Inspection And Exploration Gas Pipelines: Snake Robot

By

MOHD HAFIZ BIN ROSLAN
3914

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

INSPECTION AND EXPLORATION GAS PIPELINES: SNAKE ROBOT

by

Mohd Hafiz Bin Roslan

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Mr Lo Hai Hiung
Project Supervisor

Lo Hai Hiung
Lecturer
Electrical & Electronic Engineering
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh
Perak Darul Ridzuan, Malaysia

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Mohd Hafiz Bin Roslan

ABSTRACT

Mobility is a primary concern for robotics applications. Snake-like locomotion is a stable and tough but often overlooked mobility method. Snakes can use their entire body for propulsion, creating a larger surface area and providing greater grip. The primary objective is to come out with a construct of six-segment snake robot. The robot will be able to move simultaneously by moving its body sections. The robot is control and designed based on PIC16F84A microcontroller. The cores of the project are the applications of electronic fundamentals in circuit design and programmed development-using Basic in delivering sinusoidal motion. The circuit is designed using Eagle and fabricated as to provide the means of interconnecting between servomotors and the circuit to deliver a movement. The project methodology stresses on the research and analysis, circuit design, circuit modeling, micro controller programming, circuit fabrication and finally implementation of mechanical parts. The end of system comprises of the hardware for the micro controller, servos motor and fabricated circuit in order to achieve locomotion.

ACKNOWLEDGEMENTS

This project would have been impossible if not been for the help, support and advice from a number of people. Thus, I would like to take this opportunity to express my profound thanks and gratitude to each and every one of them.

First of all, I would like to thank my supervisor, Mr Lo Hai Hiung, for agreeing to supervise my final year project. Thanks too his constant guidance and advice, the project is able to amount to success. There were many times where Mr Lo Hai Hiung advice was vital for the continuation of progress. Mr Lo Hai Hiung recommendations and suggestions have been very helpful in inspiring the project.

Secondly, I would like to thank Mr Patrik Sebastian, for his advice and guidance throughout the development of the project.

Thirdly, I would like to express gratitude to my course mates for helping me out in buying materials and providing transportation to get things done regarding the project. They are namely Mr Wan Mohd Azfar, Mr Mohammed Yasser Arafat, Mr Syed Firdaus and also Miss Nurul Farashikin. Besides that, their moral support and endless care and valuable opinions and knowledge towards the development of the structure are very much appreciated.

Last but not least, I would also like to thank other individuals who have been both directly and indirectly helpful, especially the lab technicians, Mr Isnani and Ms. Hawa for opening the labs and providing supplies from the stores.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PCB	Printed Circuit Board
PIC	Programmable Interrupt Controller
PWM	Pulse Wave Modulation
SERVO	ServoMotor

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Commonly, robotic vehicles use a wheel and axle propulsion system, but this is often debilitating when traveling on variable ground. The simpler and often overlooked is a snake sinusoidal motion, which is a less obvious solution, but has great advantages when navigating variable ground. Snakes can use their entire body for propulsion, creating a larger surface area and providing greater grip. Their low center of gravity creates stability, lacking in legged and wheeled systems.

1.2 Problem Statement

1.2.1 Problem Identification

The PIC16F84A microcontroller is first to be modeled on the project board as to ensure there will be no faulty made in the design. Besides, if any additional components required to be embedded on the circuit, they can be identified and easily added before the finalized circuit is soldered on the Vero board. Programming the micro controller is the most critical part of the project. It is essential in enabling the system to produce a sinusoidal movement. The thorough study and good understanding on BASIC programming are necessary as to guarantee the smoothness while programming the microcontroller.

1.2.2 Significant of the Project

The snake robot that will be built are programmed consists of six segments and a head, which each segment being powered by a R/C servo. The robot is controlled by a microchip PIC16F84A microcontroller. The microcontroller is used to sequence the movement of each of the snake's body section via servos.

1.3 Objective and Scope of Study

This project is conducted to construct a six-segment snake robot, and thus is conducted with several objectives to be met, which are:

1. To develop a Basic Brain Module, which has I/O Interfaces with Brain.
2. To achieve a motion algorithm that moves in a horizontal motion and the next segment moves in a vertical motion.

1.3.1 The Relevancy of the Project

The relevancy of the project is viewed from two different perspectives, by which include hardware design and snake application.

1.3.2 Hardware Design

The microprocessor is facilitated in the circuit as an alternative to having a large number of electronics gates on the circuit. The reduction or elimination of gates is necessary o reduce the component count and the circuit size. Furthermore, this can ensure that the final product will be medium-sized and compact.

1.3.3 Snake Application

With snake application system, user can inspect to see when to start gas pipes repair working or replacement. Inspection of gas pipes of public installation such as new city or industrial estate before take-over or hand-over public installations.

1.3.4 Feasibility of the Project within the Scope and Time frame

In general, all scopes covered to complete the project are feasible for a final year student. The allocated time frame of approximately one year is sufficient to carry out the entire task required in the project. **Appendix G & H** summarized the allocated frames for all tasks performed throughout the one and two semester in a Gantt chart.

CHAPTER 2

LITERATURE REVIEW/ THEORY

2.1 Body Structure

Snake-like locomotion is often overlooked as a method of robotic mobility. Snake-like locomotors possess many advantages behavior such as stability and adaptability [10]. In recent years snake-like robot development has grown with the new of knowledge about biological snakes. Biological systems provide as great inspiration for robotic creation. [4]. Greater knowledge about biological systems facilitates more accurate and efficient robotic systems. The key to success is the understanding the target biological system. It is important to understand the interaction between structure and function.

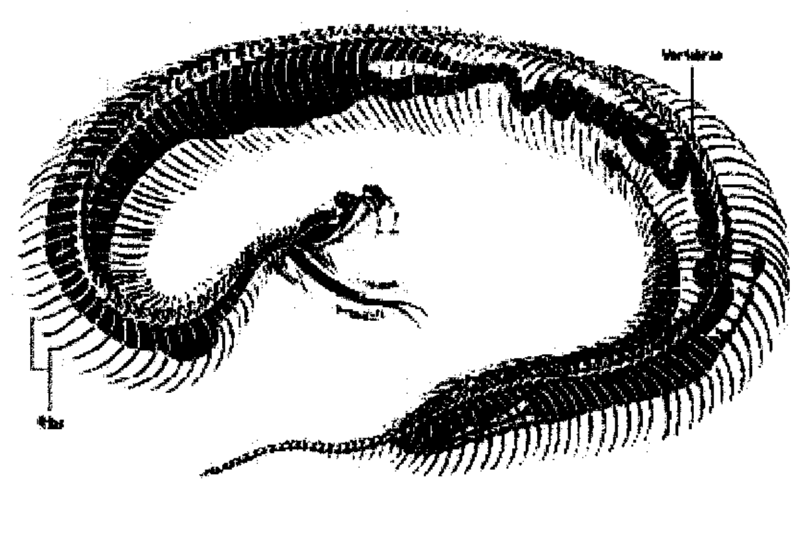


Figure 1 : Snake Skeletal Structure [11]

Snakes lack appendages simplifying their body structure. It has a low center of mass since its body stays in close contact with the ground. This creates more stability since the snake cannot fall over while moving like walking animals. The constant contact with the ground provides excellent traction. In combination with body positioning and the grip from traction, the snake can propel forward by cyclically changing its body position and manipulating its contact with the ground. The snake obtains mobility by repeatedly altering its body shape. Motion patterns form by repeating certain shapes called "gaits," a term borrowed from legged motion, identifying a certain alternating pattern [24].

2.2 Gaits

Biological snakes can characterize by several type of movements. There are lateral undulation, concertina, and side winding, among others [11]. Snakes can manipulate its entire body shape and is able to achieve directional movement.

2.2.1 Lateral Undulation

Commonly, a biological snake used the lateral undulation gait to produces propulsion and at the same time moving body sections. The sections continuously move from side to side perpendicular to the direction of forward motion. This repetition, as a directional vector has both a tangential and a normal component relative to the forward direction. The lateral, side-to-side, direction is defined as normal to the forward direction. By turning over a positive, usual direction to one side and a negative direction to the other, the net result of the lateral oscillation cancels the normal force. The tangential components for both sides are in the same direction, parallel to the direction of forward motion. The tangential force created by these components drives the body forward. The motion requires three points of contact. Two points for forward pressure and a third for balance. Dependent on sliding friction, lateral undulation is not successful on low-friction surfaces. The motion is less effective with shorter body lengths and heavy bodies. [8]

2.2.2 Concertina

Unlike the continuous, simultaneous body movements in lateral undulation, the concertina gait uses a progressive, body extension pattern. The body starts compressed, folded in a position similar to an accordion. Extending a front section, the snake reaches forward a distance, while the back sections remain stationary. The stationary sections provide a foundation for the moving section. The moving sections use the foundation for leverage to extend forward. The extension is undone, as the snake begins to refold its body, by drawing its back section forward. In this phase the

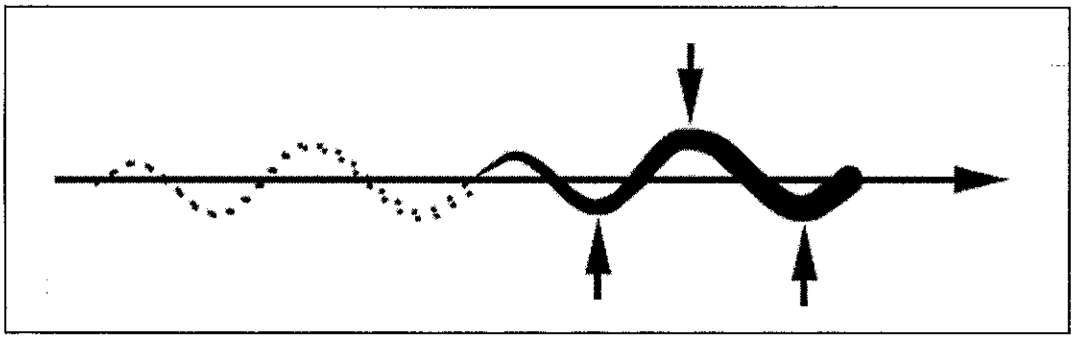


Figure 2 Lateral Undulation Motion
Source: http://www.cs.plu.edu

front section acts as the foundation, while the back section is in motion. The pattern results in a series, alternating between pushing against a back foundation and pulling against a front foundation. Static friction is the key; thus, this gait is more useful on low-friction surfaces [11].

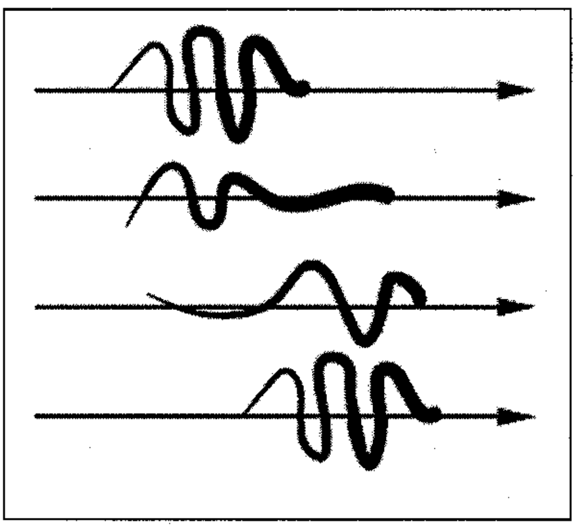


Figure 3 : Concertina Motion
Source: http://www.cs.plu.edu

2.2.3 Side Winding

Most useful on low-friction surfaces, the side-winding gait utilizes continuous waves of lateral bending. Only two points of contact are maintained with the ground. The segments not in contact with the surface are lifted and move to the side. They then become the new contact points. The previous contact point is then lifted and moved. Repeating this pattern, the snake moves in a direction to its side. Since sections must be lifted, the snake moves in both horizontal and vertical planes. Lateral undulation and concertina only necessitate horizontal mobility; thus, the side-winding gait requires more complex muscular and skeletal structures to facilitate the two degrees of freedom [11].

