

This type of configuration was first conceived during the development of Prototype 1 when trying to figure out how to increase the resistance of the Nitinol patch without altering the manufacturing scheme proposed in chapter 4. At that time, however, implementing this type of a Nitinol configuration was thought to be impossible on a large scale. Therefore the idea was put aside and the method of Prototype 1 was pursued.

During the development of the second prototype, a method was needed for getting the Nitinol fibers to extend past the edge of the polyimide patch so that they could be attached to the bistable substrate with a structural adhesive. The solution to this problem was to pre-cut holes in the polyimide tape in the locations where the Nitinol would hang over the edge of the polyimide. It was this breakthrough idea of pre-cutting a pattern in the polyimide tape that led to the technique that enables the production of the Nitinol configuration shown in Figure 7.1

Before this break through idea, this configuration could not be made because the fibers were so close together that there was no way to stencil the silver ink over the Nitinol fibers without some ink running under the stencil and causing an accidental short circuit. However, if a pattern has been pre-cut in the polyimide so that there is no material for the ink to adhere then there is no chance of an accidental short circuit forming.

The procedure for manufacturing these Nitinol patches then follows. First, holes are cut in the polyimide carriage tape in the pattern shown in Figure 7.3. Then, a layer of silicone adhesive, which is used to hold the Nitinol in place, is stenciled onto the carriage tape. This silicone is used to hold the Nitinol wires in place until the conductive ink has been stenciled. The carriage tape is then mounted in the take-up spool and the Nitinol wire is wound around the take-up spool. Then the take-up spool can be removed from the winding station and set aside to dry. Once dry, the carriage tape can be cut free of the take-up spool and the silver ink can then be stenciled over the carriage tape as shown in Figure 7.3 while the ink is still wet, the carriage tape is then lifted up and only the silver ink that was not over the holes remains on the carriage tape. The carriage tape is then cut into Nitinol patches in the same way it was for prototype 2.

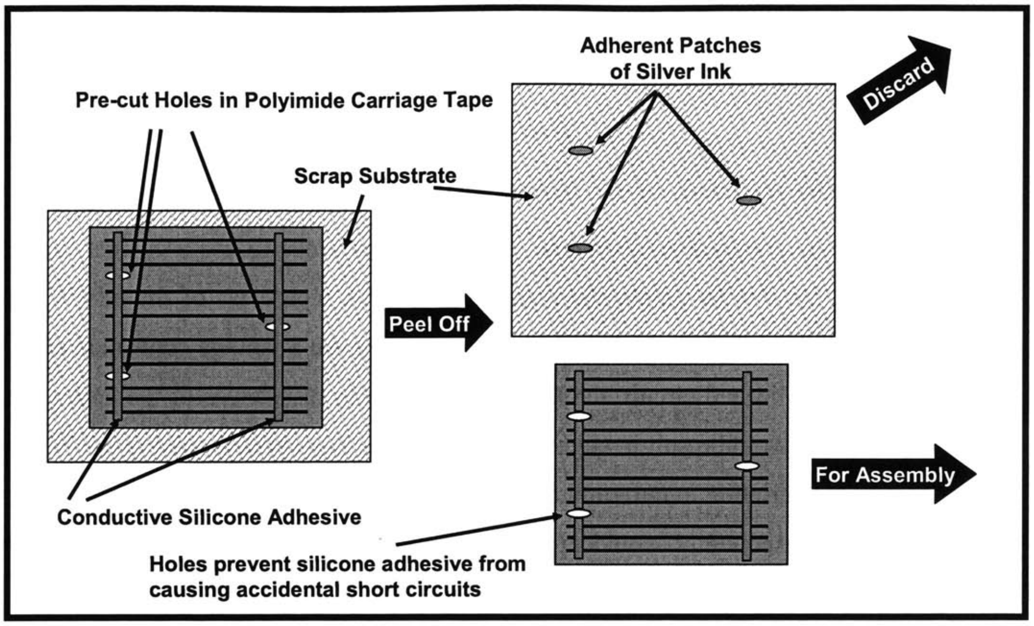

Adherent Patches of Silver Ink
Discard
Pre-cut Holes in Polyimide Carriage Tape
Scrap Substrate
Peel Off
For Assembly
Conductive Silicone Adhesive
Holes prevent silicone adhesive from causing accidental short circuits

Figure 7.3 An electrically conductive adhesive is stenciled over a Nitinol patch with pre-cut holes (left). These holes allow for the conductive adhesive to bond to the scrap substrate. Then, peeling the carriage tape off of the scrap substrate reveals a Nitinol patch (bottom right) that can be consistently produced.

7.2 Drive Electronics

The drive electronics that power this prototype need to not only supply a pulse of current but also needs to include the multiplexing capabilities as well. In chapter 4, the multiplexing circuit was presented, and components were selected that had a maximum current rating of 0.8A.

The resistance of the Nitinol fibers is a function of the resistivity of the Nitinol and the geometry of the wire and as such is fairly consistent, and therefore, the resistance of the Nitinol patch will be fairly consistent. Due to differences in the lengths of leads addressing different actuators and the variability in the resistances of electrical connections made with the conductive inks, the resistance of the entire path from source to sink will have some variance. The purpose of delivering a pulse of current to the Nitinol is to heat it by dissipating power in form of

$$P = i^2 R_{Nitinol} \tag{7.2.1}$$

where $R_{Nitinol}$ is the resistance of the Nitinol in the Nitinol patch excluding any added resistance from the conductive silver ink. It then makes sense to use a current limited system that will be able to supply the maximum of 0.8A regardless of the path's total resistance from source to sink. If the drive circuit were to use a voltage controlled circuit instead of a current limited circuit, the slight differences in resistances of the entire path from source to sink would result in slight variations in the current delivered to the Nitinol patch. This slight variation in current would have large effects on the power dissipated in the Nitinol and consequently would create large differences in the temperature of the Nitinol.

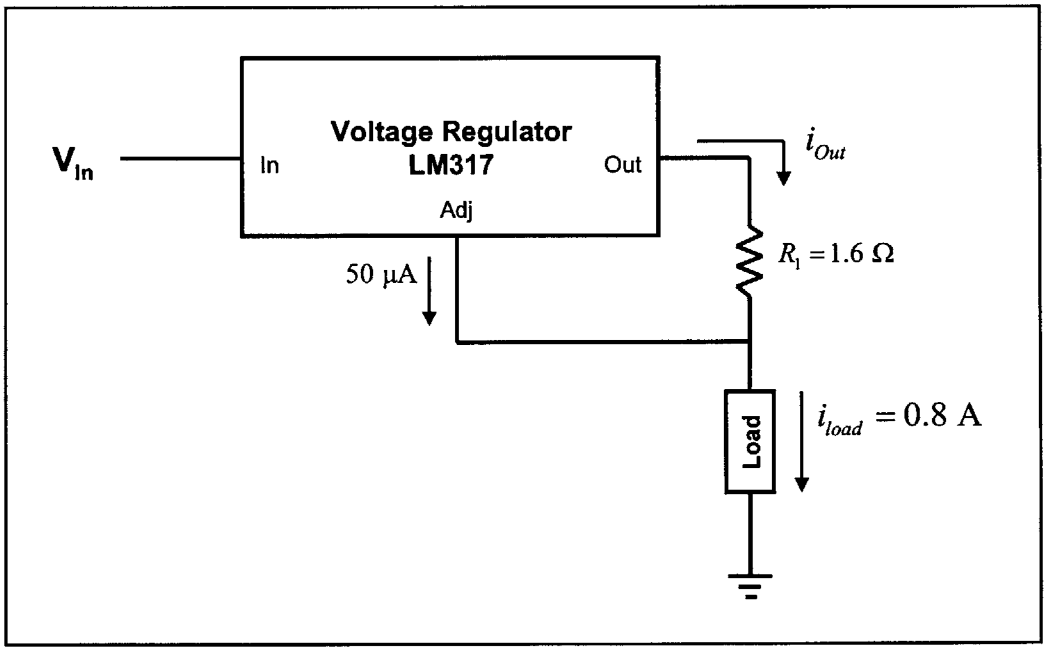


Figure 7.4 Schematic showing how a voltage regulator can be wired as a two terminal current limiter.

Voltage regulators like the popular LM317 can be wired as a two terminal current limiter by connecting a resistor from the V_{Out} pin to the Adjust pin as shown in Figure 7.4. The job of the voltage regulator is to maintain a 1.25V drop across the output and adjust pins, and so

$$V_{Out} - V_{Adj} = 1.25\ Volts \tag{7.2.2}$$

A negligible amount of current (50-100μA) flows into or out of the adjust pin making the current through the load equal to the current through R_1 (+/- 100μA). This current can then be determined from the voltage drop across R_1 as follows

$$i_{Out} = \frac{V_{Out} - V_{Adj}}{R_1} \tag{7.2.3}$$

Substituting equation (7.2.2) into equation (7.2.3) and rearranging gives

$$R_1 = \frac{1.25}{i_{Out}} = \frac{1.25 \text{ Volts}}{0.8 \text{ Amps}} = 1.56\ \Omega \tag{7.2.4}$$

Of course this is not a standard value resistor, and so the next largest resistor is used which in this case is 1.6Ω. The LM317 voltage regulator wired in this way will guarantee that a 0.8A pulse of current is delivered to the Nitinol as long as

$$V_{In} \geq V_{Out_{Max}} = 1.25 + i_L R_{L_{Max}} \tag{7.2.5}$$

where $R_{L_{Max}}$ is the maximum resistance of the entire path from source to sink.

The job of communicating to the low and high side drivers is the job of the central processor (BS2). The BASIC Stamp can be interfaced with a computer program so that a human user can program when the different actuators should be activated. The BASIC Stamp then communicates these signals to the multiplexing circuit, and remembers the state of the actuators. Because the Basic stamp is programmable, and is what is responsible for controlling when the pulse is delivered to the actuators, it can also be responsible for controlling how long the pulse lasts. So, while the pulse of current is limited to 0.8A by the components of the circuit, we can control the duration of the pulse quite accurately.

7.3 Testing

Testing of this prototype involved three different phases: testing the individual actuator, testing the multiplexing circuit, and testing the actuator arrays.

The expected resistance of the Nitinol patch of this prototype is 9.5 Ω; however, initial resistance measurements for the individual actuator were much larger (~25,000 Ω). Despite the large resistance, this actuator was tested using the pulsing circuit of Prototype 2. Surprisingly, the actuator successfully snapped through from both sides with a 17 V pulse that lasted 70 ms. When the resistance was measured again after pulsing the Nitinol, it had reduced to 40 Ω, which means that the current that was pulsed to the Nitinol was approximately 400 mA.

Several more actuators were tested to verify that the initial large resistance was not simply a faulty measurement. Although there was a fairly large spread in the "after-pulse" resistance (10-50 Ω), each actuator tested demonstrated a significant decrease in resistance associated with being pulsed. Conversations with the silver adhesive supplier indicate that curing the silver adhesive at 70 °C instead of 175°C reduces the conductivity of the ink, which explains the initial large resistance. A likely, but untested explanation of the reduction in resistance is that the heat generated from pulsing a current through this large resistance has the effect of curing at elevated temperatures; thus, increasing the conductivity of the adhesive, and reducing the resistance of the Nitinol patch. This reduction in resistance is not predictable and is a material problem that will need to be addressed in the future.

Despite the above mentioned material properties, the single actuator was quite successful. Nearly all actuators tested with the pulsing circuit worked from both sides with pulse durations ranging from 0.03 ms to 0.07 ms. There were a few mechanisms that only worked from one side, but the explanation can also be linked to the material problems of the conductive adhesive. On these particular actuators, the side that did not snap through would twitch a little at lower voltages, and then would burn up in a puff of smoke when the voltage was slowly increased. Because of the low conductance of the adhesive, the current flowing through the groups of parallel fibers might not get distributed evenly between the fibers. Therefore, the fibers that see more of the current would heat faster than the other fibers. If so, the small twitch that was observed was the result of only a few fibers on the patch contracting. The power is dissipated as $P = i^2R$; therefore small changes in current, due to increasing the voltage, have a large effect on power dissipation and the associated temperature increase melts the fibers. This failure mode is not due to the design, but due to quality control in manufacturing related to the material problem described above.

The drive electronics were tested with an LED mock-up of the actuator array to make sure that the multiplexing scheme worked as planned. With a single LED array, the circuit behaves as expected; however, when multiple LED arrays are connected some current flows through LED's that were not actuated. This does not appear to be a problem with the multiplexing scheme, but rather a problem with the high side driver leaking current. Due to time constraints on the project, this problem with the drive electronics has not been resolved. Instead, actuator array testing was conducted with only one array connected to the circuit.

In testing the actuator arrays, only 6 of 64 (9%) actuators would snap-through when triggered by the multiplexed drive circuit. None of these actuators worked from both sides, and those that did required pulse durations of roughly 150 ms. Like the individual actuators, problems with the silver ink conductance produced large Nitinol patch resistances. This means that the drive circuit probably was not delivering the 0.8A to the Nitinol because doing so would require a supply voltage greater than the 24V circuit maximum. Because large currents could not be delivered to the Nitinol, the conductive adhesive on the actuator arrays did not demonstrate as significant a reduction in resistance as the single actuator did. Therefore, the failure of the actuator array seems to be directly linked to the material problems of the conductive adhesive.

8 Conclusions

There is an absence of affordable wearable tactile displays in the consumer electronics market. The work presented in this research paper is part of a larger ongoing project to fill this void by developing portable wearable tactile displays based on alternative actuation technologies. This thesis focuses on developing a thin, flexible, lightweight actuator using shape memory alloy technology that can be used to transmit forces directly to the skin. The scope of this paper spans the first stage of the development of the actuator starting at the conceptual design and ending with several early stage prototypes. This chapter highlights the intermediate problems that have been solved and identifies the issues that still need to be addressed in the next stage of development.