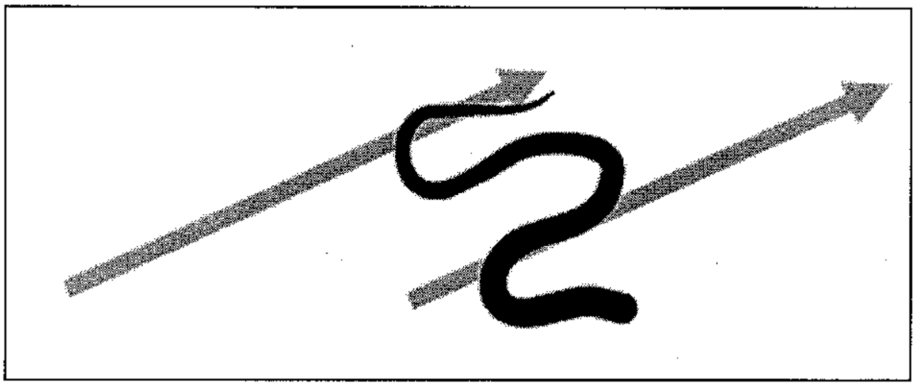


Figure 4 : Side Winding Motion

Source: http://www.cs.plu.edu

2.3 Snake-Like Robots

Robotic automation is very desirable in industrial sectors where multitasking task are common. There are many robotic arms, referred to as manipulators, have been developed for assembly lines and precision processes. These arms are usually multi-link actuating devices, very similar to a biological snake's structure. In the beginning 1970's, the work of Shiego Hirose and his Active Cord Mechanisms (ACM), which referred shown in Figure 5. Hirose developed his description of an effective with his formulation of the "serpenoid curve." [12]

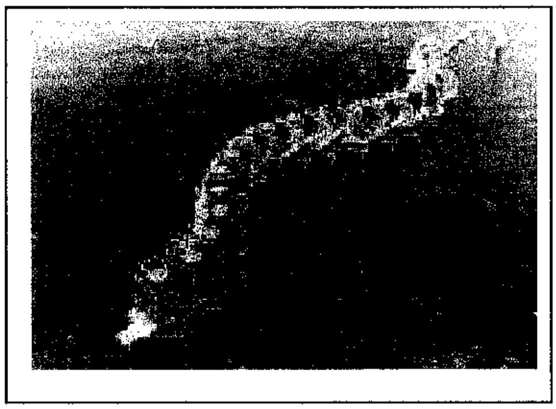


Figure 5 : Hirose's ACM III
Source: www.japaninc.com/article.php?articleID=963

Figure 6 : GMD Robotic Snake
Source: citeseer.ist.psu.edu/400188.html

Gavin Miller developed several snake-like robots, which improved biological accuracy. [19] Miller used a compound, acuating joint, where a single joint can move in 2 degrees of freedom. This was accomplished by combining two actuators working together to produce a lateral bend and against each other to create a ventral bend. Merging the degrees of actuation, the discontinuity was decrease, making the robot more snake-like.

2.4 PIC16F84A

2.4.1 Introduction

The PIC16F84A employs an advance Reduced Instruction Set Computer (RISC). This microcontroller had enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with a separate 8-bit data bus. The two-stage instruction pipeline allows all instruction to execute in a single cycle except for program branches, which is required two cycles. A total 35 instruction are available with an additions of large register set is used to achieve a very high performance level [1].

2.5 The Parallax Continuous Rotation Servo

2.5.1 Introduction

The Parallax Continuous Rotation servo is deal for robotic products that need a geared wheel drive or other projects that require a 360-degree rotation geared motor. The Parallax Continuous servo output gear shaft is a standard Futaba configuration. The servo can be adjusted with a small Phillips screwdriver if the unit becomes out adjustment on its center set point [2].

CHAPTER 3
METHODOLOGY/PROJECT WORK

3.1 Procedure Identification

In ensuring the smoothness of project execution, all required procedures are identified. **FIGURE 7** of the following page outlines the main approaches in accomplishing the project. All procedures are carried out subsequently at all times. If necessary, the previous procedure is revised as to do any modification on the project.

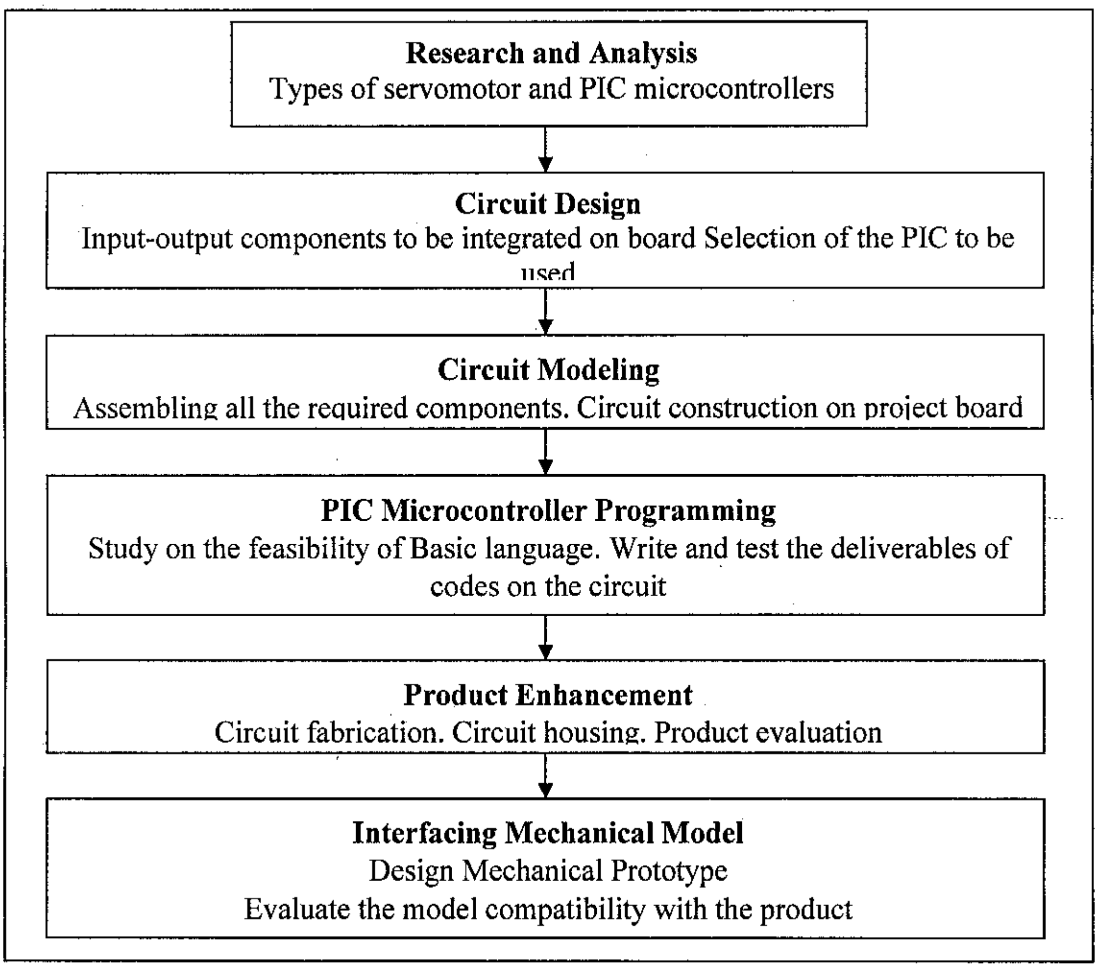


Figure 7 : Stages of procedure in delivering the project

3.2 Tools Required

Several engineering tools have been utilized in working out with the project. The tools used include EAGLE layout Editor 4.13, MicroCode Studio and WARP 13.

3.2.1 EAGLE Layout Editor 4.13

Eagle Layout Editor can be considered as a complete as a complete tool for circuit design where the schematic can be drawn and the PCB automatically routed from designed schematic. All components used in the circuit are available in its library including PIC 16F87X. The only limitation with the tool is that the maximum size of PCB board allowed to be laid out is only 4"X3.6. As more than one unit of the receiver needs to be produced, the circuit had been transformed into PCB layout. The layout is further edited to ensure that the PCB produced is of the smallest size possible [21].

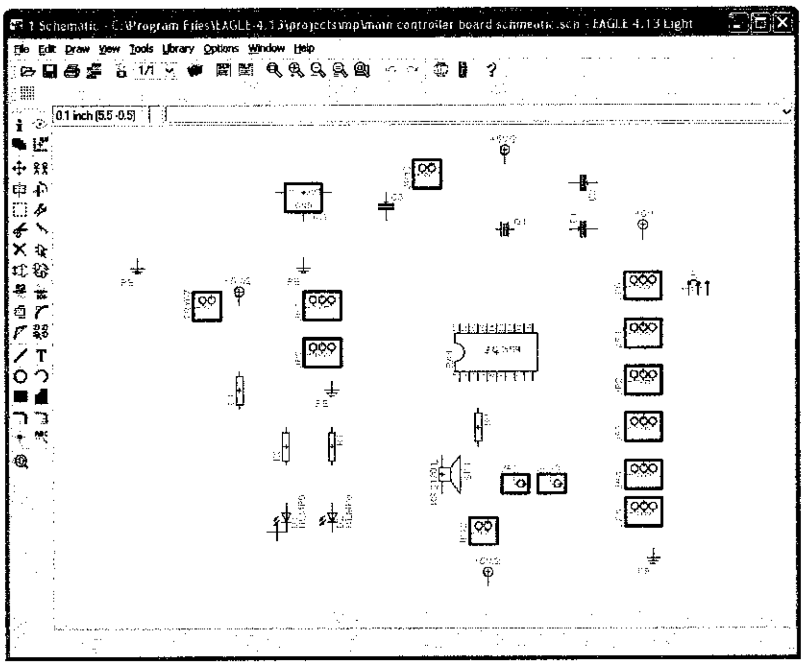


Figure 8 : Eagle Layout Editor program window

3.2.2 MicroCode Studio Visual Integrated Development Environment

Mecanique's MicroCode studio is a powerful, visual integrated development environment (IDE), with an in-circuit debugging (ICD) capability designed specifically for microEngineering Labs's PicBasic Pro Compiler. The MicroCode Studio user is shown in **Figure 9**. This studio makes programming PIC microcontroller very easy with a one-button process compiling, assembling, and programming. The main editor provides full syntax highlighting of code, with context-sensitive keyword help and syntax hints. The code explorer allows an automatically jump to include files, defines, constants, variables, aliases and modifiers, symbols, and labels that are contained within search and replace features. It also gives the ability to identify and correct compilation and assembler errors. MicroCode Studio also lets to view serial output from microcontroller. It includes keyword-based context-sensitive help, and it also supports MPASM and MPLAB [22].

It is easy to set up compiler, assembler and programmer options, or let MicroCode Studio do it with its built-in autosearch feature. MicroCode Studio has support for MPLAB-dependet programmers, such as PICStart Plus. Compilation and assembler errors can be easily identified and corrected using the error results window. Simply click on a compilation error and MicroCode Studio will automatically take to the error line. MicroCode Studio even comes with a serial communications window, allowing debugging and viewing serial output from microcontroller [22].

With MicroCode Studio, the preferred programming software from within IDE. This enables to compile and then program microcontroller with just a few mouse clicks. MicroCode Studio also supports MPLAB-dependent parameters [22].