A novel conceptual design has been established that uses a bistable mechanism to provide the restoring force that strains the martensitic Nitinol. This bistable mechanism is critical in overcoming the fundamental properties of Nitinol that otherwise make it unsuitable for portable tactile applications. To implement this concept, a method was established in which numerical simulation determined the parameters that make up a valid bistable buckled beam. This method makes it easy change the dimensions of the beam for specific applications by automating the process of identifying valid beam geometries.

The individual actuator alone is not enough to make an interesting tactile display, and so the conceptual design includes methods for interconnecting the individual actuators into larger arrays of many actuators. To implement these large arrays, a multiplexing circuit was designed that increases the BS2's capacity from 8 to 120 actuators. Considering the fact that the microcontroller is the most expensive part of the array, this increased capacity represents a large reduction in cost per actuator.

In order to bring the conceptual design to life, prototyping techniques were developed and specialized equipment was fabricated to enable the small scale production of components that would typically be produced in volume by large scale manufacturing procedures that require expensive tooling. These techniques and equipment made it possible to change the design in response to experimental observations without suffering from delays associated with lead time. These advances in prototyping are as important to the development of the actuators as advances in the design itself.

Despite three different implementations, an effective method for heating the Nitinol in this specific application has not yet been found. The equivalent resistance of an electrically parallel configuration of Nitinol fibers is so low that heating the Nitinol by passing a current directly through it would require currents larger than allowed by the multiplexing drive circuit. The next stage of development needs to focus on solving this problem if the vision of the conceptual design is ever to be realized.

The idea of increasing the equivalent resistance of the Nitinol patch as described in Prototype 3 shows potential for addressing the issue of the low equivalent resistance of parallel Nitinol fibers. Calculations show that this implementation should have worked, and experiments indicate that it may still be possible; however, material problems with the conductive adhesive limited our ability to judge the success of the actuator. The conductive adhesive was not as conductive as advertised and so the currents that were needed to heat the Nitinol were larger than the current controlled drive electronics could supply. Future work should investigate different conductive adhesives to try and improve the electrical connections so that the success of the concept can be more accurately assessed.

The reason that addressing the Nitinol fibers in parallel is a problem is due to the constraint that the multiplexing circuit imposes on current. If this constraint can be relaxed so that the Nitinol can be addressed in parallel, the actuator design would be greatly simplified. Therefore, the multiplexing circuit should be examined to see if there is any way to increase the current limit of the circuit. The main reason that this limit was in place was to reduce the cost of the actuator array. Therefore, a financial analysis should be done to see if the added cost associated with the increased current rating is really that much more than the added manufacturing cost associated with the increased complexity of the design needed to work at lower currents.

If neither of the above mentioned areas of future consideration produce a viable actuator, a third course of action is recommended in which the actuator of Prototype 2 is revisited. The concept of Prototype 2, in which an external heater is used to heat the Nitinol still holds promise. Problems with the carbon ink heating element prevented the Nitinol fibers from being embedded in the heater. Consequently, the thermal coupling between the heater and the Nitinol was not good enough to heat the Nitinol quickly. Further work should develop better embedding techniques with different heating materials to improve the thermal coupling between the heater and the Nitinol fibers.

There is an absence of tactile displays in the everyday life of the average American. The conceptual design presented in this paper, if realized through further development, would represent a significant advancement toward filling that void. The possible applications of a device as thin, flexible, and affordable as the one presented in this paper would find use in applications ranging from video games to massage chairs to silent communication devices and beyond.

Works Cited

[1] C. Leary, "Parts for TSAS NP3 Tactile Vests," FedBizOpps (FBO), 1999.

[2] A. H. Rupert, F. E. Guedry, and M. F. Reschke, "The use of a tactile interface to convey position and motion perceptions," presented at AGARD 541 Virtual Interfaces: Research and Applications, Lisbon, 1993.

[3] H. Z. Tan, "Information transmission with a multifinger tactual display," *Perception & Psychophysics*, vol. 61, pp. 993-1008, 1999.

[4] D. G. Caldwell and N. Tsagarakis, "Integrated Haptic Feedback," presented at IEEE Virtual Reality 2000, New Brunswick, NJ, 2000.

[5] A. B. Vallbo and R. S. Johansson, "The tactile sensory innervation of the glabrous skin of the human hand," in *Active Touch*, G. Gordon, Ed. New York: Pergamon Press, 1978, pp. 29-54.

[6] J. M. Hollerbach, I. W. Hunter, and J. Ballantyne, "A comparative analysis of actuator technologies for robotics," in *The Robotics Review 2*, O. Ktatib, J. J. Craig, and T. Lozaon-Perez, Eds. Cambridge, MA: MIT Press, 1992, pp. 299-342.

[7] R. Fletcher, "Force transduction materials for human-machine technology interfaces," *IBM Systems Journal*, vol. 35, pp. 630-638, 1996.

[8] J. Biggs and M. A. Srinivasan, "Haptic Interfaces," in *Handbook of Virtual Environments*, K. Stanney, Ed. London: Lawrence Earlbaum, Inc., 2002, pp. 93-116.

[9] P. M. Taylor, A. Moser, and A. Creed, "A sixty-four element tactile display using shape memory alloy wires," *Displays*, vol. 18, pp. 163-168, 1998.

[10] T. W. Duerig, K. N. Melton, D. Stockel, and C. Wayman, *Engineering Aspects of Shape Memory Alloys*. Boston: Butterworth--Heinemann, 1990.

[11] D. E. Hodgson, M. H. Wu, and R. J. BierMann, "Shape Memory Alloys," vol. 2005: SMA Inc., 2003.

[12] G. B. Kauffman and I. Mayo, "The Story of Nitinol: The Serendipitous Discovery of the Memory Metal and Its Applications," *The Chemical Educator*, vol. 2, 1996.

[13] J. Walker, "The Amateur Scientist: Wire that "remembers" its shape is put to work running an engine.," *Scientific American*, vol. 254, pp. 124-127, 1986.

[14] "Technical Characteristics of Flexinol Actuator Wires," Dynalloy, Inc., 3194-A Airport Loop Drive Costa Mesa, CA 92626-3405.

[15] R. Dunlop and A. C. Garcia, "A Nitinol Wire Actuated Stewart Platform," presented at Australasian Conference on Robotics and Automation, Auckland, 2002.

[16] "Proper Length of Flexinol Wire is Device Specific," Dynalloy, Inc.
1998.

[17] T. Starner and Y. Maguire, "Heat Dissipation in wearable computers adided by thermal coupling with the user," *Mobile Networks and Applications*, vol. 4, pp. 3-13, 1999.

[18] L. L. Howell, *Compliant Mechanisms*. New York: John Wiley & Sons, Inc., 2001.

[19] M. Vangbo, "An analytical analysis of a compressed bistable buckled beam," *Sensors and Actuators*, vol. A, pp. 212-216, 1998.

[20] P. Mitiguy and A. K. Banerjee, "Determination of Spring Constants for Modeling Flexible Beams." Working Model Technical Paper, 2000.

[21] "LMD18400 Quad High Side Driver," National Semiconductor 2004.

[22] "TPIC2603 6-Channel Serial Interface Low-Side Driver," Texas Instruments, Dallas 1996.

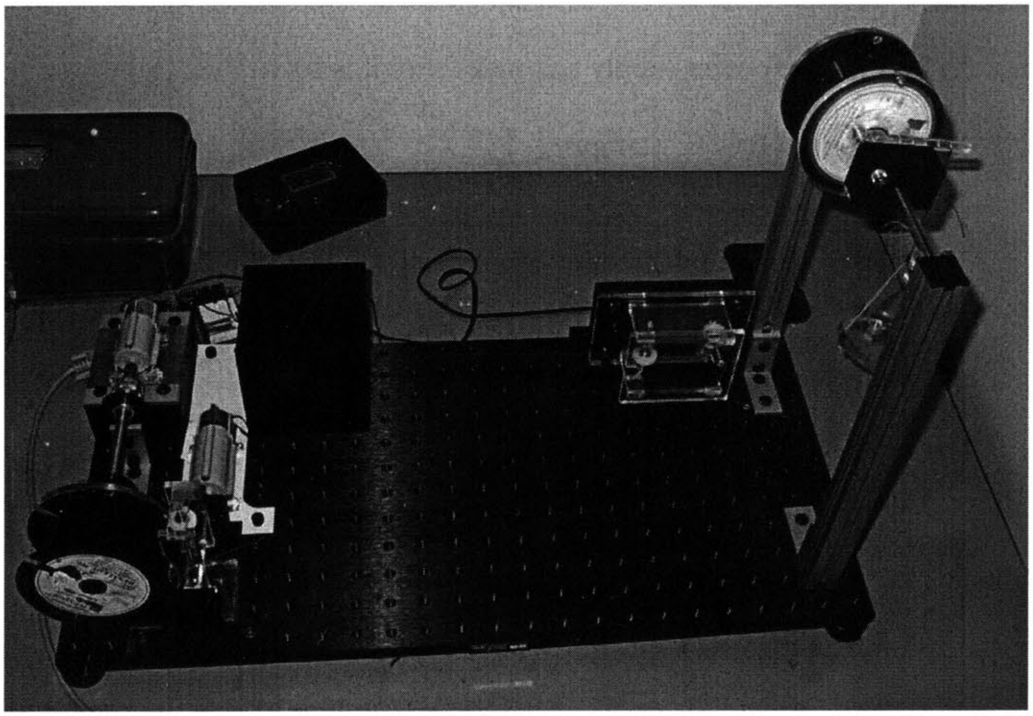
[23] "BAW56WT1 Dual Switching Diode," ON Semiconductor 2001.

[24] S. Kalpakjian, *Manufacturing Processes for Engineering Materials*, Third ed. Reading: Addison Welsley Longman, 1997.

[25] P. Horowitz and W. Hill, *The Art of Electronics*, Second ed. New York: Cambridge University Press, 1994.

Appendix A: Nitinol Winding Station

Nitinol Winding Station



User's Manual

Winding Procedure

1.) Remove the take-up spool from the winding station by unscrewing the bolts that hold it in place. When removing the take-up spool, make sure the notch is lined up with the corner of the right sensor housing as shown in the figure.

2.) Secure the carriage tape to the take-up spool with a small piece of tape and mount the take-up spool back onto the winding station. Make sure to line up the notch in the take-up spool with the notch in the take-up spool mount.

3.) Turn on the power supply and make sure it is set to 9V.

4.) Turn on the BASIC stamp.

5.) Turn on the Winding Station. The on and off positions are indicated in the picture below, but are not marked on the actual unit.

6.) Put the toggle switch in the reverse direction (pointing left) and allow the linear stage to return to its fully left position.

7.) Check to make sure that the Nitinol is situated in the pulleys grooves. Feed the Nitinol through the notch in the take-up spool and the take-up spool mount and tape it to the outside of the take-up spool mount.

8.) Put the toggle switch in the forward position (pointing to the right) to start winding the Nitinol onto the take-up spool.

NOTE: The spool will continue winding until it the take up stage hits the sensor on the right side. Flipping the toggle to the left DOES NOT stop the motors. To stop the motors before the stage triggers the sensor the Winding station must be turned off.

9.) When the appropriate number of fibers has been wound onto the take-up spool, turn the winding station off with the take-up spool notch in a roughly vertical position. This allows easier access to the keyway.

10.) Place a piece of scotch tape over the Nitinol to secure the Nitinol to the carriage tape. The tape should be positioned above the notch in the take-up spool.

11.) Cut the Nitinol strand in near the locator pulley and tape the free end to the outside of the locator pulley's frame.

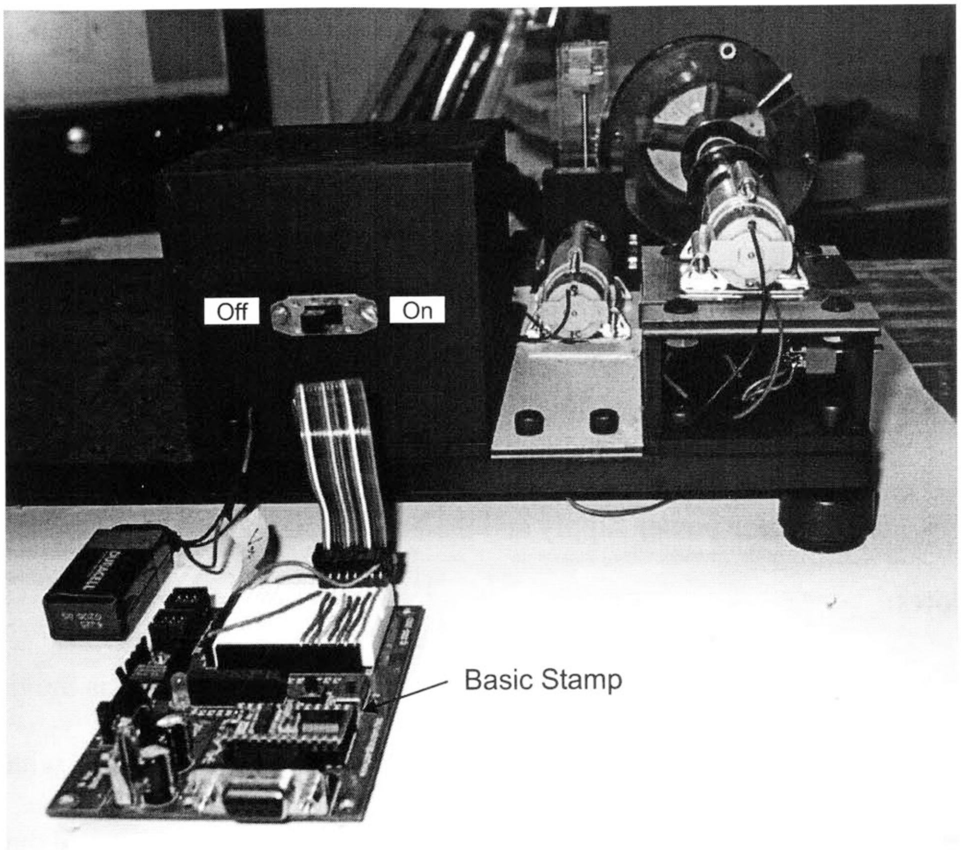
12.) The take-up spool can then be removed from the winding station and placed in an oven to cure if needed.

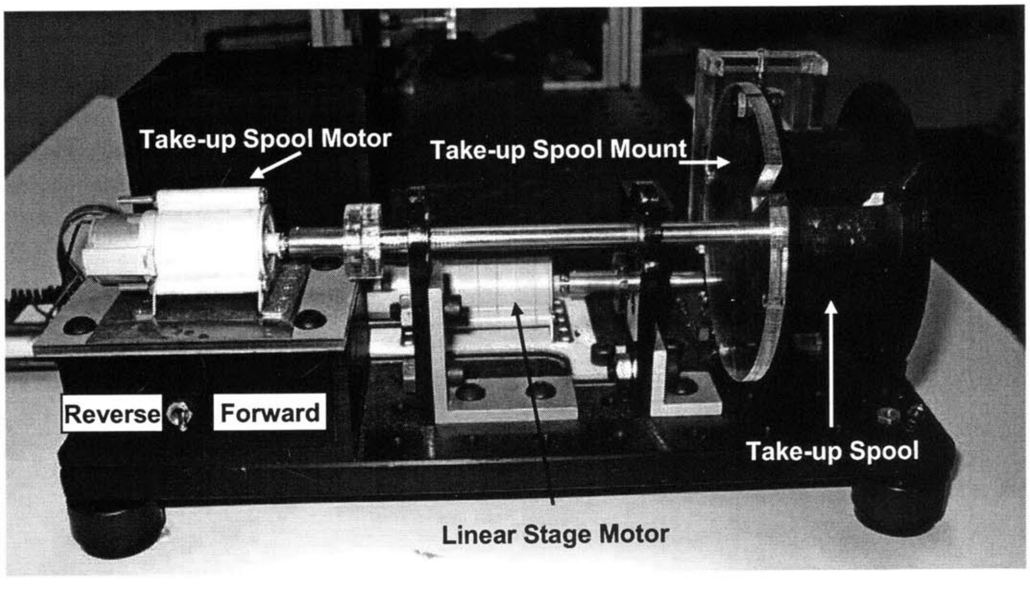
13.) The carriage tape can then be cut from the take-up spool with scissors through the notch in the spool.

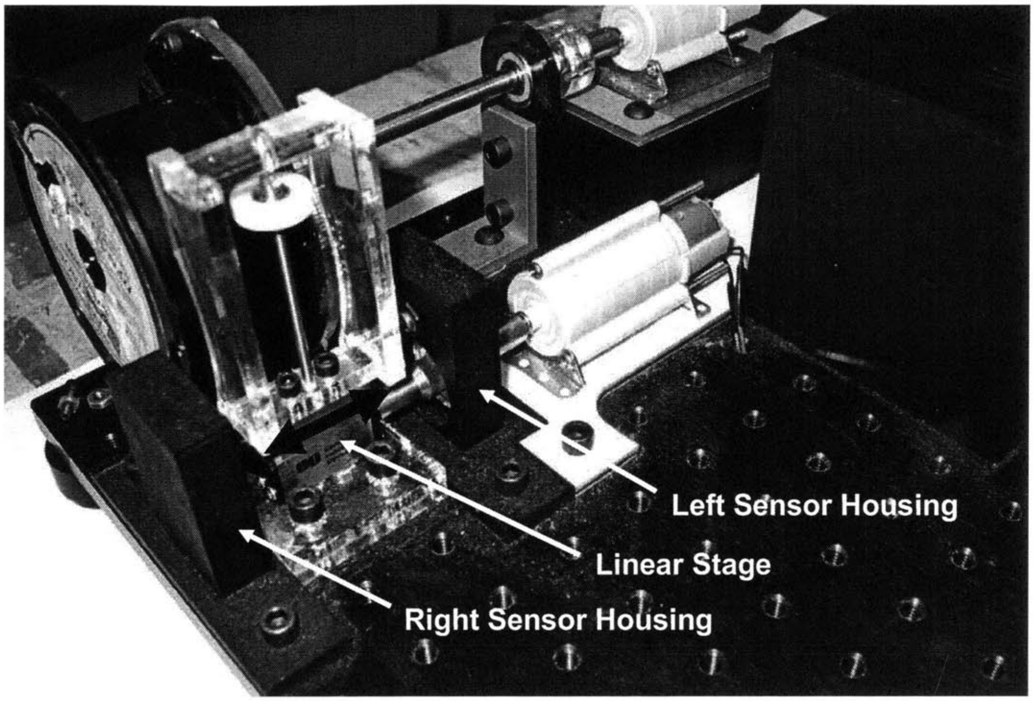
14.) Turn off the power supply and the basic stamp when done.

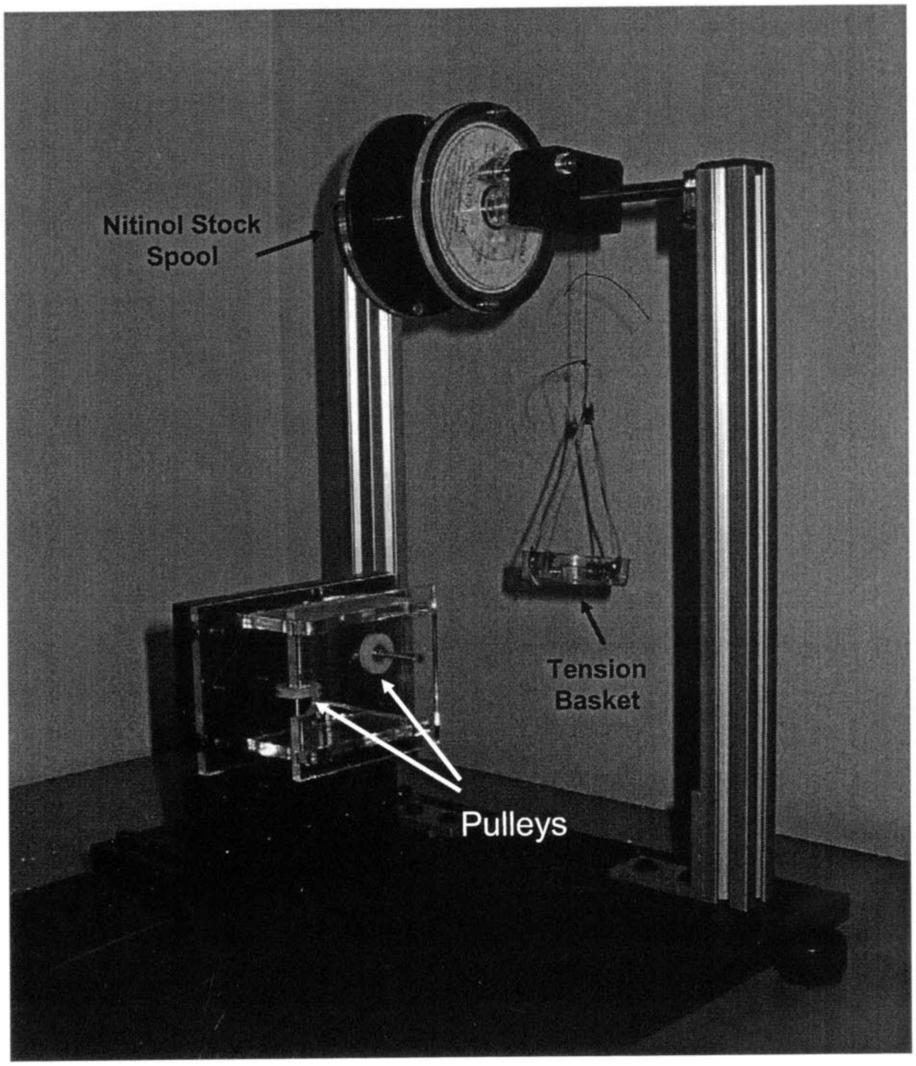
Notes:

- Flipping the toggle switch to the left when the linear stage is moving to the right will **NOT** stop the spool.
- There are two ways to change the direction of the linear stage without waiting until the stage contacts the sensor: (1) manually push the appropriate sensor with your finger, or (2) if you want to move the stage to the left, you can hit the reset button on the basic stamp, and move the toggle switch to the reverse position.
- Push buttons are used as sensors to stop the motors when the linear stage is at the end of its travel.


Off
On
Basic Stamp


Take-up Spool Motor
Take-up Spool Mount
Reverse
Forward
Take-up Spool
Linear Stage Motor


Left Sensor Housing
Linear Stage
Right Sensor Housing

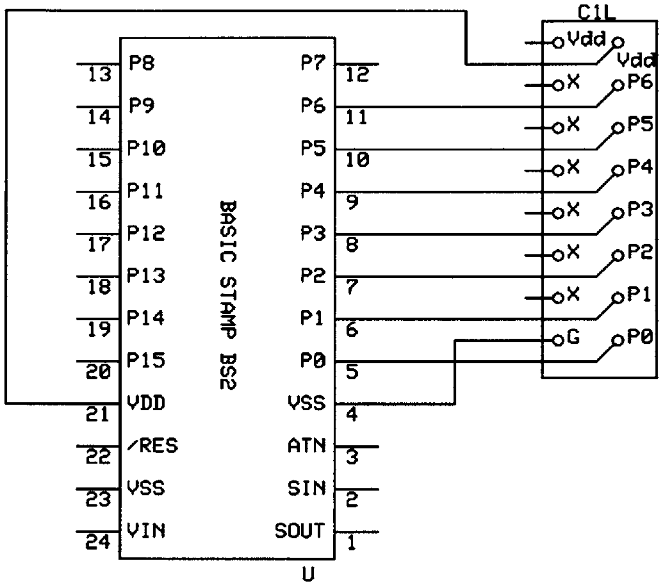

Nitinol Stock Spool
Tension Basket
Pulleys

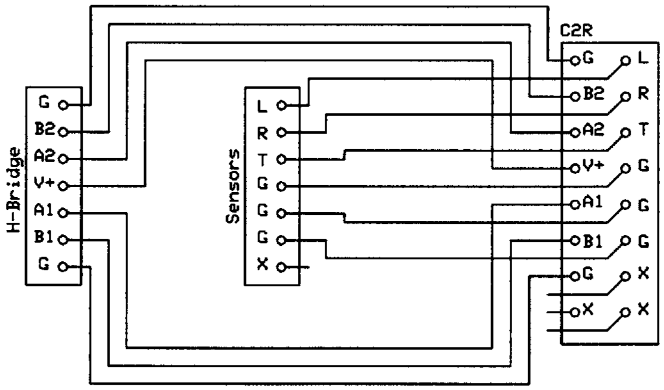
BASIC Stamp Program for Controlling Motor Speed

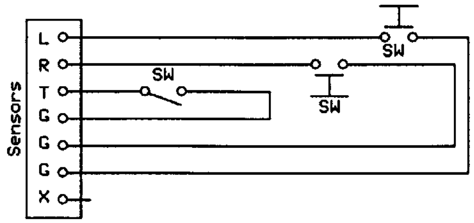
```
'Program: SpoolingMotors.BS2
'This program is used to control the speeds of two motors used to wind Nitinol in
parallel.
'It will use pulse width modulation to control the speed.

'======================== Constants and Pin Assignments
====================================
Stage_Pulse         CON       6         'Off time for the motor that controls the linear
stage ( A higher number turns the motor slower) This number should not exceed the
Interval
R_Stage_Pulse       CON       1         'Off time for the motor that controls the linear
stage when traveling in the reverse direction.
Spool_Pulse         CON       10        'Off time for the motor that turns the take-up
spool (A higher number is slower)
Interval  CON       30        'Total width of both stage and spool pulse

Left_Contact        VAR       IN0       'State of right button that senses contact
(0=contact)
Right_Contact       VAR       IN1       'State of left button that senses contact
(0=contact)
Switch              VAR       IN2       'State of toggle switch that controls direction
of the stage.(1=right)

Stage_1             CON       5         'The BASIC stamp pin that connects to B1 on the
H-bridge
Stage_2             CON       6         'The BASIC stamnp pin that connects to B2 on the
H-bridge
Spool_1             CON       3         'The BASIC stamp pin that connects to A1 on the
H-bridge
Spool_2             CON       4         'The BASIC stamp pin that connects to A2 on the
H-bridge

I                   VAR       byte      'Loop counter.  Largest possible value is 255
'=====================================================================================
====
'Stop all the motors and wait for the toggle switch to be pointed left
GoSub Stop_Motors
          debug cls, "Stop and wait for Reverse"
GoTo Check_Reverse


'======================== Forward Drive
=======================================================
Check_Forward:
          If Switch = 1 Then Forward
GoTO Check_Forward

Forward:
          debug cls, "Forward"
          High Stage_1                                      'Puts A1 into high logic
which drives the stage right (Counter Clockwise)
          High Spool_2                                      'Puts B2 into high logic
which drives the spool Clockwise (top toward you)
          For I=1 to Interval
                    If Right_Contact = 0 Then Exit_Forward
                    If I >= Interval - Stage_Pulse Then Toggle_Stage_1
                    Check_Spool_2:
                    If I >= Interval - Spool_Pulse Then Toggle_Spool_2

                    GoTo Forward_Loop_End
                    Toggle_Stage_1:
                              Low Stage_1                   ' Sets the Stage_1 (A1) Pin
to low.
                              GoTo Check_Spool_2
                    Toggle_Spool_2:
                              Low Spool_2
                    Forward_Loop_End:
          Next

          GoTo Forward

Exit_Forward:
          GoSub Stop_Motors
          debug cls, "Stop and wait for Reverse"

Check_Reverse:
```