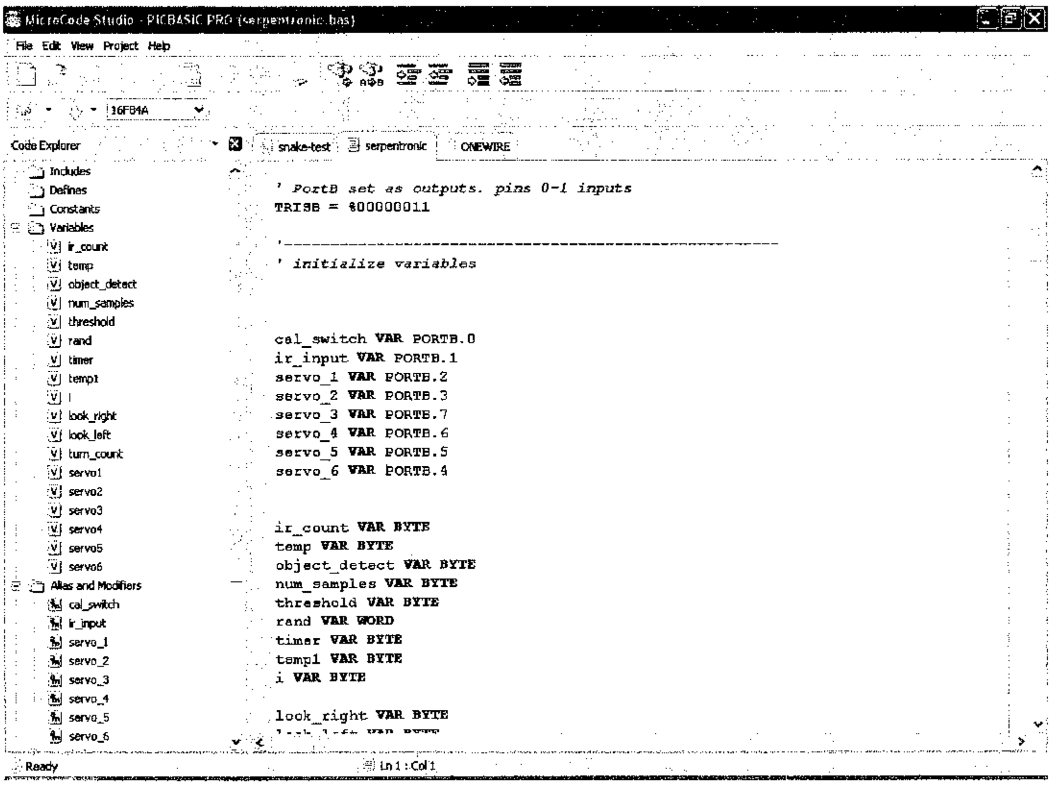


Figure 9 : MicroCode Studio program window

3.2.3 WARP 13

In succession to the HEX file generation, the HEX code can be burned into the PIC using a chip burner using WARP 13 program [23]. This is the vital part of the microcontroller programming as to ensure its capability to control the circuit system as desired. Before burning any particular program into the PIC, its EEPROM should be erased or blanked. Once the program button is pressed, the program will be written into the PIC memory. **Figure 10** shows the chip burner used for program burning.

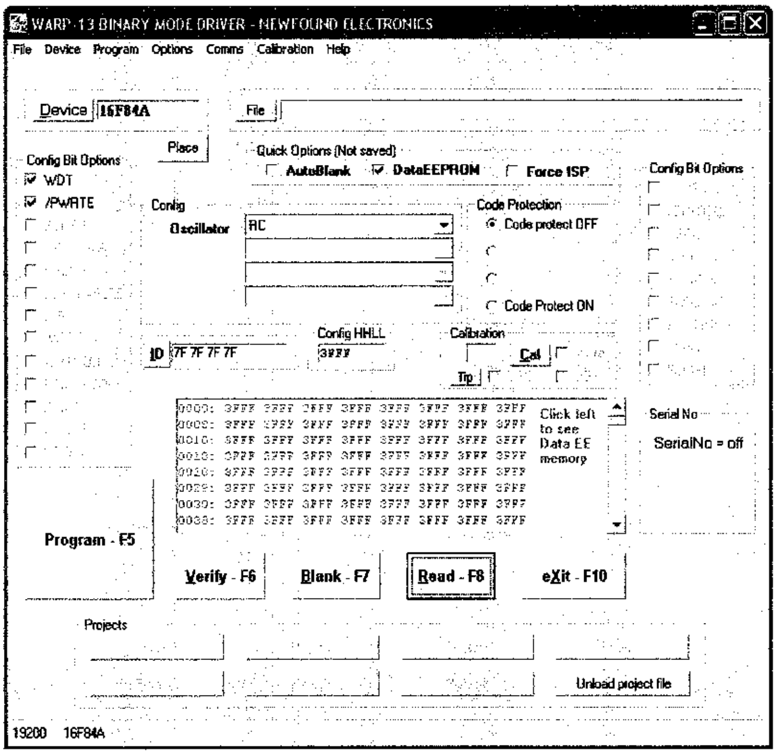


Figure 10 : WARP 13 program window

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Findings

In general, the construction of the robot snake body and the controller board had been successfully produced. Several test have been performed on the device, by which have given the expected results. The Gantt chart enclosed in **Appendix G & H** shows the project planning throughout the first semester and second semester.

4.1.1 Mechanical Construction

The construction of the snake robot begun with the mechanical construction of the body. The parts needed for the mechanical construction listed on the **Appendix E.** The body is constructed using 1/16-inch thick flat zinc.

4.1.1.0 Construction The Body Sections

The construction started by cutting six pieces of the 1/16-inch thick flat zinc to a size of 7-1/2 inches X 2-1/2 inches. These pieces would be identified as piece A of each the body sections. According to the **Figure 11**, the diagram showed the six pieces dimensions to be cut. When the pieces were cut, the 5/32-inch drill bit to drill the holes. Any rough edges from the pieces were file. Each piece is bend in a table vise or on the edge of a table, as indicated. Each of the six pieces looked like the one pictured in **Figure 12**.

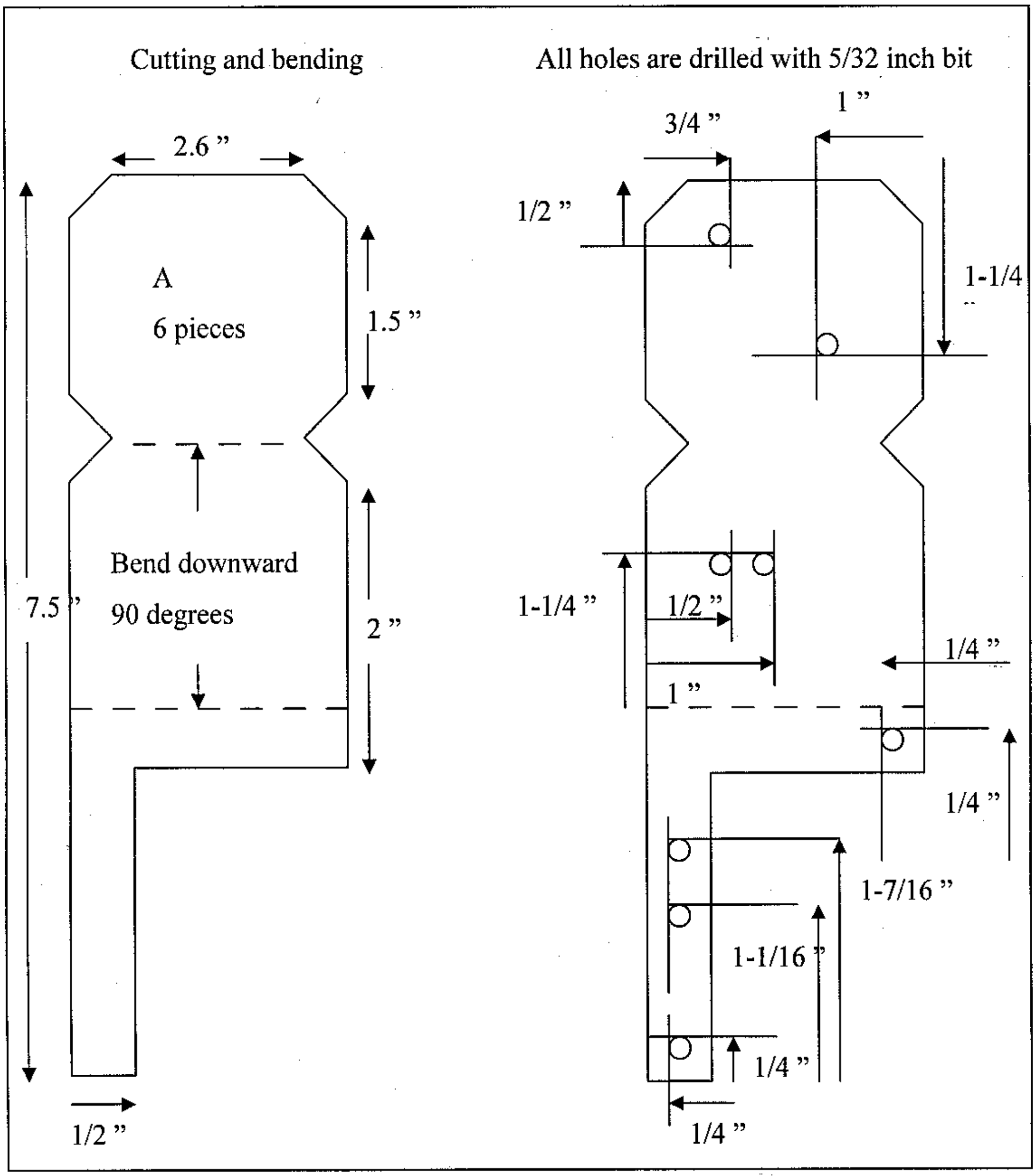


Figure 11 : Cutting, bending, and drilling for body piece (A)

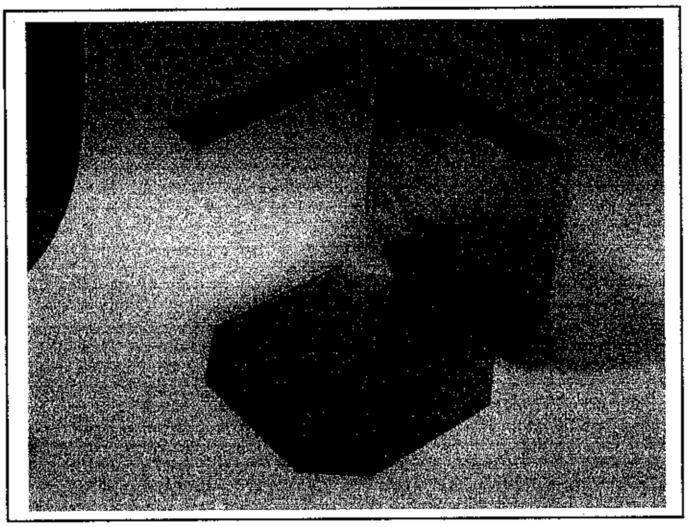


Figure 12 : Finished body part (A)

The next piece that would make up each of the body sections is also cut from 1/16-inches thick flat zinc. Six pieces were cut to a size of 2-1/2 inches X 5-3/4 inches each. These pieces would be identified as a piece of each of the body sections. According to the **Figure 13**, the diagram showed the six pieces dimensions to be cut. After the pieces were cut, a 5/32-inch drill bit to drill the holes, as indicated in the diagram. Any rough from the pieces were file. Each piece is bend in a table vise or on the edge of a table, as indicated. Each of the six pieces looked like the one pictured in **Figure 14**.