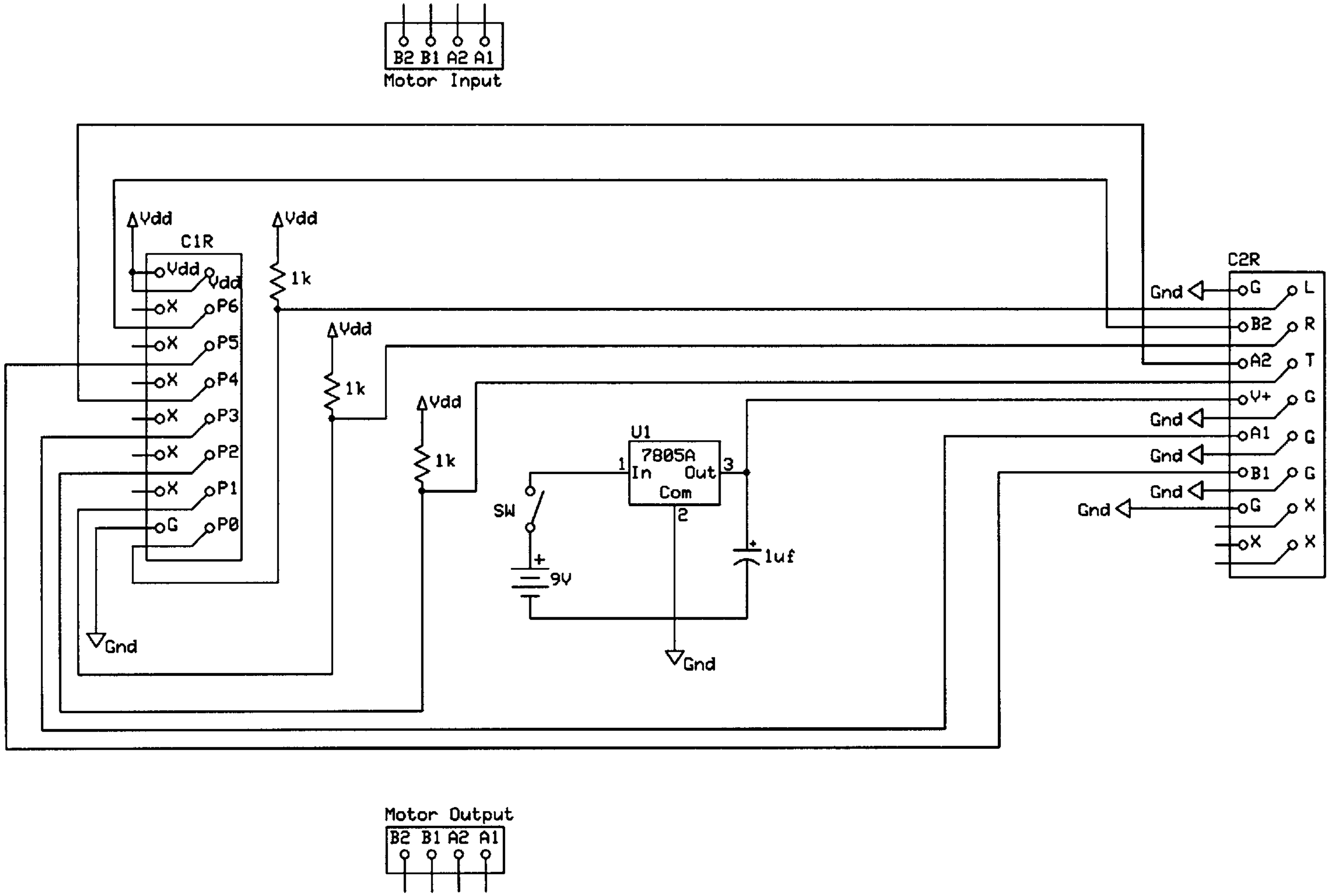
```
          If Switch = 0 Then Reverse_Mode
GoTo Check_Reverse

Reverse_Mode:
          debug cls,"Reverse"
          High Stage_2
          High Spool_1
          For I=1 to Interval
                    If Left_Contact = 0 Then Exit_Reverse
                    If I >= Interval - R_Stage_Pulse Then Toggle_Stage_2
                    Check_Spool_1:
                    If I >= Interval - Spool_Pulse Then Toggle_Spool_1

                    GoTo Reverse_Loop_End
                    Toggle_Stage_2:
                              Low Stage_2
                              GoTo Check_Spool_1
                    Toggle_Spool_1:
                              Low Spool_1
                    Reverse_Loop_End:
          Next

          GoTo Reverse_Mode

Exit_Reverse:
          GoSub Stop_Motors
                    debug cls, "Stop and wait for Forward"
          GoTo Check_Forward


'======================== Stop SubRoutine =====
' Sets all the output pins to low
'==============================================
Stop_Motors:
          low Stage_1
          low Stage_2
          low Spool_1
          low Spool_2
Return
```

Electronics Schematics








B2 B1 A2 A1
Motor Input
Vdd
C1R
Vdd
Vdd
X
P6
X
P5
X
P4
X
P3
X
P2
X
P1
G
P0
Gnd
Vdd
1k
Vdd
1k
Vdd
1k
U1
7805A
1
In
Out
3
Com
2
SW
+
9V
+
1uf
Gnd
C2R
Gnd
G
L
B2
R
A2
T
V+
G
Gnd
A1
G
Gnd
B1
G
Gnd
G
X
Gnd
X
X
Motor Output
B2 B1 A2 A1

Electronics Notes

- C1R is the right side of a ribbon cable connector that connects the basic stamp to a pc board containing the voltage regulator, the three resistors, and the capacitor. The "x" next to a pin indicates that it is not used.
- C2R is a ribbon cable connector that connects the pc board with the voltage regulator circuit to the H-Bridge.
- C1L connects to the left side of the ribbon cable of C1R. This side connects to the BASIC Stamp pins indicated.

• List of Components

Part Number	Description	Qty	Unit Price	Total Price	Vendor
R53-383	In-Line Precision Stage Mechanism (19mm Total Travel)	1	$55.00	$55.00	Edmund Industrial Optics
#3-225	Planetary Gear Box Kit	3	$27.95	$83.85	Robot Store
#3-301	Mini Dual H-Bridge	1	$24.95	$24.95	Robot Store
CGSX-2-06	Stainless Steel Precision Shaft 1/8" dia. 6" long	3	$1.90	$5.70	Small Parts Inc.
MCGSX-08-6	Stainless Steel Precision Shaft 8mm dia. 6" long	1	$5.95	$5.95	Small Parts Inc.
PNB-2/12-01	Nylon Pulley Bearing Mounted 1/8" Bore 3/4" OD	3	$6.25	$18.75	Small Parts Inc.
28150	Board Of Education BASIC Stamp Kit	1	$65.00	$65.00	Parallax
[redacted]	Push Button	2	[redacted]	[redacted]	Electronics Supplier
[redacted]	Toggle Switch	1	[redacted]	[redacted]	Electronics Supplier
[redacted]	7805A Adjustable Voltage Regulator	1	[redacted]	[redacted]	Electronics Supplier
[redacted]	1K Resistor	3	[redacted]	[redacted]	Electronics Supplier

Suppliers

Edmund Industrial Optics
101 E. Gloucester Pike
Barrington, NJ 08007-1380
Phone: (800) 363-1992
Fax: (856)573-6295
Website: www.edmundoptics.com

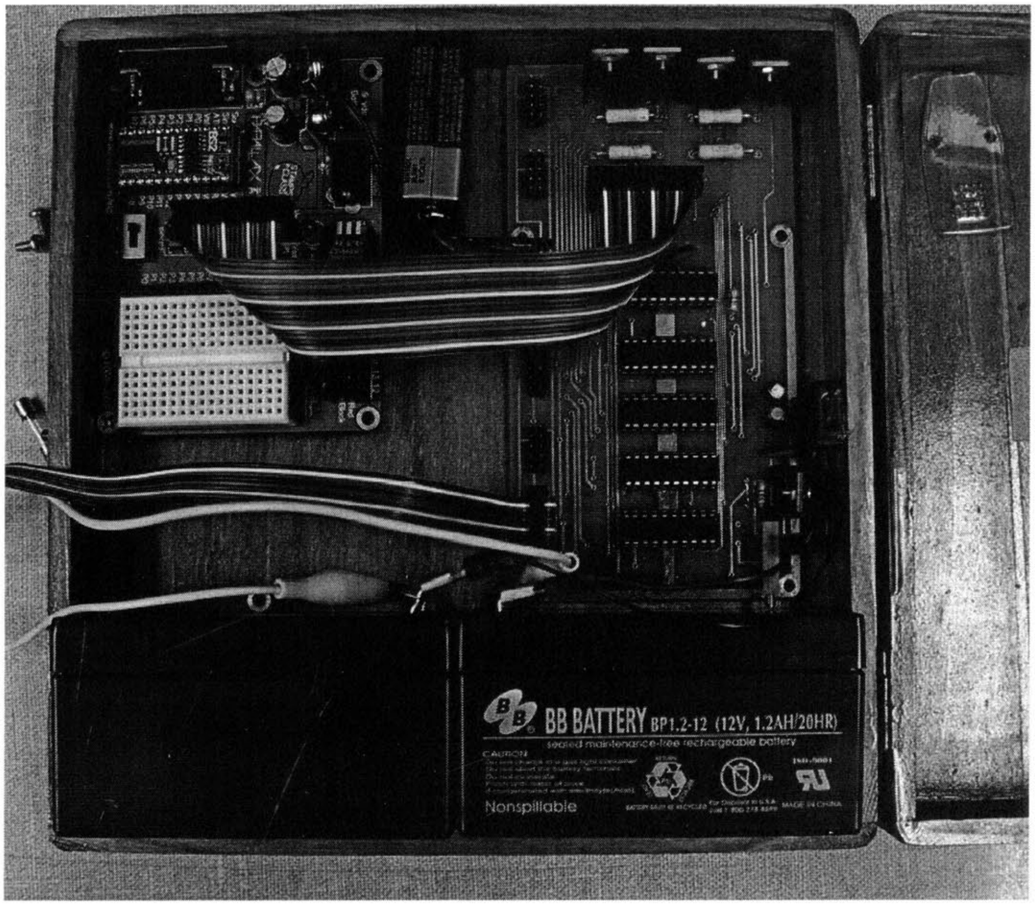
Mondu-tronics Inc.
124 Paul Drive, Suite 12
San Rafael, CA 94903
Phone: (800) 374-5764
Fax: (415)491-4696
Website: www.robotstore.com

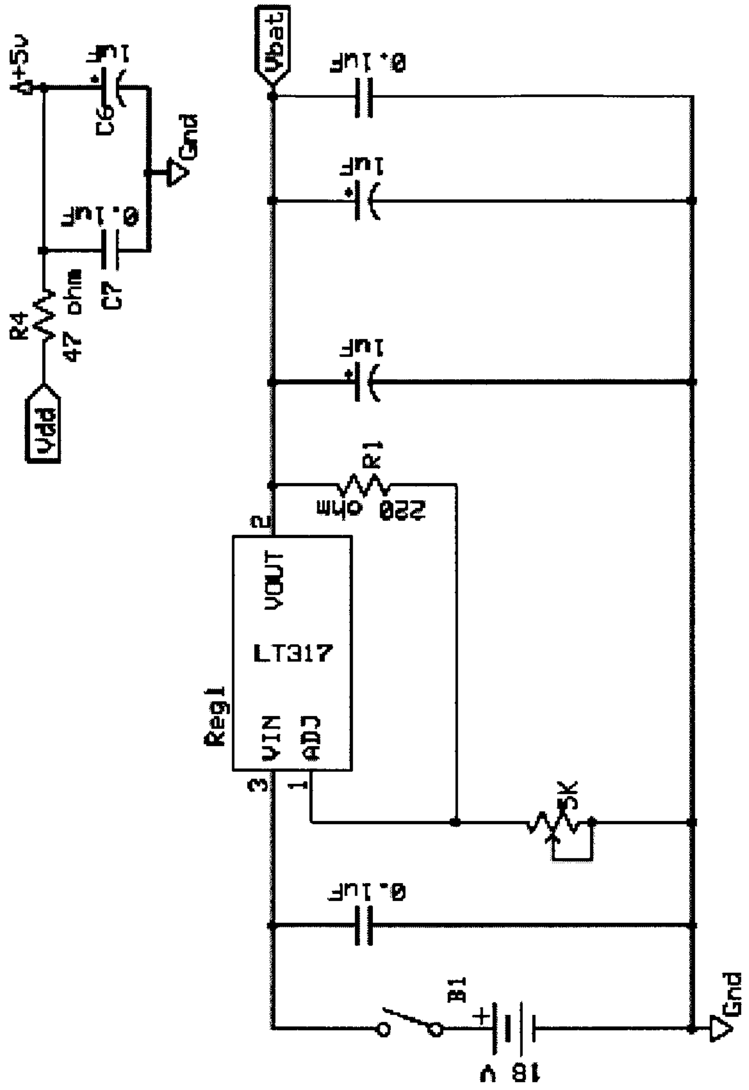
Parallax Inc.
599 Menio Drive, Suite 100
Rocklin, CA 95765
Phone: (916) 624-8333
Fax: (916) 624-8003
Website: www.parallax.com

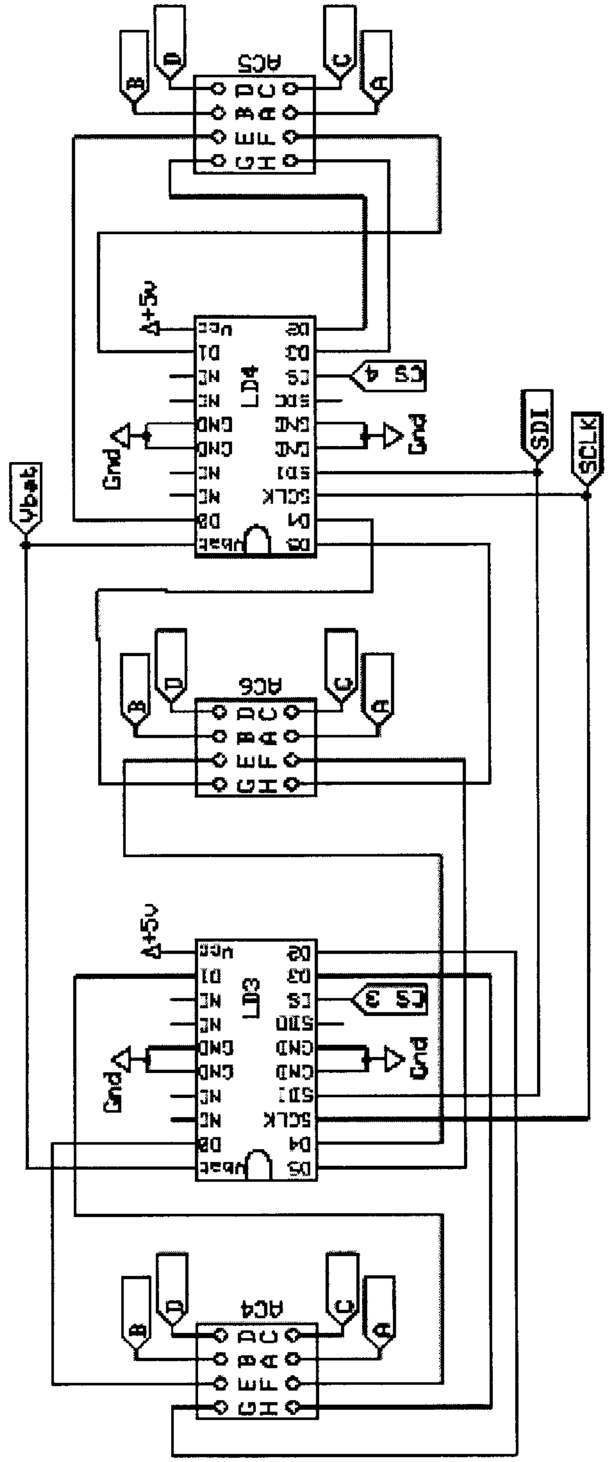
Small Parts Inc.
13980 N.W. 58th Court
P.O. Box 4650
Miami Lakes, FL 33014-0650
Phone: (800) 220-4242
Fax: (800)423-9009
Website: www.smallparts.com

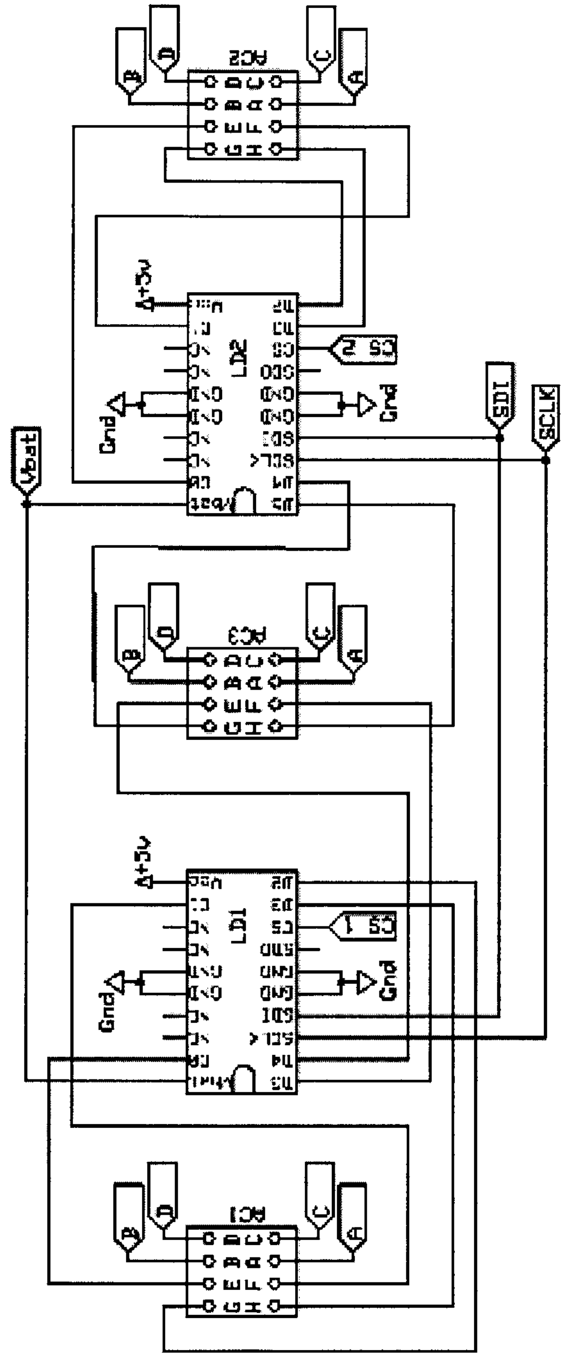
Appendix B: Drive Electronics for Prototype 3

Drive Electronics Prototype 3

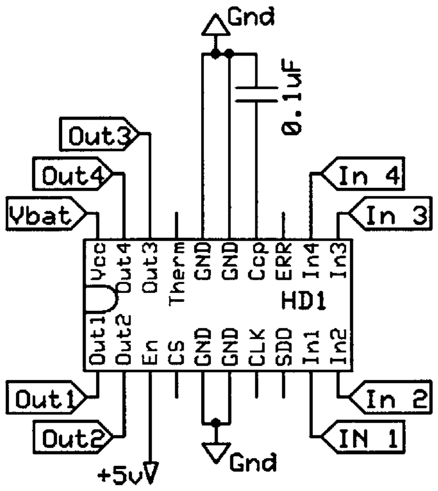

BB BATTERY BP1.2-12 (12V, 1.2AH/20HR)
sealed maintenance-free rechargeable battery
Nonspillable
MADE IN CHINA

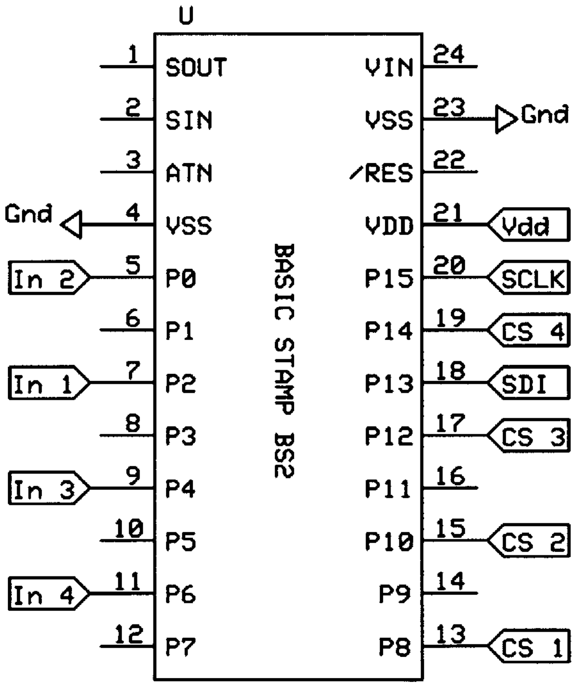

Vdd
R4
47 ohm
C7
0.1uF
C6
1uF
+5v
Gnd
Vbat
0.1uF
1uF
1uF
R1
220 ohm
Reg1
LT317
VIN
ADJ
VOUT
3
1
2
5K
0.1uF
B1
18 V
Gnd