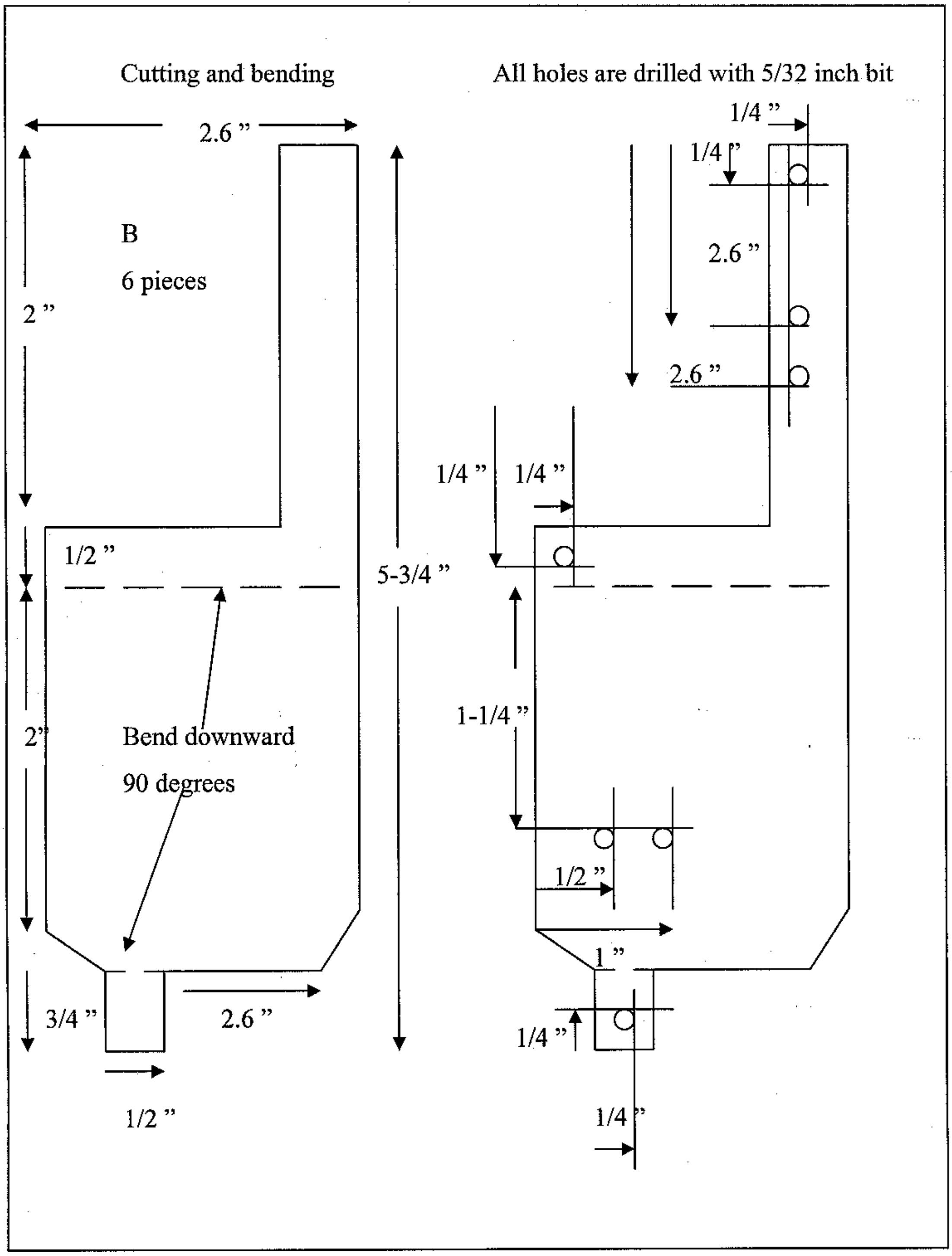


Figure 13 : Cutting, bending, and drilling diagram for body piece (B)

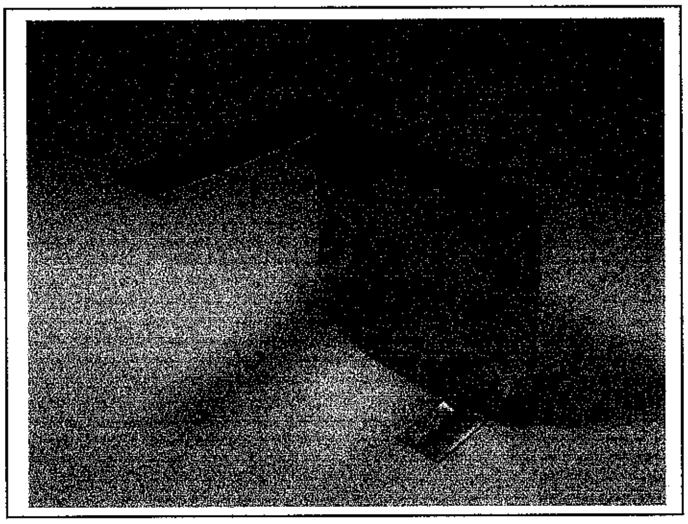


Figure 14 : Finished body part (B)

Pieces A and B were attached together using three 6/32-inch X ½-inch machine screws and locking nuts, as shown in **Figure 15**. The piece B is attached so that it is positioned on top of piece A. The above procedures are continuing until all six segments are
completed.

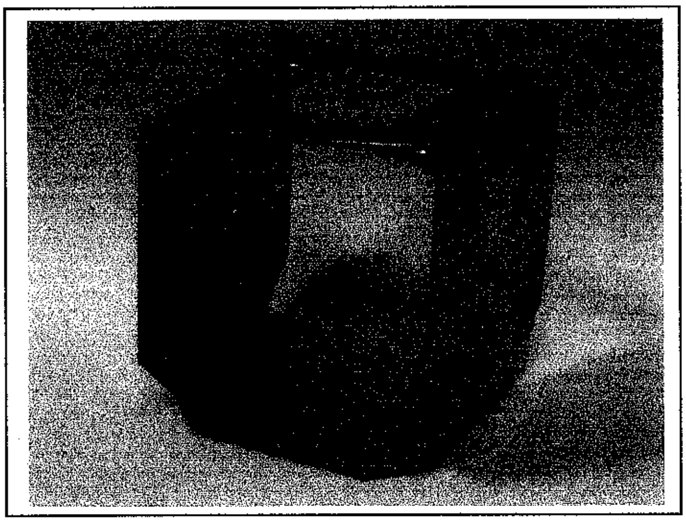


Figure 15 : Completed snake body segment made up of pieces A and B.

4.1.1.1 Construction The Tail Sections

The snake robot will need a tail that will be used to brace the rear end of the body and provide friction when the robot is moving forward and turning, as well as for the aesthetic purpose of completing the body. The tail section is constructed using 1/16-inch thick zinc stock. It is cut into a piece 2-1/2 inches X 8-1/2 inches. Any rough edges is filing and place the piece on a table. Then is drill the mounting holes as indicated, using 5/32-inch drill bit. The zinc then bends in a vise or on the edge of a table. The finished tailpiece is shown in **Figure 16.**

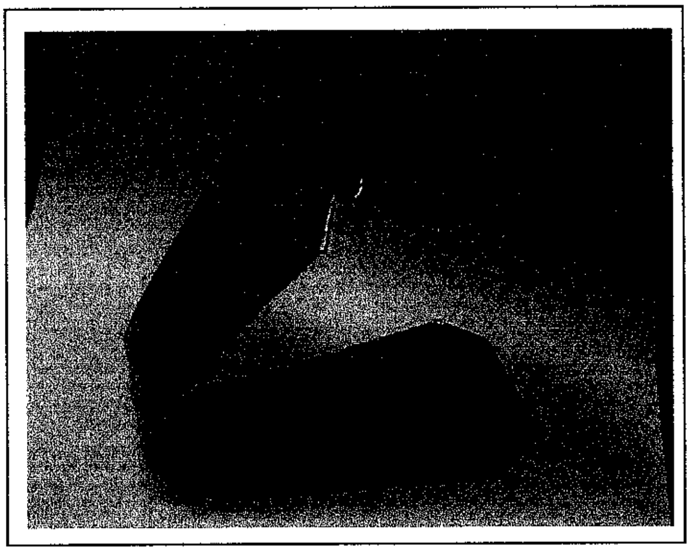


Figure 16 : Finished tail section

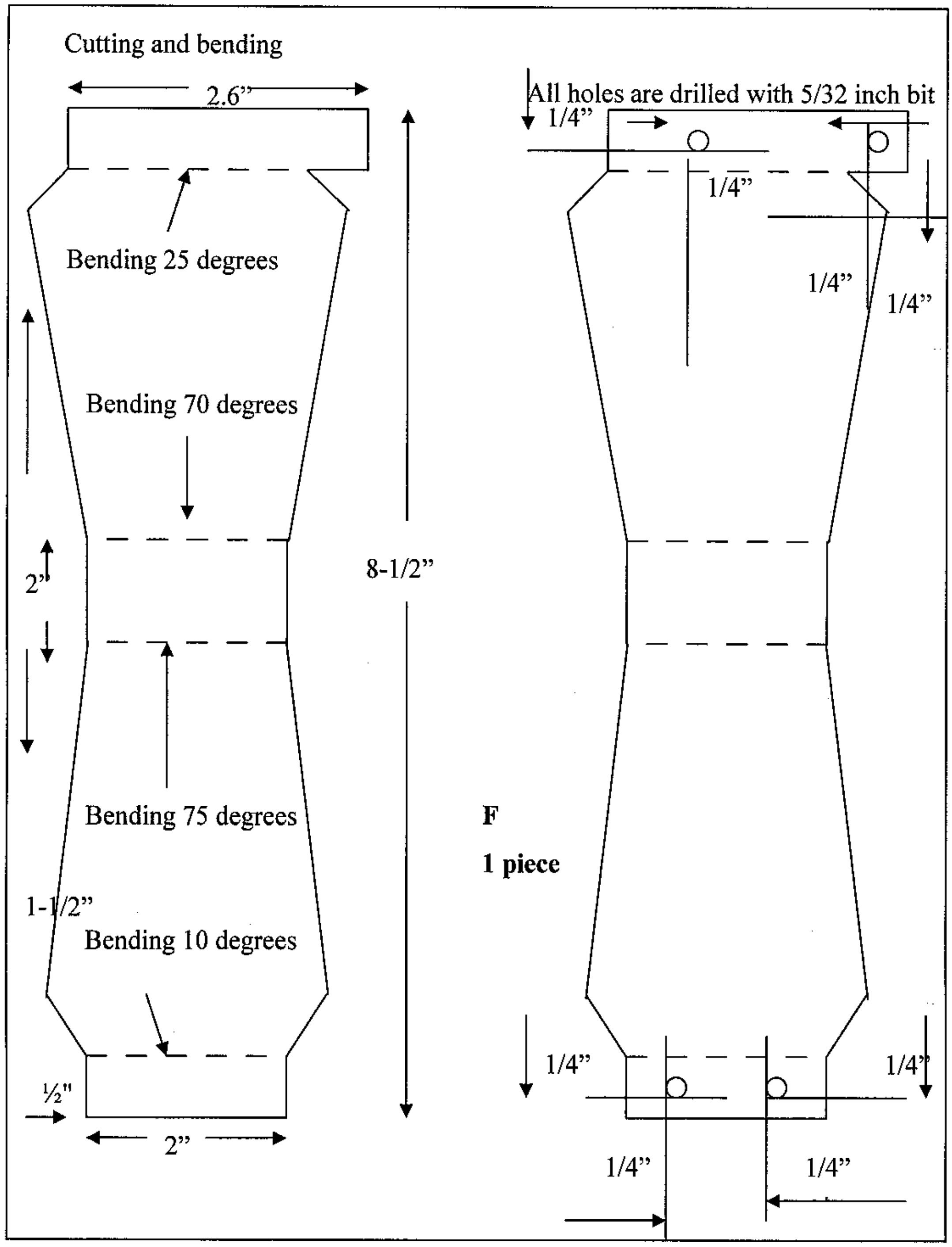


Figure 17 : Cutting, bending, and drilling diagram for snake's tail section

4.1.1.2 Construction The Head Sections

The snake's head will house the controller board that will sequence all of the servos in each body section. It cut a piece 1/16-inch thick zinc to a size of 3 inches X 6-1/4 inches. Then it cut, drill and bend the piece as shown in **Figure 18**. The finished piece, labelled G, is shown.Then it cut two pieces of 1/16-inch zinc to a size of 1 inch X 3-1/2 inches. Then it bend and drill each piece according to the dimension as shown in **Figure 19**. These two pieces are labeled I. The finished figures are shown and will be used as the side supports for the robot's head.