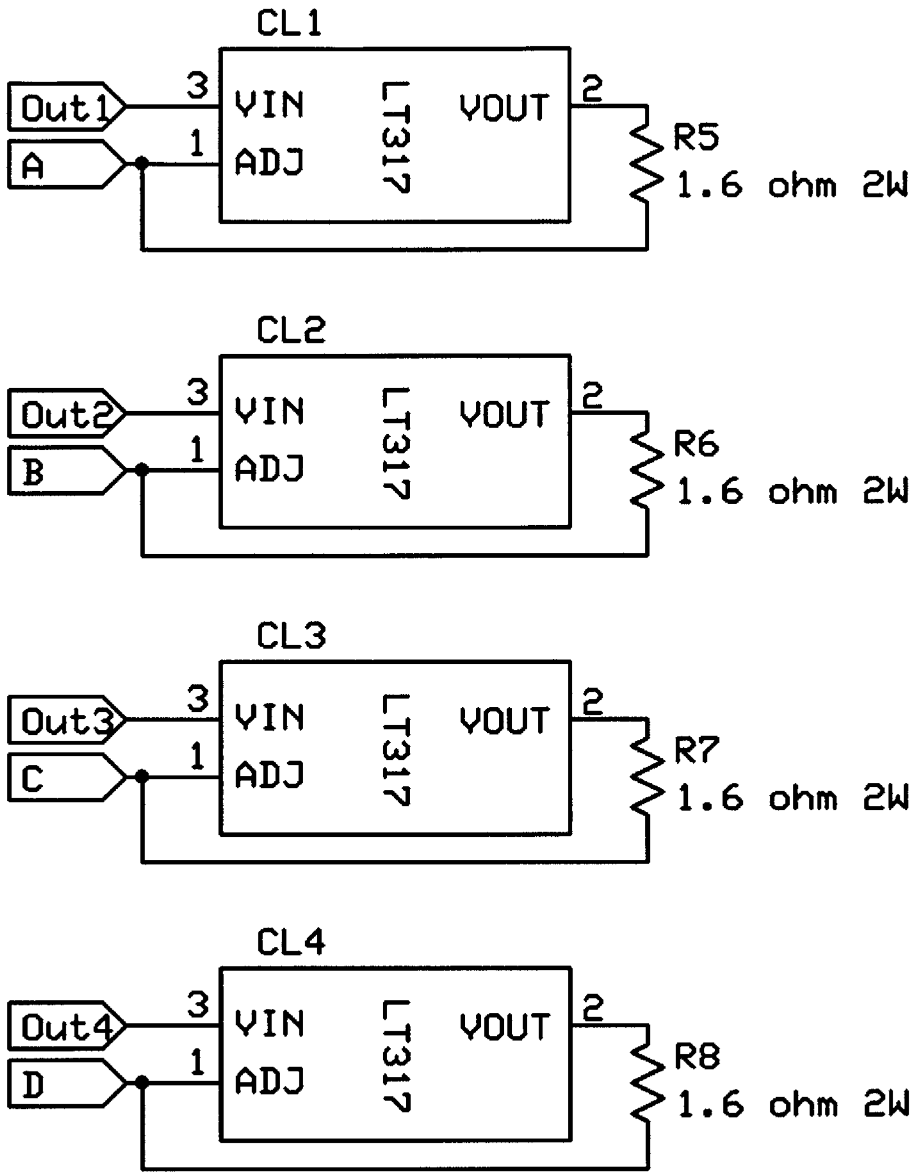

AC5
A
B
C
D
E
F
G
H
LD4
Vcc
D1
NC
GND
D0
Vbat
D6
D3
CS
SDO
SDI
SCLK
D4
D5
+5v
Gnd
CS 4
SDI
SCLK
Vbat
AC6
LD3
CS 3
AC4

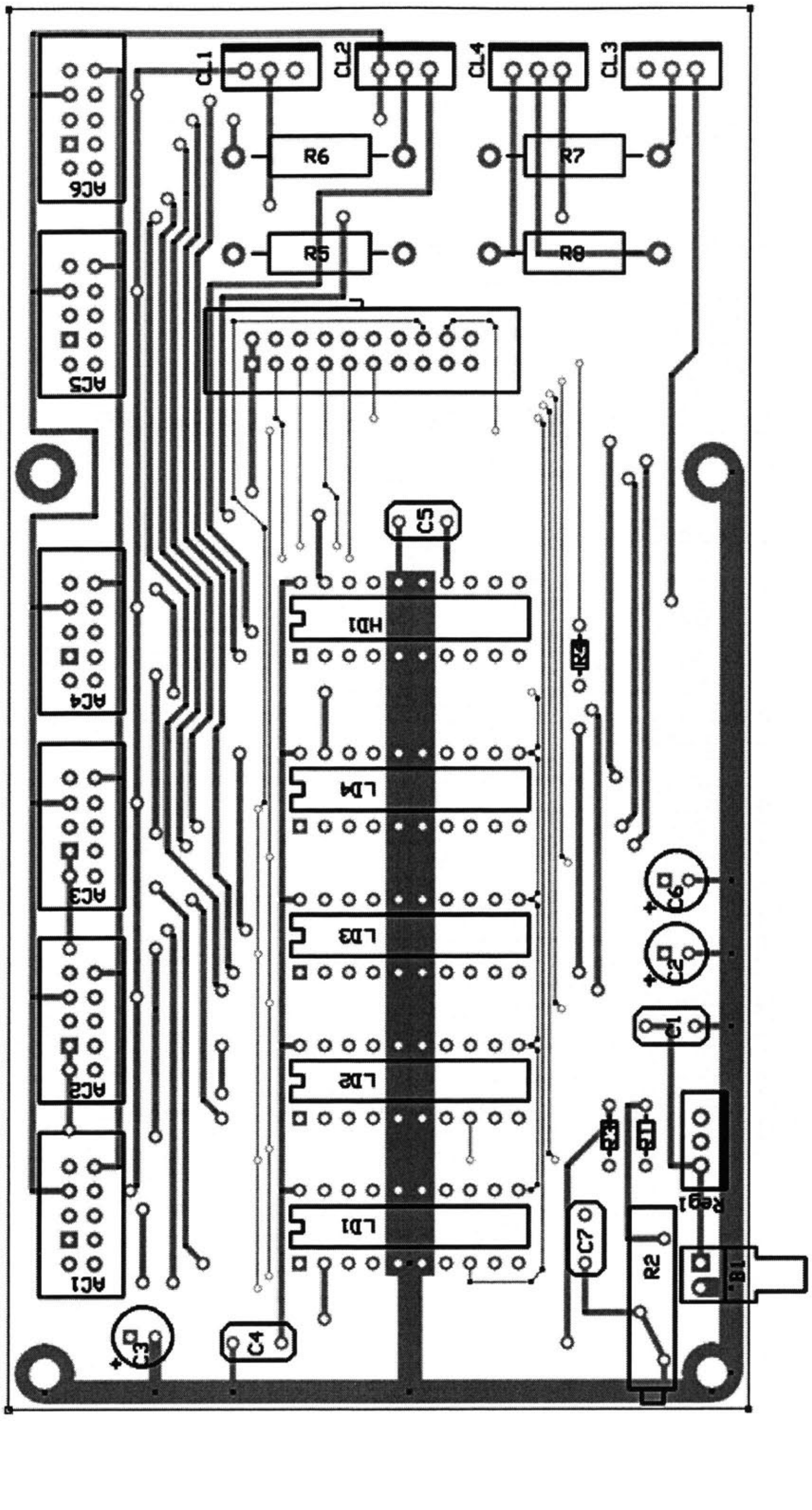


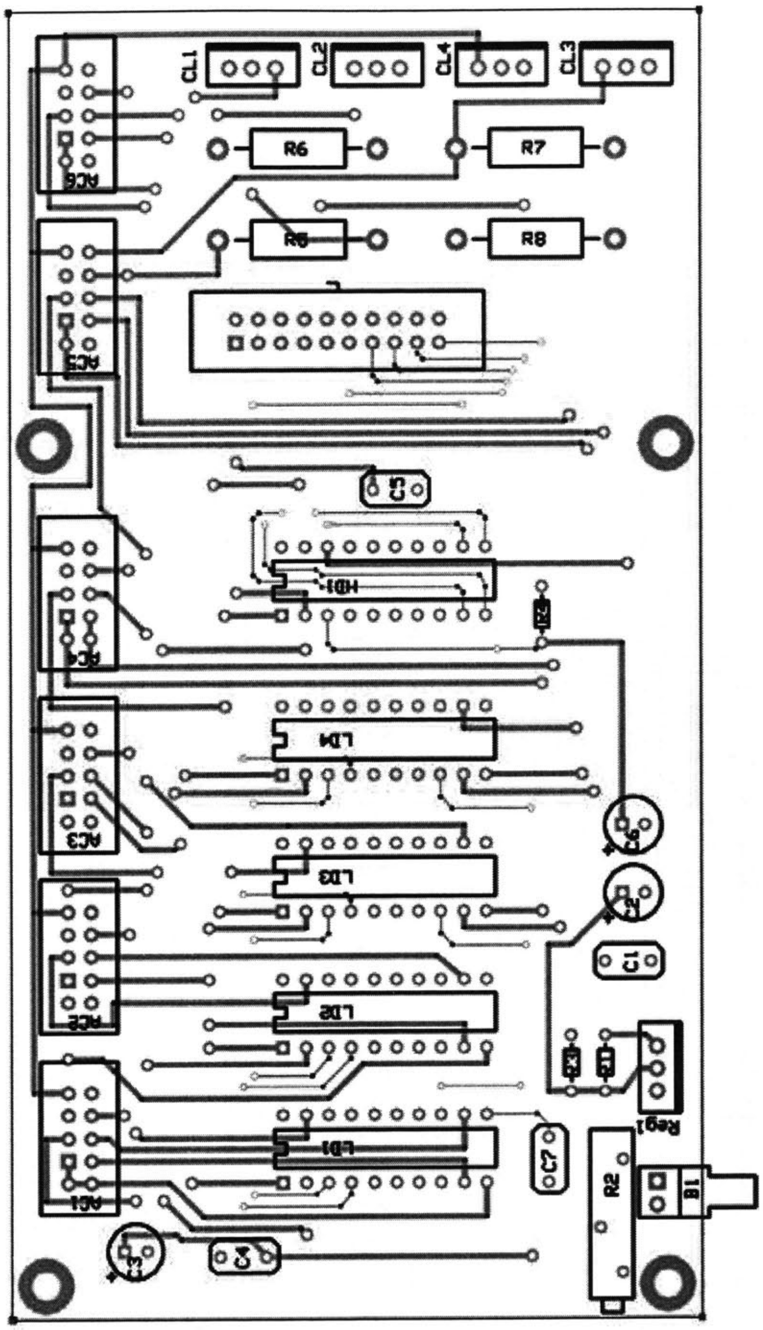
AC2
A
B
C
D
+5V
LD2
CS 2
Gnd
Gnd
SDI
SCLK
Vbat
AC3
A
B
C
D
+5V
LD1
CS 1
Gnd
Gnd
AC1
A
B
C
D

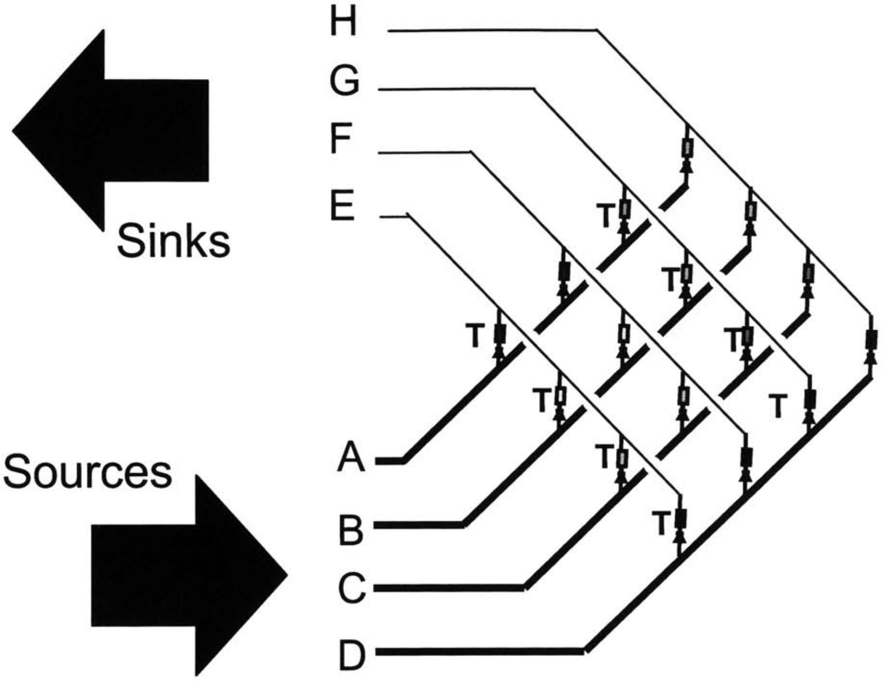

Gnd
0.1uF
Out3
Out4
Vbat
In 4
In 3
Vcc
Out4
Out3
Therm
GND
GND
Ccp
ERR
In4
In3
HD1
Out1
Out2
En
CS
GND
GND
CLK
SDD
In1
In2
Out1
Out2
+5v
Gnd
In 2
IN 1

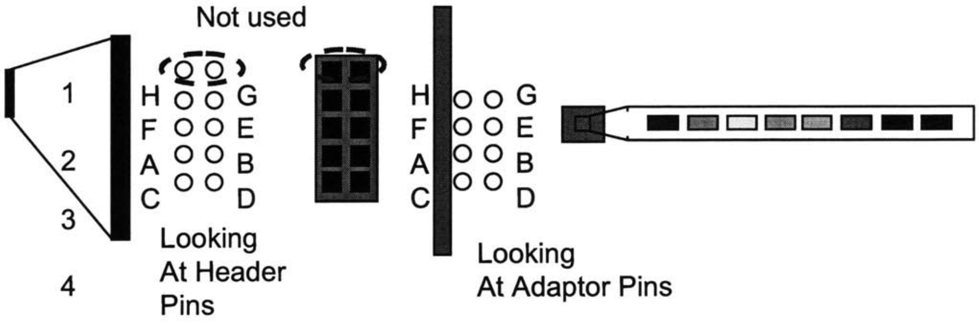

U
1 SOUT
2 SIN
3 ATN
4 VSS
Gnd
5 P0
In 2
6 P1
7 P2
In 1
8 P3
9 P4
In 3
10 P5
11 P6
In 4
12 P7
BASIC STAMP BS2
VIN 24
VSS 23
Gnd
/RES 22
VDD 21
Vdd
P15 20
SCLK
P14 19
CS 4
P13 18
SDI
P12 17
CS 3
P11 16
P10 15
CS 2
P9 14
P8 13
CS 1

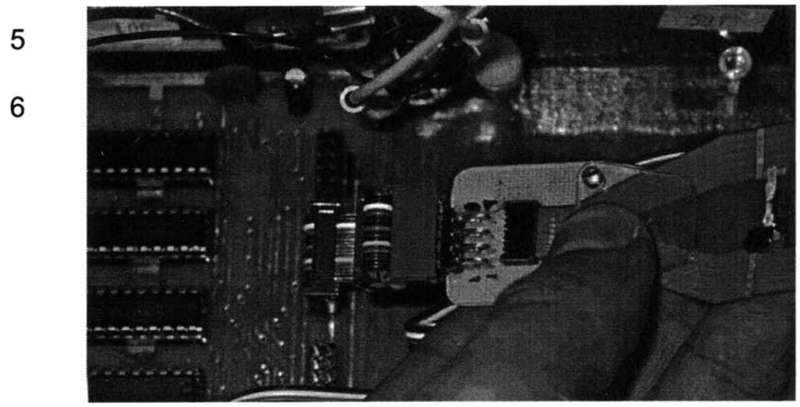

CL1
Out1
A
3
1
VIN
ADJ
LT317
VOUT
2
R5
1.6 ohm 2W
CL2
Out2
B
3
1
VIN
ADJ
LT317
VOUT
2
R6
1.6 ohm 2W
CL3
Out3
C
3
1
VIN
ADJ
LT317
VOUT
2
R7
1.6 ohm 2W
CL4
Out4
D
3
1
VIN
ADJ
LT317
VOUT
2
R8
1.6 ohm 2W


CL1
CL2
CL4
CL3
R6
R7
R5
R8
AC6
AC5
AC4
AC3
AC2
AC1
C5
HD1
R4
LD4
LD3
LD2
LD1
C6
C2
C1
R3
R1
Reg1
C7
R2
B1
C3
C4


CL1
CL2
CL4
CL3
R6
R7
R5
R8
AC6
AC5
AC4
AC3
AC2
AC1
C5
HD1
LD4
LD3
LD2
LD1
C6
C2
C1
Reg1
C7
R2
B1
C4
C3


H
G
F
E
Sinks
T
T
T
T
T
T
T
T
Sources
A
B
C
D


Not used
1
2
3
4
H G
F E
A B
C D
Looking
At Header
Pins
H G
F E
A B
C D
Looking
At Adaptor Pins


5
6

Component List

Part Description	Qty	Component #
LM317 Voltage Regulator	5	CL1, CL2, CL3, CL4, Reg1
220 Ω Resistor	1	R1
5,000 Ω 15 Turn Cermet Potentiometer	1	R2
47 Ω Resistor	1	R4
1.6 Ω Metal Oxide Resistor (2 Watt Rating)	4	R4, R6, R7, R8
0.1 µF Ceramic Disk Capacitor (Rating > 25V)	4	C1, C4, C5, C7
1 µF Electrolytic Capacitor (Rating > 25V)	3	C2, C3, C6
10 Pin Header	6	AC1, AC2, AC3, AC4, AC5, AC6
TPIC2603NE Low-Side Driver	4	LD1, LD2, LD3, LD4
LMD18400 High Side Driver	1	HD1
20 Pin Header	1	U
2 Pin Wire Connector	1	SW1

Electronic components can be purchased from

Digi-Key Corporation
701 Brooks Avenue South
Thief River Falls, MN 56701
Phone: (800) 344-4539
Fax: (218) 681-3380
Website: www.digikey.com

The printed circuit boards can be purchased from

PCBexpress.com

They provide free software to make the schematics and the layout files

User Interface Source Code C++

By Don Jin Lee

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "Stdio.h"
#include "math.h"
#include "VDCommon.h"
#include "VDLine.h"
#include "VDPolyline.h"

#define ElementAngle    40.0
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma link "VDStdLib_OCX"
#pragma link "VrButtons"
#pragma link "VrControls"
#pragma link "VrDesign"
#pragma link "VrNavigator"
#pragma link "iComponent"
#pragma link "iCustomComponent"
#pragma link "iLed"
#pragma link "iLedRectangle"
#pragma link "iLedRound"
#pragma link "iVCLComponent"
#pragma link "VrSystem"
#pragma link "AdPort"
#pragma link "OoMisc"
#pragma resource "*.dfm"
TForm1 *Form1;

//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
  : TForm(Owner)
{

  InitialStateFlag = true;
  ReadyFlag = false;

}
//---------------------------------------------------------------------------
```

```
void __fastcall TForm1::FormCreate(TObject *Sender)
{

   ActionData->Cells[1][0] = "Array #";
   ActionData->Cells[2][0] = "Element #";
   ActionData->Cells[3][0] = "Time";
   for ( int i=1; i<=1000; i++ )
      ActionData->Cells[0][i] = IntToStr(i);
   ActionData->ColWidths[0] = 30;
   ActionData->ColWidths[1] = 39;
   ActionData->ColWidths[2] = 53;
   ActionData->ColWidths[3] = 30;

   VectorDrawFuntionsInitialize();

   dOpen->InitialDir = ExtractFilePath( Application->ExeName );
   dSave->InitialDir = dOpen->InitialDir;
   Com1->PutBlock("",0);

}
//---------------------------------------------------------------------------

void __fastcall TForm1::bbOpenClick(TObject *Sender)
{
   FILE * fp;

/*   ActionDataType d;
   TSpeedButton * Tmp;
   if ( dOpen->Execute() )
   {
      fp = fopen( dOpen->FileName.c_str(), "rb" );
      if ( fp == NULL )
      {
         Beep();
         ShowMessage("Can't Open Action Data File." );
         return;
      }
      else
      {
         InitialStateFlag = false;
         fread( InitialState, sizeof(ElementArray), 6, fp );