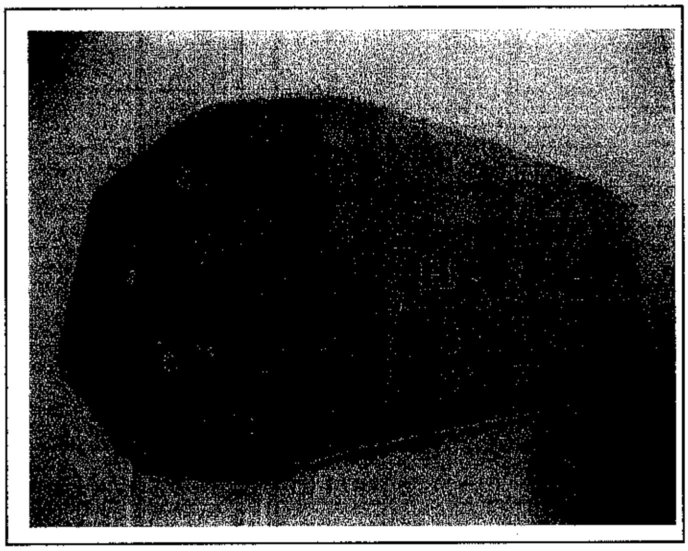


Figure 18 : Finished top head piece H

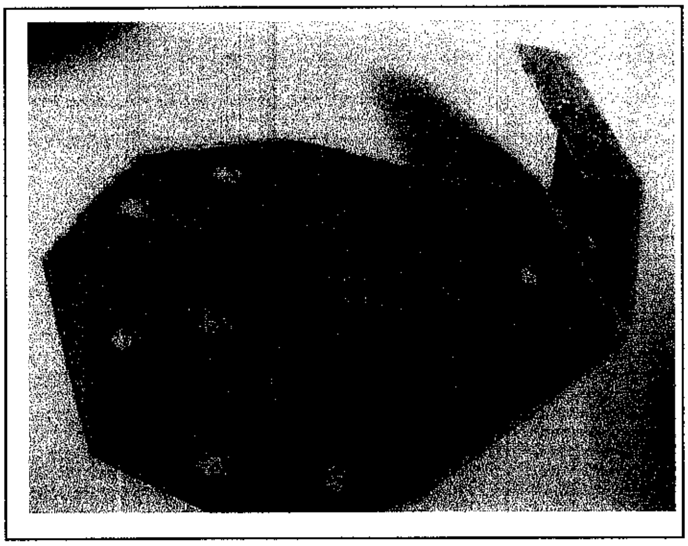


Figure 19 : Finished bottom head piece G

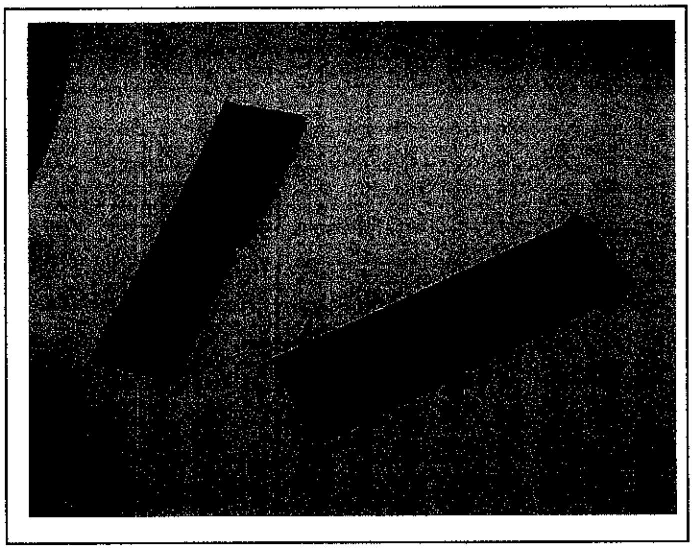


Figure 20 : Finished head support pieces labeled I

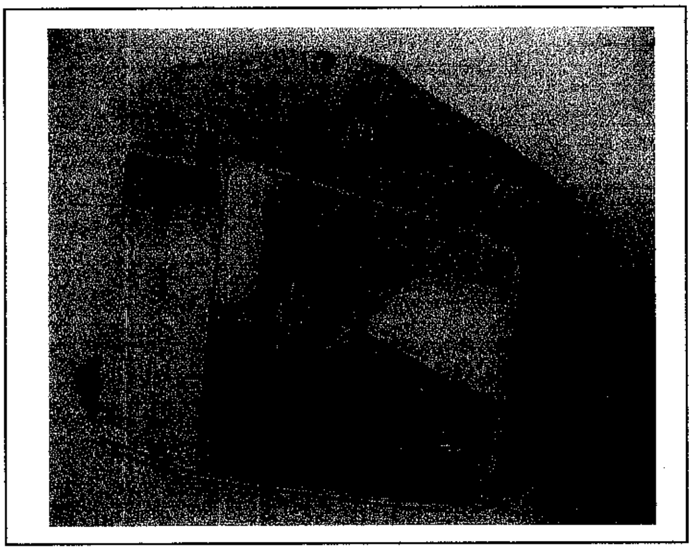


Figure 21 : Completed Head Assembly

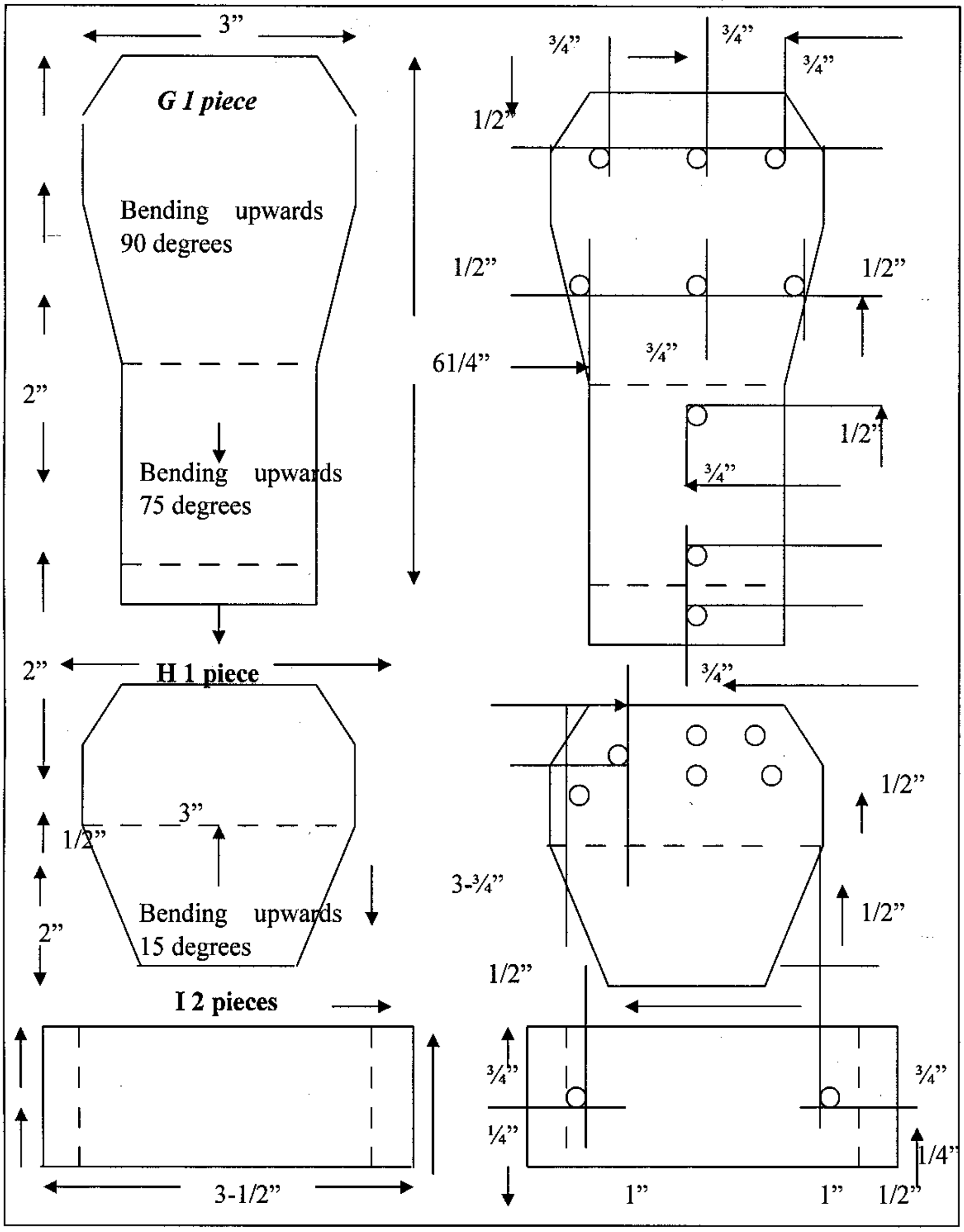


Figure 22 : Cutting, bending, and drilling diagram for bottom head piece G, top head piece H, and head support pieces labeled I

4.1.2 Electrical Construction

Construction of the Robot Main Controller Circuit

According to the **Appendix F**, lists all of the parts necessary to build the controller board. All of the robot's functions are controlled by a Microchip PIC 16F84A microcontroller. The microcontroller is an entire computer on a chip and makes it possible to eliminate a large amount of hardware that would otherwise be required [7]. The micro controller serves as the robot's "brain", controlling and managing all functions, sensors, and reflexes. The 16F84A microcontroller that the author is using will be clocked at 4 MHz and operates on a 5-Volts DC supply, produced from a 78L05 voltage regulator, with the source being a 9-volt battery. Each of the six servos used to move the body sections be powered by a separate 6-volt DC power source. The 6-volt power source is made up of the individual 3-volt battery packs in each of the body sections. Based on the schematic shown in **Figure 23**, the input/output (I/O) lines are used to control the six servos, turn on two light-emitting diode (LEDs), and output sound to a piezo speaker.

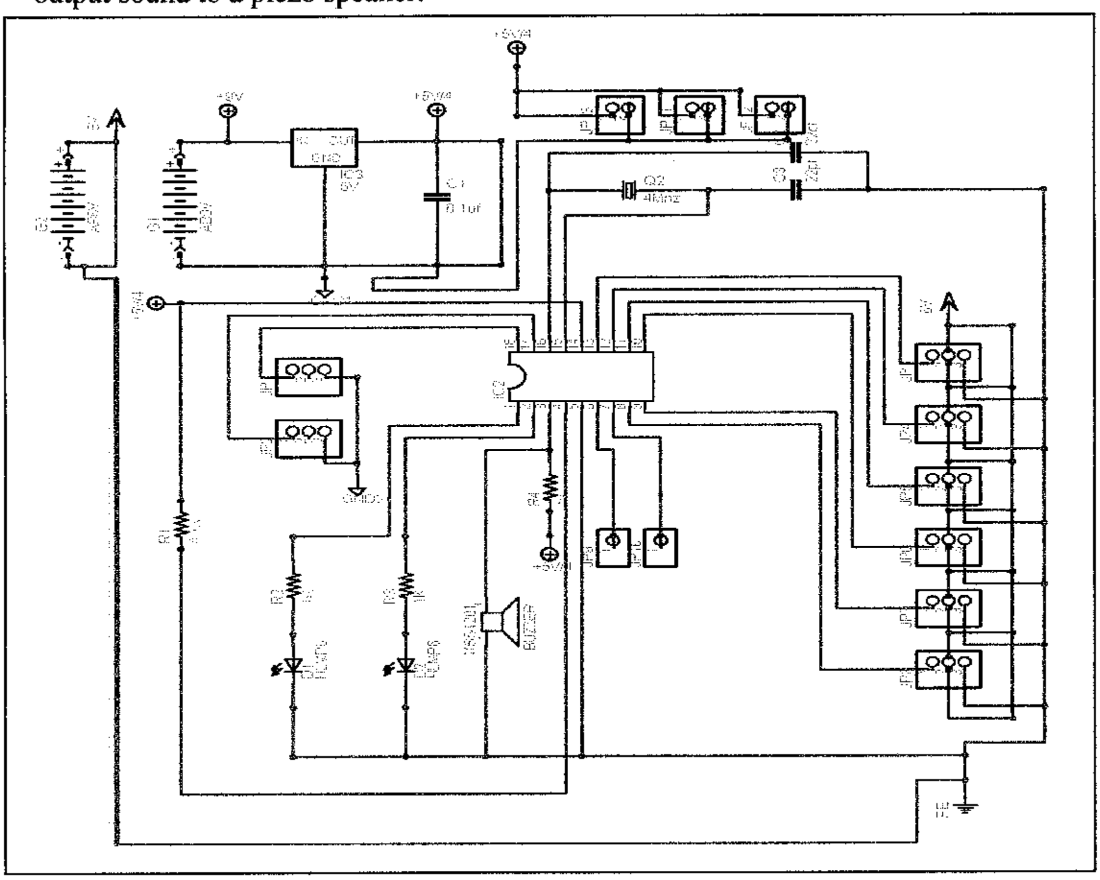


Figure 23 : Main Controller board Schematic

Circuit Modeling

During circuit modeling, the voltage is obtained from battery packed 3V and 9V. **Figure 24** exhibits the assembled components of the snake main controller board. Only one project board is used to assemble all required components for the main controller board. The jumpers are used, as there is limited space to allow all resistors to be connected in series with the respective microcontroller pins.

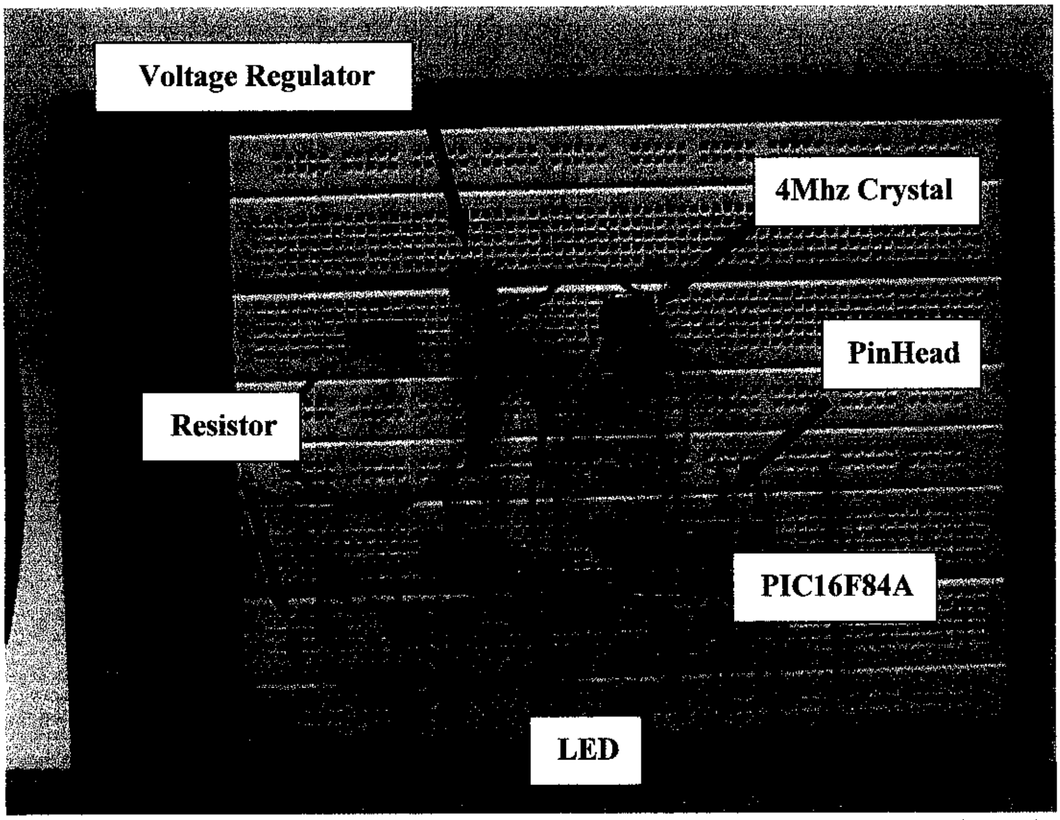


Figure 24 : Main Controller Board Implementation

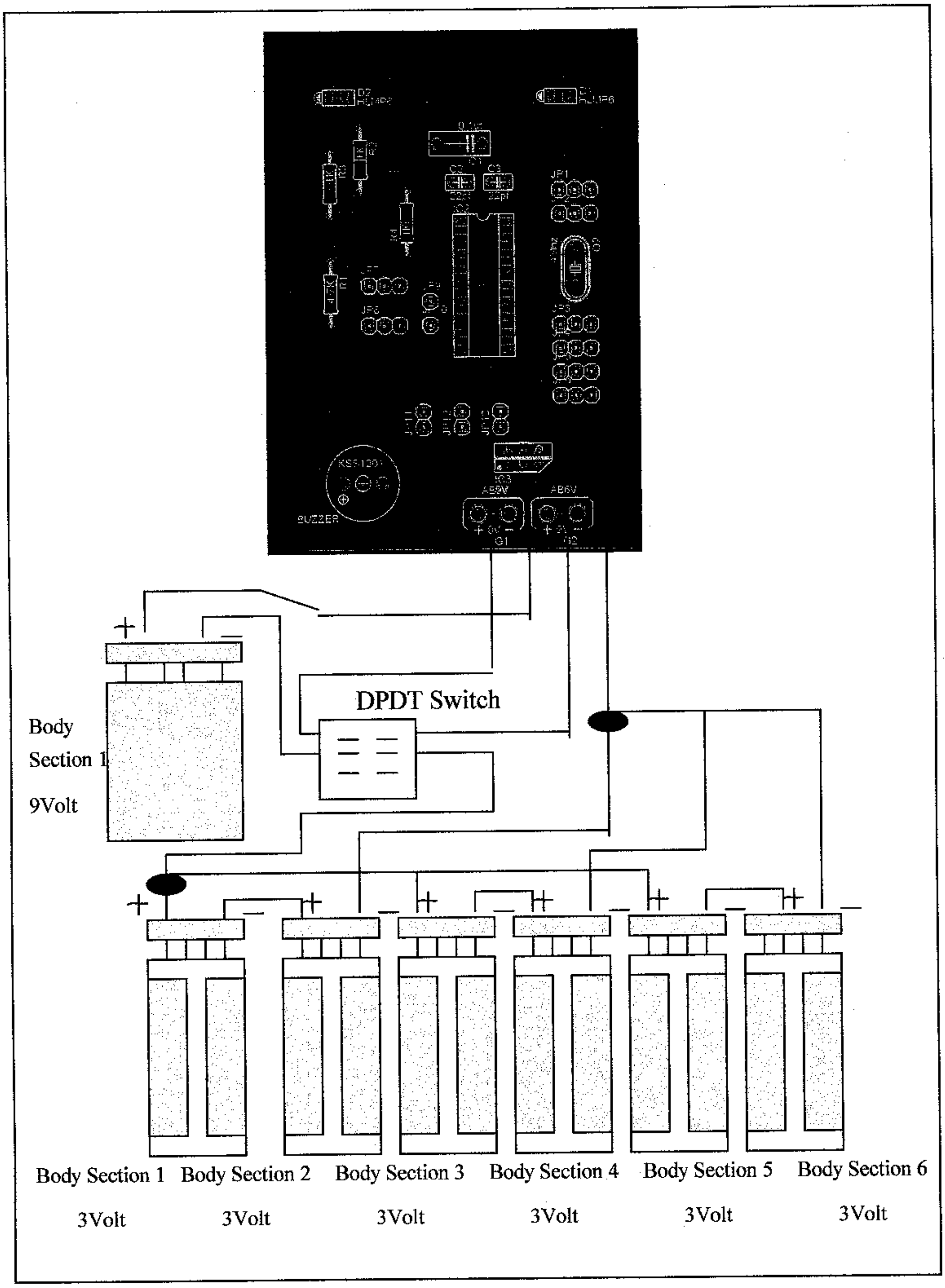


Figure 25 : Electrical Wiring Diagram

4.1.3 Motion Algorithm Design

The next task will be to coordinate the movement of each of the snake's body segments to achieve locomotion. To produce a forward movement, the snake will move its body in a sine wave pattern vertically, with slight side-to-side movement of the horizontal segments. The use servos makes this sort of programming easy because all that is needed to coordinate this pattern is to give each of the servos two sets of movements positions. The body segments will move through the complete range of motion between the two sets of points determined by the position values. This means that the author really only need to set the servo position for all of the servo twice, and then repeat the pattern to get the snake to move forward. The same holds true when sequencing the servos and body segments for a left or right turning movement. **Figure 26 & 27** shows the pulse out values for the extreme and middle positions, along with the micro controller port address for each servo. This information will be needed when putting the control program together.

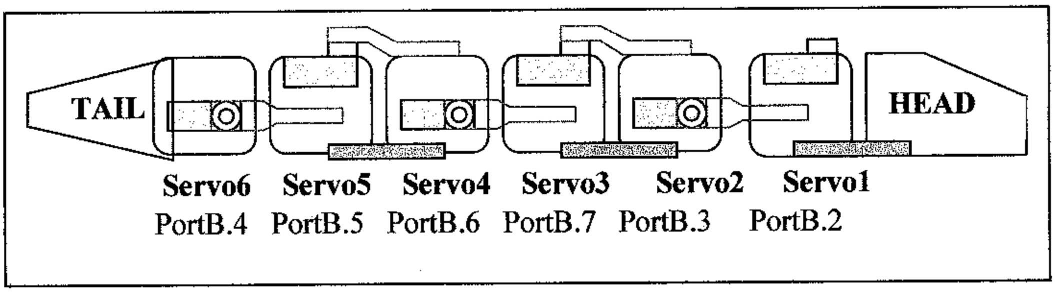


Figure 26 : Microcontroller port addresses for each of the body segment servos

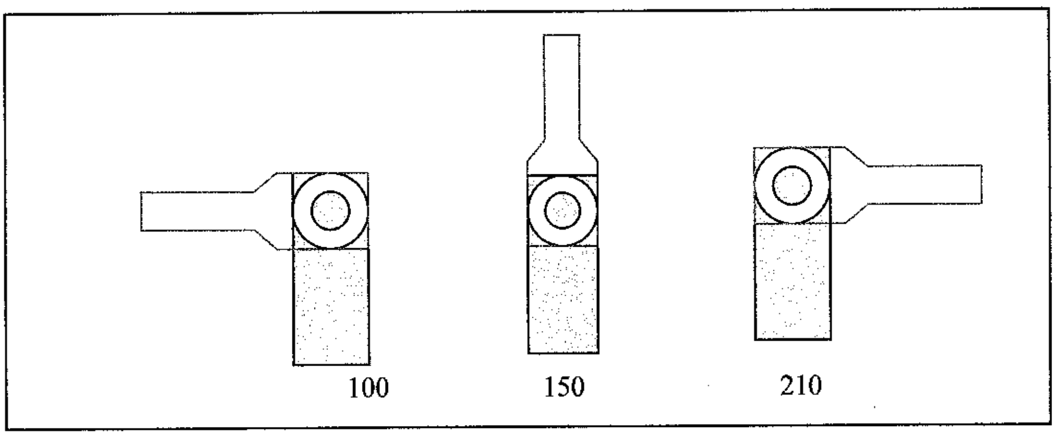


Figure 27 : Pulsout values and the corresponding servo position

To sequence the forward movement of the snake, using the servo position values shown in table can generate a sine wave pattern. The servos that move the horizontal body segments also move in a slight side-to-side movement to aid in locomotion. **Figure 28** shows the sequence that the snake's body goes through when moving in a forward direction. Frame 1 shows the snake resting before the sequence begins. Frame number 2 shows the body segment positions that correspond to the first set of positions in **Table 2**. Frame number 3 shows that the snake's body moves through the original position on its way to the second set of positions in **Table 2**. Frame number 4 shows the body segments positions that correspond to the second set of positions in **Table 2**. When the sequence is running, the body moves in a sine wave pattern. For the snake to continue moving forward, this entire sequence repeats. In the control program, the servo positions only need to be set twice, and then sequence repeats.