         ActionData->Cols[1]->Clear();
         ActionData->Cols[2]->Clear();
         ActionData->Cols[3]->Clear();
         ActionData->Cells[1][0] = "Array #";
         ActionData->Cells[2][0] = "Element #";
         ActionData->Cells[3][0] = "Time";

         ActionData->Row = 1;
         while( !feof(fp) )
         {
            if ( fread( &d, sizeof(ActionDataType),1,fp) != 1 ) break;
            ActionData->Cells[1][ActionData->Row] = (int)d.Array;
            ActionData->Cells[2][ActionData->Row] = (int)d.Element;
            ActionData->Cells[3][ActionData->Row] = d.Time;
            ActionData->Row++;

         }
         fclose(fp);
         ActionData->Row = 1;
         MakeCurrentState( InitialState );
      }
   }
*/
   if ( dOpen->Execute() )
   {
      fp = fopen( dOpen->FileName.c_str(), "rb" );
      if ( fp == NULL )
      {
         Beep();
         ShowMessage("Can't Open Action Data File." );
         return;
      }
      else
      {
         InitialStateFlag = false;

         for ( int i=0; i<6; i++ )
         {
            int tmp;
            fscanf( fp, "%x\n", &tmp );
            for ( int j=0; j<8; j++ )
            {
               if ( tmp & 1 )
                  InitialState[i].Element[j].Bit = 1;
               else
                  InitialState[i].Element[j].Bit = 0;
               tmp = tmp >> 1;
            }
         }
      ActionData->Cols[1]->Clear();
      ActionData->Cols[2]->Clear();
      ActionData->Cols[3]->Clear();
      ActionData->Cells[1][0] = "Array #";
      ActionData->Cells[2][0] = "Element #";
      ActionData->Cells[3][0] = "Time";

      ActionData->Row = 1;
      int an, en, t;
      while( !feof(fp) )
      {
        if ( fscanf( fp, "%d %d %d\n", &an, &en, &t ) != 3 ) break;
        ActionData->Cells[1][ActionData->Row] = an;
        ActionData->Cells[2][ActionData->Row] = en;
        ActionData->Cells[3][ActionData->Row] = t;
        ActionData->Row++;

      }
      fclose(fp);
      ActionData->Row = 1;
      MakeCurrentState( InitialState );
    }
  }


}
//---------------------------------------------------------------------------

void __fastcall TForm1::udPauseTimeChanging(TObject *Sender, bool &AllowChange)
{
  ePauseTime->Text = udPauseTime->Position;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::nActionDataNaviButtonClick(TObject *Sender,
   TVrButtonType Button)
{
  switch( Button )
  {
    case Vrnavigator::btPrev :
      ActionData->Row = 1;
      MakeCurrentState(InitialState );
      break;
    case Vrnavigator::btBack :
      if ( ActionData->Row > 1 )
      {
            ActionData->Row--;;
            FindCurrentState();
            MakeCurrentState(ElementTmp );
        }
        break;
    case Vrnavigator::btStep :
        if ( ActionData->Row < 1000 )
        {
            ActionData->Row++;
            FindCurrentState();
            MakeCurrentState(ElementTmp );
        }
        break;
    case Vrnavigator::btNext :
        for ( int i=1; i<=1000; i++ )
        {
            if ( ActionData->Cells[1][i] == "" )
            {
                ActionData->Row = i;
                break;
            }
        }
        FindCurrentState();
        MakeCurrentState(ElementTmp );

        break;
    case Vrnavigator::btEject :
        for ( int i=ActionData->Row; i<1000; i++ )
        {
            if ( ActionData->Cells[1][i+1] != "" )
            {
                ActionData->Cells[1][i] = ActionData->Cells[1][i+1];
                ActionData->Cells[2][i] = ActionData->Cells[2][i+1];
                ActionData->Cells[3][i] = ActionData->Cells[3][i+1];
            }
            else
                break;
        }
        ActionData->Cells[1][1000] = "";
        ActionData->Cells[2][1000] = "";
        ActionData->Cells[3][1000] = "";
        FindCurrentState();
        MakeCurrentState(ElementTmp );
        break;

}
```

```
}
//---------------------------------------------------------------------------

void __fastcall TForm1::sbA6Element8Click(TObject *Sender)
{
   int ArrayNo, ElementNo;
   TSpeedButton * Tmp = dynamic_cast<TSpeedButton*>(Sender);

   if ( dbRecord->Tag == 0 && InitialStateFlag == false ) // if not recording action or
there are any actions, then return.
   {
      Tmp->Down = !Tmp->Down;
      return;
   }

   if ( Tmp->Down == true )
      Tmp->Glyph = blButtonImage->GetBitmap(10); //  Concave;
   else
      Tmp->Glyph = blButtonImage->GetBitmap(11);

   ArrayNo = (Tmp->Tag - (Tmp->Tag %8))/8 + 1;      // Array No.
   ElementNo = Tmp->Tag % 8 + 1;                    // Element No.


   if ( dbRecord->Tag == 0 )
   {
      RotateElement( ArrayNo, ElementNo );
      return;
   }

   for ( int i=1000; i>ActionData->Row; i-- )
   {
      if ( ActionData->Cells[1][i-1] == "" ) continue;

      ActionData->Cells[1][i] = ActionData->Cells[1][i-1];
      ActionData->Cells[2][i] = ActionData->Cells[2][i-1];
      ActionData->Cells[3][i] = ActionData->Cells[3][i-1];
   }

   ActionData->Cells[1][ActionData->Row] = ArrayNo;

   ActionData->Cells[2][ActionData->Row] = ElementNo;

   ActionData->Cells[3][ActionData->Row] = udPauseTime->Position;
   ActionData->Row += 1;
```

```
}
//---------------------------------------------------------------------------




void __fastcall TForm1::dbPowerClick(TObject *Sender)
{

   if ( dbPower->Tag == 0 )   // if PowerButton is Off
   {
      dbPower->Tag = 1;
      dbPower->Bitmap = blButtonImage->GetBitmap(dbPower->Tag);
      dbPlay->Enabled = true;
      dbPause->Enabled = true;
      dbStop->Enabled = true;
      dbRecord->Enabled = true;
      pElement->Enabled = true;
      pNavi->Enabled = true;
      pCommand->Enabled = true;
      pDemo->Enabled = false;
   }
   else                   // if PowerButton is On
   {
      dbPower->Tag = 0;
      if ( dbRecord->Tag == 1 )
         dbRecordClick(dbRecord);
      if ( dbPlay->Tag == 1 )
         dbPlayClick(dbPlay);
      dbPower->Bitmap = blButtonImage->GetBitmap(dbPower->Tag);
      dbPlay->Enabled = false;
      dbPause->Enabled = false;
      dbStop->Enabled = false;
      dbRecord->Enabled = false;
      pElement->Enabled = false;
      pNavi->Enabled = false;
      pCommand->Enabled = false;
      pDemo->Enabled = true;
   }
}
//---------------------------------------------------------------------------

void __fastcall TForm1::dbPlayClick(TObject *Sender)
{
  if ( dbPlay->Tag == 0 )
  {
    if ( ActionData->Cells[1][ActionData->Row] == "" )
    {
      Beep();
      ShowMessage( "There are no action to play." );
      return;
    }
    dbPlay->Tag = 1;
    dbPlay->Bitmap = blButtonImage->GetBitmap(dbPlay->Tag+2);
    dbRecord->Enabled = false;
    ActionData->Enabled = false;
    tPlay->Interval = ActionData->Cells[1][ActionData->Row].ToInt();
    tPlay->Enabled = true;
  }
  else
  {
    dbPlay->Tag = 0;
    dbPlay->Bitmap = blButtonImage->GetBitmap(dbPlay->Tag+2);
    dbRecord->Enabled = true;
    ActionData->Enabled = true;
    tPlay->Enabled = false;
  }
}
//---------------------------------------------------------------------------


void __fastcall TForm1::dbStopClick(TObject *Sender)
{
  if ( dbRecord->Tag == 1 )
  {
    dbRecordClick(dbRecord);
    return;
  }

  if ( dbPlay->Tag == 1 )
  {
    dbPlayClick(dbRecord);
    return;
  }

}
//---------------------------------------------------------------------------

void __fastcall TForm1::dbRecordClick(TObject *Sender)
{
   TSpeedButton * Tmp;
   if ( dbRecord->Tag == 0 )
   {
      dbRecord->Tag = 1;
      dbRecord->Bitmap = blButtonImage->GetBitmap(dbRecord->Tag+8);
      dbPlay->Enabled = false;

      if ( InitialStateFlag == false ) return;
      for ( int i=0; i<6; i++ )
      {
         for ( int j=0; j<8; j++ )
         {
            Tmp = (TSpeedButton *)FindComponent( "sbA" + IntToStr(i+1) + "Element"
+ IntToStr(j+1));
            if ( Tmp->Down )
               InitialState[i].Element[j].Bit = 1;
            else
               InitialState[i].Element[j].Bit = 0;

         }
      }
      InitialStateFlag = false;
   }
   else
   {
      dbRecord->Tag = 0;
      dbRecord->Bitmap = blButtonImage->GetBitmap(dbRecord->Tag+8);
      dbPlay->Enabled = true;
   }
}
//---------------------------------------------------------------------------


void __fastcall TForm1::sbA1Element1MouseMove(TObject *Sender,
     TShiftState Shift, int X, int Y)
{
   TSpeedButton * Tmp = dynamic_cast<TSpeedButton*>(Sender);
   Tmp->ShowHint = true;
   Tmp->Hint = "Array #" + IntToStr( (Tmp->Tag - (Tmp->Tag % 8 ))/8 + 1) + "
Element #" + IntToStr( Tmp->Tag % 8 + 1);
}
//---------------------------------------------------------------------------


void __fastcall TForm1::bbSaveClick(TObject *Sender)
{
  FILE * fp;

  if ( dSave->Execute() )
  {
    fp = fopen( dSave->FileName.c_str(), "wt" );
    if ( fp == NULL )
    {
      Beep();
      ShowMessage("Can't Create Action Data File." );
      return;
    }
    else
    {
      for ( int i=0; i<6; i++ )
      {
        int tmp =0;
        for ( int j=0; j<8; j++ )
        {
          if ( InitialState[i].Element[j].Bit )
            tmp += (1 << j);
        }
        fprintf( fp, "0x%x\n", tmp );
      }
      for ( int i=1; i<1000; i++ )
      {
        if ( ActionData->Cells[1][1] == "" ) break;
        fprintf( fp , "%s %s %s\n", ActionData->Cells[1][i].c_str(),ActionData-
>Cells[2][i].c_str(),ActionData->Cells[3][i].c_str());
      }
      fclose(fp);
    }
  }
}
//---------------------------------------------------------------------------

void __fastcall TForm1::bbNewClick(TObject *Sender)
{
  TSpeedButton * Tmp;
  InitialStateFlag = true;
  ActionData->Cols[1]->Clear();
  ActionData->Cols[2]->Clear();
  ActionData->Cols[3]->Clear();
  ActionData->Cells[1][0] = "Array #";
  ActionData->Cells[2][0] = "Element #";
  ActionData->Cells[3][0] = "Time";
```

```
   for ( int i=0; i<6; i++ )
   {
      for ( int j=0; j<8; j++ )
         InitialState[i].Element[j].Bit = 0;

   }
   MakeCurrentState( InitialState );
   ActionData->Row = 1;


}
//---------------------------------------------------------------------------

void __fastcall TForm1::ActionDataClick(TObject *Sender)
{
   if ( dbPlay->Tag == 1 ) return;

   if ( ActionData->Row > 1)
   {
      if ( ActionData->Cells[1][ActionData->Row] == "" && ActionData-
>Cells[1][ActionData->Row-1] == "")
      {
         while ( ActionData->Row > 1 )
         {
            ActionData->Row--;
            if ( ActionData->Cells[1][ActionData->Row] != "" )
            {
               ActionData->Row++;
               break;
            }
         }
      }
   }
   FindCurrentState();
   MakeCurrentState(ElementTmp );

}
//---------------------------------------------------------------------------


void __fastcall TForm1::FindCurrentState()
{
   int ArrayNo, ElementNo;
   for ( int i=0; i<6; i++ )
      ElementTmp[i] = InitialState[i];
```

```
  for ( int i=1; i<ActionData->Row; i++ )
  {
    ArrayNo = ActionData->Cells[1][i].ToInt()-1;
    ElementNo = ActionData->Cells[2][i].ToInt()-1;
    ElementTmp[ArrayNo].Element[ElementNo].Bit =
!ElementTmp[ArrayNo].Element[ElementNo].Bit;
  }
}

void __fastcall TForm1::MakeCurrentState( ElementArray *Array)
{
  TSpeedButton * Tmp;
  for ( int i=0; i<6; i++ )
  {
    for ( int j=0; j<8; j++ )
    {
      Tmp = (TSpeedButton *)FindComponent( "sbA" + IntToStr(i+1) + "Element" +
IntToStr(j+1));

      if ( Array[i].Element[j].Bit == 1 && Tmp->Down == false )
      {
        Tmp->Glyph = blButtonImage->GetBitmap(10); //  Concave;
        Tmp->Down = true;
        continue;
      }
      if ( Array[i].Element[j].Bit == 0 && Tmp->Down == true )
      {
        Tmp->Glyph = blButtonImage->GetBitmap(11);
        Tmp->Down = false;
      }
    }
  }

  if( Display->ActiveDocument->Entities->get_Count() > 0)
    Display->ActiveDocument->Entities->EraseAll();
  for ( int i=0; i<6; i++ )
    DrawArray( i+1, Array[i] );
  if ( Timer1->Enabled == false )
    Display->Redraw();
}
void __fastcall TForm1::tPlayTimer(TObject *Sender)
{
  TSpeedButton * Tmp;
  tPlay->Enabled = false;

  int i=ActionData->Cells[1][ActionData->Row].ToInt();
  int j=ActionData->Cells[2][ActionData->Row].ToInt();