Table 1 : Servo Position Values to Sequence Forward Movement of the Snake in Figure 28

	Body Position 1	
Servo Number	**Port Number**	**Pulse Out Values**
1	B.2	157
2	B.3	210
3	B.7	143
4	B.6	100
5	B.5	157
6	B.4	210
	Body Position 2	
Servo Number	**Port Number**	**Pulse Out Values**
1	B.2	143
2	B.3	100
3	B.7	157
4	B.6	210
5	B.5	143
6	B.4	100

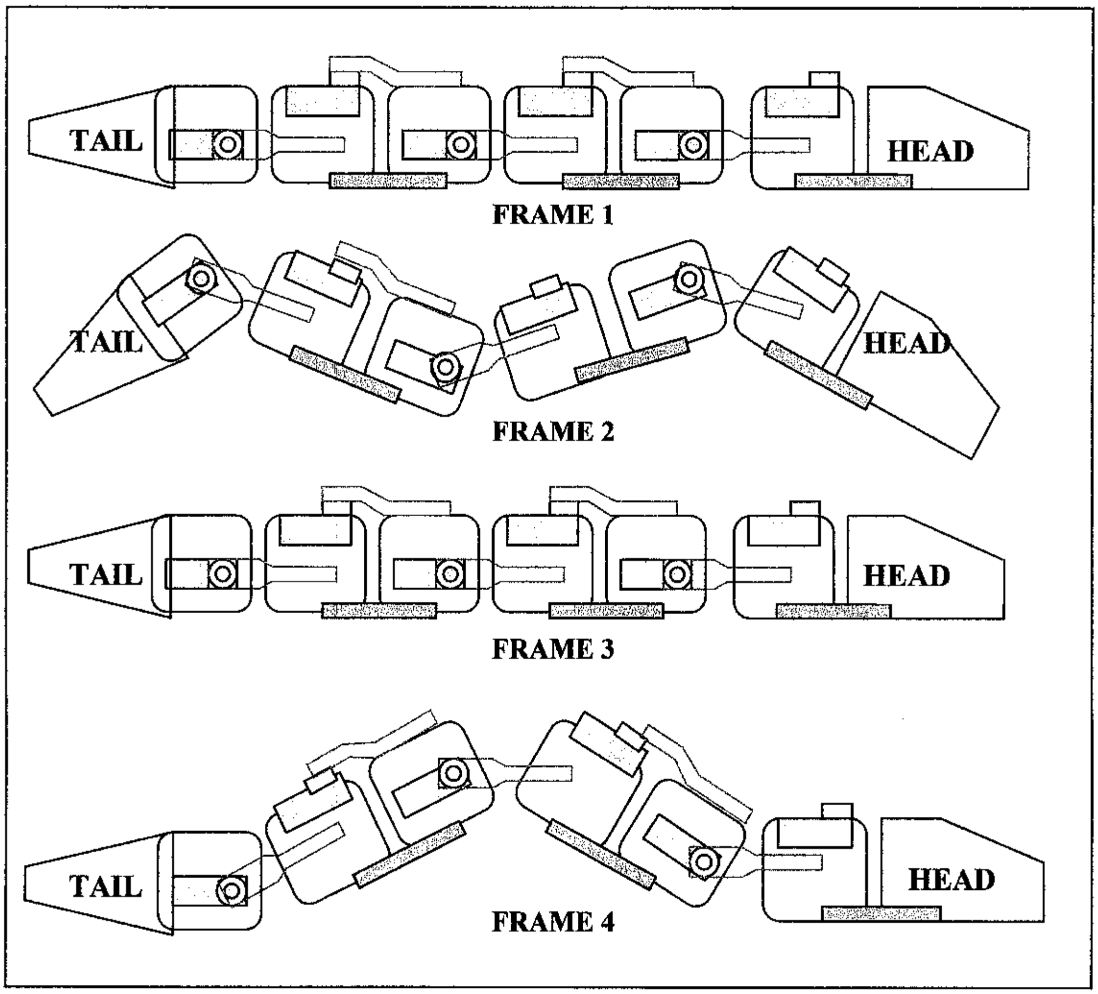


Figure 28 : Sequence of snake's body in a forward direction

4.1.4 PIC Programming

Microchip technology has developed a line of reduced instruction set computer (RISC) microprocessors called the programmable interface controller (PIC) [6]. As stated earlier, PIC16F84A will be used as the microcontroller for the design control. Please refer to **Appendix D PIC16F84A** datasheet. This device can be reprogrammed over and over again as it uses flash read only memory for program storage [7]. Thus, this makes it ideal for experimenting, as the chip does not need to be erased using ultraviolet light source. PIC16F84A is an 18-pin device 14-bit wide instruction [7]. The device is clocked using 4Mhz oscillator. The PIC is equipped with two different ports. Port A has 4 I/O lines and Port B has 7 I/O lines. Programming is done using

Basic language. Basic language is a high –level language used to provide amazingly greater amounts of functionality and speed. The BASIC language is much easier to read and write than Microchip assembly language.

4.1.5 Intelligence

The next section outlines conditioning the input received by the infrared sensor. The motion control algorithm and sensor input routines would then be put together into one main control program.

The infrared software routine will need to take input from the infrared sensor so that the robot can change its behavior to safely avoid any obstacles it may encounter while moving through its environment. A software subroutine will be developed to monitor the infrared sensor modules, perform signal processing to clean up any background noise or transient signals processing to make the information more useful, and then return results to the robot's main program. In this behavior-based method of artificial intelligence, the robot will continue on with the dominant behavior of exploring, and will change that course of action immediately based on sensor input.

The author want the main program to call the subroutine and have the subroutine simply return a value of either a 1 or a 0, with 0 indicating that no object was sensed and 1 indicating that an object is present. These values will be stored in the variable object_detect. When the program execution is returned back to the main program, certain decision can easily be made, based on this information.

The peripheral Bus Protocol (PBP) standardizes communications between the Brain and peripheral devices. Serial communication is used to transfer data. A couple of control signals regulate the flow of this data. Communication is initiated by an interrupt request (or ready) signal. When a peripheral device has data o transfer, it send this request signal to the Brain.

If the Brain receives an interrupt request, it enters an interrupt handling routine. When the Brain is ready to receive data, it outputs an allow signal. The interrupting peripheral device waits for the allow signal, and then it send its data. The layout of data lines and control signals are shown in **Figure 29**.

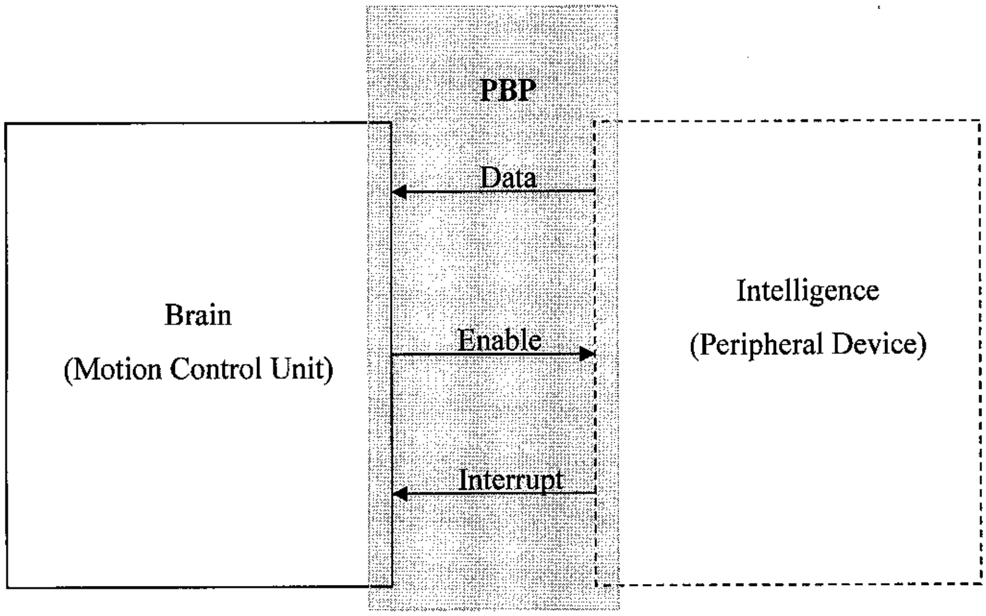


Figure 29 : PBP Layout Diagram and an organization of control signals and the data line between the Brain and peripheral device

Infrared Construction

The infrared collision module consists of one sensor. This sensor is PNA4602M IR short-range proximity sensor manufactured by Panasonic. The sensor can detect objects up to about 4 inches away. This sensor will be utilized on the head of the snake to detect objects in close proximity. This sensor has the highest priority in the author design and trigger the brain to take immediate action. When no object is present the output is high and when an object is detected, the output is asserted low. With this sensor, the snake can look ahead and plan a path to avoid problem obstacles. The Characteristics of the PNA4602M Module needed for the Infrared construction listed on the **Table 3.**

Implementing the infrared sensor module

An infrared sensor board will be fabricated to give the snake obstacle avoidance capabilities. The sensor board is comprised of an infrared LED and a Panasonic PNA4602M IR sensor module. A single-channel sensor is being used because the sensor board will be mounted at the front of the robot's moveable head. The snake is able to move its head in an arc of 180 degrees, allowing it to sense objects in front, and to either side of its body as it explores the surrounding environment. The sensor schematic is shown in **Figure 30**.

The 555 timer in the circuit is used to modulate the infrared LED at a frequency determined by C1 and R3. R3 is an adjustable 10K potentiometer that will be used to find the optimum frequency during calibration. In the application, the frequency will use between 38 and 42 kHz. So that, a meaningful signal will be sent from the PNA4602 sensor module to the microprocessor.

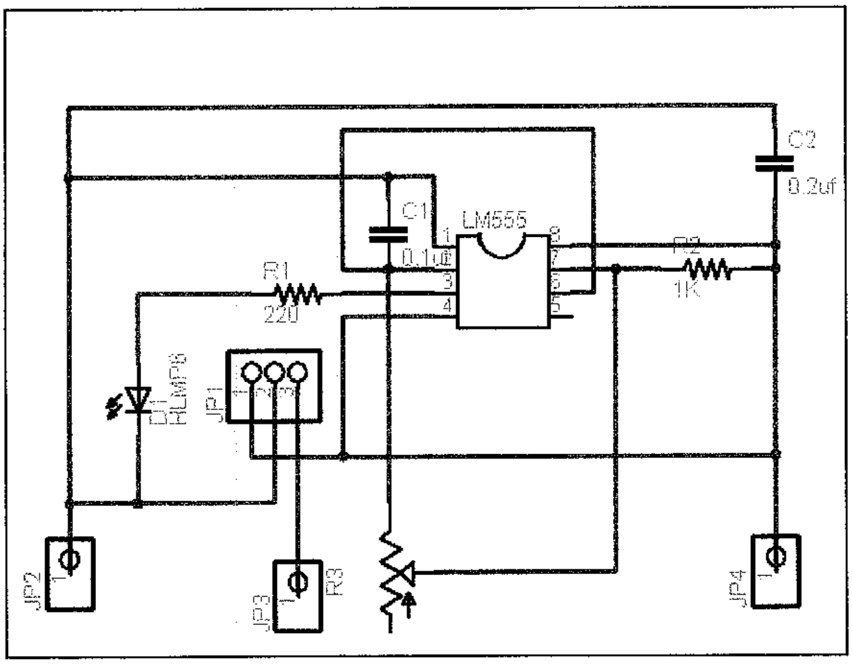


Figure 30 : Sensor Board Schematic

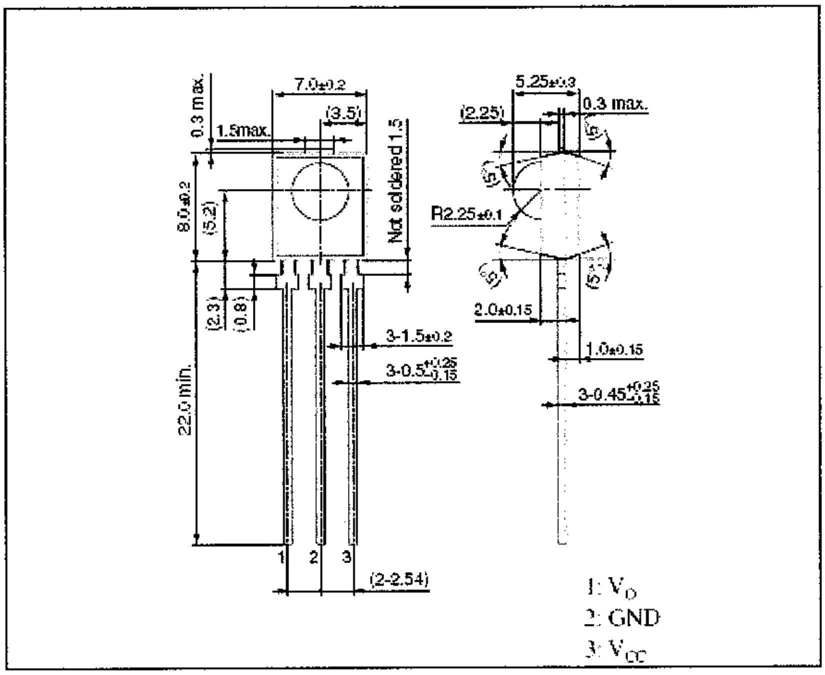


Figure 31 : Diagram of PNA4602M infrared sensor module. [7]

The PNA4602M shown in **Figure 31** is designed to detect only infrared radiation that is modulated at 38Khz, and rejects all other light sources. This makes the module an ideal sensor for daylight conditions. The features include an extension distance of 8 meters or more. No external parts are required, and a resin filter makes the module unsusceptible to visible light. **Table 3** lists the PNA4602M module's main characteristics. The output signals from the module will be processed and filtered by the microcontroller with a software routine.