  Tmp = (TSpeedButton *)FindComponent( "sbA" + IntToStr(i) + "Element" +
IntToStr(j));

  Tmp->Down = !Tmp->Down;

  if ( Tmp->Down == true )
    Tmp->Glyph = blButtonImage->GetBitmap(10); //  Concave;
  else
    Tmp->Glyph = blButtonImage->GetBitmap(11);

  while( ReadyFlag == false )
    Application->ProcessMessages();
  ReadyFlag = false;
  RotateElement( i, j );

  ActionData->Row++;

  if( ActionData->Cells[1][ActionData->Row] == "" )
  {
    tPlay->Enabled = false;
    dbStopClick(dbStop);
    return;
  }

  tPlay->Interval = ActionData->Cells[3][ActionData->Row].ToInt();
  tPlay->Enabled = true;

}
//---------------------------------------------------------------------------

void __fastcall TForm1::VrDemoButton8Click(TObject *Sender)
{
  Close();
}
//---------------------------------------------------------------------------


void __fastcall TForm1::DEMO1Click(TObject *Sender)
{
  pDemo->Enabled = false;
  for ( int i=1; i<=1; i++ )
  {
    for ( int j=1; j<=8; j++ )
    {
         while( ReadyFlag == false )
            Application->ProcessMessages();
         ReadyFlag = false;
         RotateElement(i,j);
      }
      for ( int j=8; j>=1; j-- )
      {
         while( ReadyFlag == false )
            Application->ProcessMessages();
         ReadyFlag = false;
         RotateElement(i,j);
      }
   }
   pDemo->Enabled = true;
}
//---------------------------------------------------------------------------


void __fastcall TForm1::DrawArray(int ArrayNo, ElementArray Setting)
{
   double sx=0,sy=0,mx=0,my=0,ex,ey;
   double angle=0;

   sx = (( ArrayNo - 1 ) % 3 ) * 200;
   if ( ArrayNo > 3 )  sy = 200;

   for ( int i=0; i<8; i++ )
   {
      mx = sx + cos(angle)*16;
      my = sy + sin(angle)*16;
      if ( Setting.Element[i].Bit )
      {
         angle -= ElementAngle/180. * M_PI;
      }
      else
      {
         angle += ElementAngle/180. * M_PI;
      }
      ex = mx + cos(angle)*10;
      ey = my + sin(angle)*10;
      VDAddPolyline( MakeElementPoint( sx,sy,mx,my,ex,ey), i + 2 );

      sx = ex; sy = ey;
   }
}
```

```
void __fastcall TForm1::RotateElement(int ArrayNo, int Bit)
{

   vdPolyline * L;
   Variant P1,P2,P3;
   double Angle,x,y,RotatedAngle;
   L = (vdPolyline*)Display->ActiveDocument->Entities->get_Item((ArrayNo-1)*8+Bit-
1);
   P1 = L->GetVertexAt(0);
   P2 = L->GetVertexAt(1);
   P3 = L->GetVertexAt(2);

   x = P2.GetElement(0);
   y = P2.GetElement(1);
   Angle = Display->Utility->geomAngle(P2,P1) - Display->Utility-
>geomAngle(P2,P3);
   RotatedAngle = Display->Utility->geomAngle(P1,P2);
   if ( Angle < 0 )
      Angle += 2*M_PI;

   if ( Angle < M_PI )
   {
      x = x + ( cos(RotatedAngle)*cos(-ElementAngle/180.*M_PI) -
sin(RotatedAngle)*sin(-ElementAngle/180.*M_PI))*10;
      y = y + ( sin(RotatedAngle)*cos(-ElementAngle/180.*M_PI) +
cos(RotatedAngle)*sin(-ElementAngle/180.*M_PI))*10;
      for ( int i=Bit+1; i<=8; i++ )
      {
         Display->CommandAction->CmdRotate( ( Variant)Display->ActiveDocument-
>Entities->get_Item((ArrayNo-1)*8 +i-1),
         ((vdPolyline*)Display->ActiveDocument->Entities->get_Item((ArrayNo-1)*8
+Bit-1))->GetVertexAt(1),(Variant)(-2*ElementAngle/180.* M_PI));
      }
   }
   else
   {
      x = x + ( cos(RotatedAngle)*cos(ElementAngle/180.*M_PI) -
sin(RotatedAngle)*sin(ElementAngle/180.*M_PI))*10;
      y = y + ( sin(RotatedAngle)*cos(ElementAngle/180.*M_PI) +
cos(RotatedAngle)*sin(ElementAngle/180.*M_PI))*10;
      for ( int i=Bit+1; i<=8; i++ )
      {
        Display->CommandAction->CmdRotate( ( Variant)Display->ActiveDocument-
>Entities->get_Item((ArrayNo-1)*8 +i-1),
        ((vdPolyline*)Display->ActiveDocument->Entities->get_Item((ArrayNo-1)*8
+Bit-1))->GetVertexAt(1),(Variant)(2*ElementAngle/180.* M_PI));
      }
   }

   L->SetVertexAt(2, VP(x,y));
   Display->Redraw();
   Com1->PutChar( ArrayNo + '0');
   Sleep(5);
   Com1->PutChar( Bit + '0');

}
void __fastcall TForm1::VrDemoButton1Click(TObject *Sender)
{
   pDemo->Enabled = false;
   for ( int i=1; i<=6; i++ )
   {
      for ( int j=1; j<=8; j++ )
      {
         while( ReadyFlag == false )
            Application->ProcessMessages();
         ReadyFlag = false;
         RotateElement(i,j);
         Sleep(200);
      }
      for ( int j=1; j<=8; j++ )
      {
         while( ReadyFlag == false )
            Application->ProcessMessages();
         ReadyFlag = false;
         RotateElement(i,j);
         Sleep(200);
      }
   }
   pDemo->Enabled = true;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::VrDemoButton2Click(TObject *Sender)
{
   Display->CommandAction->Zoom( (Variant)WideString("E"),
(Variant)WideString("USER"), (Variant)WideString("USER"));
}
//---------------------------------------------------------------------------
```

```
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{

   MakeCurrentState( InitialState );
   Display->CommandAction->Zoom( (Variant)WideString("E"),
(Variant)WideString("USER"), (Variant)WideString("USER"));
   Display->CommandAction->Zoom( (Variant)WideString("S"), (Variant)0.5,
(Variant)0);
   Timer1->Enabled = false;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Com1TriggerAvail(TObject *CP, WORD Count)
{
   char tmp[100];
   Com1->GetBlock( tmp, Count );
   tmp[Count] = 0;

//   Memo1->Lines->Add( tmp );
   if ( (String)tmp == "OK" && ReadyFlag == false )
      ReadyFlag = true;
}
//---------------------------------------------------------------------------
```

BASIC Stamp Source Code

By Dodge Daverman & Don Jin Lee

```
'////////////////// Hardware Interface //////////////////////////////

DATAn     CON         13  'Bits are shifted out this pin # to TPIC2603.
CLKn      CON         15  'Data valid on rising edge of this clolck pin.

Delay     CON         3000  'This will be used to control the length of pulse to Nitinol

'================== Source and Sink Info

CLEAR     CON         %00000000  'This flag is used as a NULL flag to clear the contents
of flags


Drain_0   CON         %00000001  'These flags turn on the corresponding drains
Drain_1   CON         %00000010
Drain_2   CON         %00000100
Drain_3   CON         %00001000
Drain_4   CON         %00010000
Drain_5   CON         %00100000

'=======================  Variables ==================================================

' ////////////////// Input Variables from User

ArrayNo         VAR       Nib  'The Number of the array we are going to activate (1-6)
ElementNo       VAR       Nib  'The position of the element on the array (1-9)
i               VAR       Byte  'Used for initializtation
j               VAR       Byte

' //////////////////      Flags

CS              VAR       Nib  'Chip Select variable for Low Side Drivers.
SourcePin       VAR       Nib  'Which High Side Driver pin will be used to source the
current.
DrainFlag       VAR       Byte  'Bit pattern to be sent to the low side driver through
serial communication (shiftout)
ElementFlag     VAR       Byte    'Used to determine the current state of an individual
element
StateFlag       VAR       Byte  'Bit pattern that represents the configuration of the
array.  (1 = Bottom will be activated next)


State           VAR       Byte(6)
TmpArray        VAR       Byte(6)
Src             VAR       Byte(4)



Main:

i = 1
Src(0) = 2
Src(1) = 0
Src(2) = 4
Src(3) = 6

HIGH 8
HIGH 10
HIGH 12
HIGH 14

'DEBUG CR, "========== Shape Display Demo =========="
'DEBUG CR, "Please set the initial state flags."
'DEBUG CR, "(0) Top is ready for activation"
'DEBUG CR, "(1) Bottom is ready for activation"
'DEBUG CR, "Element 1 is left most bit.", CR

' TO DO: PROMPT USER TO INITIALIZE THE STATE OF THE ELEMENT
StateFlag = CLEAR

'DO
'  DEBUG CR, "Enter the State of Element ",DEC1 n, ": "
'  DEBUGIN BIN8 State(n-1)
'  DEBUG CR, "State", DEC1 n, " = ", BIN8 State(n-1)
'  n=n+1
'LOOP WHILE n<7
  State(0)=0
  State(1)=0
  State(2)=0
  State(3)=0
  State(4)=0
  State(5)=0

PAUSE 1000

Menu:
  '========== USE WHEN ENTERING QWERTY LETTERS
'  DEBUG CLS
  DEBUG "OK"
  DEBUGIN DEC1 i
  IF ( i =1 ) THEN
    FOR  i=0 TO 5
      DEBUGIN DEC TmpArray(i)
      DEBUG CR,DEC TmpArray(i)
    NEXT
    FOR  i=0 TO 5
      ArrayNo = i + 1
      FOR j=1 TO 8
        ElementNo = j
'        DEBUG CR, DEC i," ",DEC j
        IF ( (State(i) & (1<<(8-j))) <> (TmpArray(i) & (1<<(j-1)) )) THEN
          DEBUG CR,DEC ArrayNo," ", DEC ElementNo
'          DEBUG CR,DEC State(i)," ", DEC TmpArray(i)
          GOSUB TakeAction
        ENDIF
      NEXT
      State(i) = TmpArray(i)
    NEXT
  ELSE
    DEBUGIN DEC1 ArrayNo
    DEBUG DEC1 ArrayNo
    DEBUGIN DEC1 ElementNo
    DEBUG DEC1 ElementNo
    GOSUB TakeAction
  ENDIF

GOTO Menu

TakeAction :

  'debug ? Array_Number

  IF ( ArrayNo > 6 )THEN GOTO Menu
  IF ( ArrayNo < 1 ) THEN GOTO Menu
  IF ( ElementNo > 8 ) THEN GOTO Menu
  IF ( ElementNo < 1 ) THEN GOTO Menu

  'debug ? Array_Number

  ElementFlag = 1 << ( 8 - ElementNo )
  SourcePin = Src( (ElementNo-1)/2 )         ' HIGH side driver PIN

'  DEBUG DEC Src, CR

  LOOKUP ArrayNo, [0, 8, 10, 0, 12, 14, 0], CS    'SET THE CHIP SELECT (0 IS FOR EVERY
THIRD ELEMENT WHERE THE CHIP SELECT DEPENDS ON THE ELEMENT #)

'  DEBUG DEC CS,CR

  StateFlag = State(ArrayNo-1)      'SET THE STATE FLAG TO THE STATE OF THE CURRENT ARRAY
  State(ArrayNo-1) = State(ArrayNo-1) ^ ElementFlag  'TOGGLES THE STATE OF THE ELEMENT
THAT WILL BE ACTIVATED


  IF ( ArrayNo = 3 OR ArrayNo = 6 ) THEN

    IF (StateFlag & ElementFlag) = 0 THEN
      DrainFlag = Drain_4
    ELSE
      DrainFlag = Drain_5
    ENDIF

    IF ( ElementNo // 2 ) = 0 THEN
      IF ( ArrayNo = 3 ) THEN
```

```
        CS = 10
      ELSE
        CS = 14
      ENDIF
    ELSE
      IF ( ArrayNo = 3 ) THEN
        CS = 8
      ELSE
        CS = 12
      ENDIF
    ENDIF
  ELSE
    SELECT ArrayNo
      CASE 1
          CS = 8
      CASE 2
          CS = 10
      CASE 4
          CS = 12
      CASE 5
          CS = 14
    ENDSELECT

    IF (StateFlag & ElementFlag) = 0  THEN
      IF ( ElementNo // 2 = 1 ) THEN
        DrainFlag = Drain_0
      ELSE
        DrainFlag = Drain_2
      ENDIF
    ELSE
      IF ( ElementNo // 2 = 1 ) THEN
        DrainFlag = Drain_1
      ELSE
        DrainFlag = Drain_3
      ENDIF
    ENDIF

  ENDIF

'  DEBUG CR, HEX DrainFlag


ActivateElement :
  'TURN ON THE SOURCE PIN
  HIGH SourcePin

  'TURN ON THE APPROPRIATE DRAIN
  LOW CS
  SHIFTOUT DATAn, CLKn, MSBFIRST, [DrainFlag]
  HIGH CS

  PAUSE 100 ' Delay  'KEEPS THE DRAIN ON FOR PULSE_LENGTH MILLISECONDS

  'TURN OFF ALL DRAINS
  LOW CS
  SHIFTOUT DATAn, CLKn, MSBFIRST, [CLEAR]      'TURNS OFF ALL DRAINS
  HIGH CS

  'TURN OFF SOURCE PIN
  LOW SourcePin

  ' UPDATE THE STATE FLAG WITH THE CURRENT STATE OF THE ELEMENT
  StateFlag = StateFlag ^ ElementFlag     'Toggles the bit in the bit pattern
corresponding to the current element

'  DEBUG CR, "State flag is " ,BIN8 StateFLag


'  GOTO Menu 'ActivateElement

RETURN
```