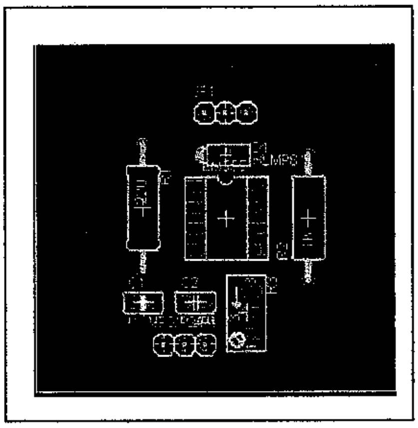


Figure 32 : Infrared sensor board PCB component side parts placement

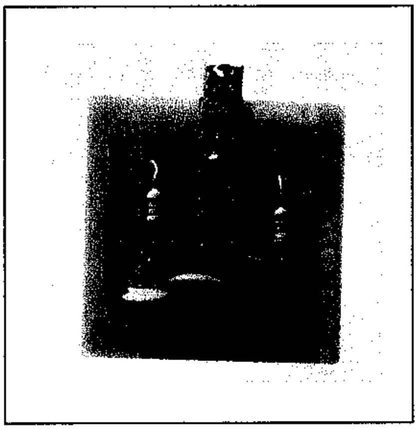


Figure 33 : Diagram of infrared sensor module. [7]

Table 2 : Characteristics of the PNA4602M Module [6]

Parameter	Symbol	Minimum	Typical	Maximum	Unit
Operating supply Voltage	V_{cc}	4.7	5.0	5.3	V
Current consumption	I_{cc}	1.8	2.4	3.0	MA
Max. reception distance	L_{max}	8	10		M
Low-level output voltage	V_{ol}		0.35	0.5	V
High-level output voltage	V_{oh}	4.8	5.0	V_{cc}	V
Low-level pulse width	T_{wl}	200	400	600	Us
High0level pulse width	T_{wh}	200	400	600	Us
Carrier frequency	F_o		38.0		kHz

4.1.6 Interfacing Video Capture Card with Wireless Camera

In order the snake robot to operate in a remote visual operation, a mini wireless camera is added on the front of the snake's head.

To install the wireless camera, it is needed to attach the Reception Antenna to the Wireless Receiver by twisting into it. Then the wireless receiver is connected to the Monitor with Audio/Video Cable. The Wireless Receiver is powered up with a DC 9V/12V Power Adapter. For the wireless camera, a DC 8V Power Adapter is connected to Power Supply Socket of Wireless Camera. Note that incorrectly connecting the Wireless Camera to the 9V/12V Power Adapter may cause permanent camera damage. The lens-protecting cap from the camera lens is removed from the camera and everything should be connected nicely. To obtain the best picture quality, the Adjust Frequency Controller is adjusted. Wireless Camera lens can be adjusted to a position that picture is in focus. The Wireless Camera can now be attached with the servomotors to obtain pan and tilt movements.

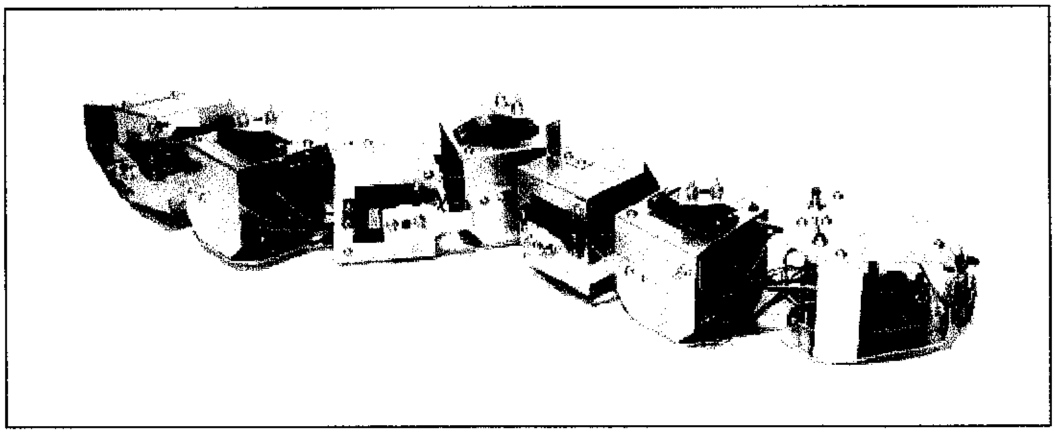


Figure 34 : The completed Snake Robot construction.

4.2 Discussion

The 3-volt battery packs, located in each body segment will be wired to provide 6 volts to the controller board. The 6-volt supply will be used to directly power the servos. To accomplish this, the first two battery packs will be wired in as series to create 6 volts. The next pair of battery packs are also wired in a series to create 6 volts, as are the last two. Each of these three pairs are then wired in parallel so that the supply is 6 volts, but capable of providing higher current and a longer robot operating time. This is important since the robot will be coordinating the movement of six servos that may all be in operation at the same time. The 9-volt supply is from a single battery mounted in the first body segment. This supply is used to power the controller board. The use of dual power supplies with a robot is preferred because it provides the microprocessor with isolation from the noise introduced by the direct current motors in the servos. It also allows the robot to run for a much longer time because the microcontroller can keep operating from the 9-volt supply, even if the 6-volts supply drops down to 4 volts. The servos are capable of operating at lower voltages, but if the PIC's supply drops below 5 volts, it will go into a resetting loop. By powering the microcontroller with its own 9-volt source, this problem is eliminated

Refer to **Figure 25** when wiring each of the 3-volt battery packs and the 9-volt battery to the DPDT switch and the controller board. Start by mounting the DPDT switch in one of the ¼-inch mounting holes on the top of the snake's head. Wire each of the 3-volt battery packs in the body sections with the battery clips that attach to each holder. It may be easiest to connect each of the battery clips together before attaching them to the battery packs. The connections are soldered in place and that insulating heat-shrink tubing is placed around each connection. All of the wires should run inside the snake from one section to another. The connector wire is used to attach the switch to the power terminal blocks on the controller board. A 9-volt battery in the battery holder that is located in the first body section behind the head. The negative lead of the 9-volt battery clip is attached to the 9-volt power terminal connector on the controller board, and solders the positive leas to the switch.

The sensor board works by producing modulated infrared radiation with an infrared LED and using the PNA4602 module to detect any radiation reflected from the surface of solid objects. The PNA4602 sensor is designed to respond only to infrared that is modulated at a frequency somewhere between 38-42kHz. The circuit is tuned to modulate the infrared LED at this frequency. (Referring to the code at Appendix C) Depending on the proximity of the sensor to the object, a greater `pr` least number of infrared pulses will be reflected back. The number of reflected "hits" that the sensor receives in a given time frame allows the robot to determine how close it to objects. The higher the number reflected pulses, the closer the sensor to the object. The output pin from the PNA4602 is connected to a microcontroller input pin, and a software routine is used to monitor the sensor.

The infrared subroutine takes 40 samples from the module and counts the number of positive hits received. (Referring to the code at Appendix C) Changing the variable `num_samples` can also configure the number of samples taken. Because of stray infrared and signals from the environment, the module is constantly producing false positive signals that are referred to as "noise". The average acceptable amount of noise picked up by the sensor module is called the noise floor. The routine need to set a threshold point above the typical amount of noise and report a detected object only if the number of positive signals received throughout the number of samples taken exceeds the noise floor.

With the PNA4602M sensor modules, the author found that the typical false positve was actually very low: five for every 40 samples taken. On the safe side, the threshold is set at 25 for every 40 samples, to ensure that an object is present. By changing the threshold value, we can change the sensitivity and distance detection response of the module. (Referring to the code at Appendix C) If we want a more accurate reading, the `num_samples` value can be increased, but will take more time for the routine to execute.

The last option is using the mode select push button to invoke the infrared sensor calibration routine. This will enable the user to simply push the button on the robot's head to calibrate the sensor. The experimenter can also develop a software routine to use the push button to choose different modes of behavior when the robot starts up.

When the main software routine senses that the button has been pushed, it goes into tight loop until it senses that the switch has been let up before going to the infrared calibration mode. This is so that when the program execution jumps to the calibration routine, it does not immediately jump back to the main routine because the operator still has the button pushed.

4.3 Problem Encountered

Naturally, it is almost inevitable to avoid problems and complications when it comes to developing a circuit from scratch. It can be said that the time spent on designing is almost equal to the time spent for troubleshooting. There are several occasions where implementation has suffered some problems even though the design theory is successful. The main area affected by this consequence is the integration of all various components to establish the servomotor. Besides, the power sources distribution between the microcontroller and servo motor.

CHAPTER 5
CONCLUSION & RECOMMENDATIONS

Specifically for this project, designing Snake Like Robot, it involves all aspects from mechanical design up to programming. Snake Robot project requires some fundamental of mechanical design, applying practical usage of motors and sensors, electronic circuit design as well as microcontroller programming. This project not only provides the opportunity to apply the knowledge practically but also enhance the problem solving and designing skills.

The most important part of this development of sine wave pattern is the mechanical design. The structure has to be light in weight, rigid, and also simple and in great balance. This includes critical design in choosing the right material, and using the correct balancing concept. The material chosen is aluminum as it is light in weight, strong and rigid as well as available in nearby hardware store. All sawing and drilling is in the workshop in Building 22,UTP.

Snake like robot technology is implemented in designing the algorithm for locomotion to achieve a stable and simple sine wave pattern. The project was to design and build a robot snake that could move much its biological counterpart. This robot has the ability to explore autonomously, avoid obstacles using an infrared sensor and can also be guided by radio control. A robot snake is ideal for traversing rough terrain and keeping a low profile. This robot has many applications such as exploration, reconnaissance missions, remote sensing in hostile areas, land mine detection, and maintenance work inside narrow passageways. The objective of the project is fully achieved where the finished product is cost effective, light, robust, and intelligently applicable.

One of the major factors in difficulties during the whole project is constructing each of the features available and putting them all together into one piece. A lot of knowledge such as Circuit Theory, Analog, and Communication System are required to complete this project. On top of that, programming skills in BASIC is essential as well.

This concludes the construction and programming of the robot snake. Much more can be done with this robot than what has been covered. A remote control cab easily is added to this project, since there are two connectors on the controller board for this purpose. There are several recommendations based on the completion of the project. Use of the infrared sensor and the snake's head movement to scan the area around the snake for objects. Use this information to determine the correct path before moving. In order to look more likely snake this can create a skin for the robot using a waterproof material such as latex rubber. The snake also can be developed a side-winding movement routine. More than, figure out a routine that will enable the robot to move in reverse, unlike a real snake. Besides, add a tilt sensor so that the robot will know when it has tipped over, and can then right itself. Last but not least to write a routine enabling the snake to roll over.

REFERENCES

[1] Asrul Nizar Bin Ahmad, FINAL YEAR PROJECT REPORT, " Remote Control For PC", 2005.

[2] "Anonymous, Snake anatomy," 2003, snake anatomy picture. [Online]. Available: http://www.reptileallsorts.com/anatomy.htm

[3] K. Bonsor, "How snakebots work," 2003. [Online]. Available:

http://www.howstu_works.com/snakebot.htm/printable

[4] H. Choset, "Modular serpentine robot locomotion," 2003. [Online]. Available:

<http://voronoi.sbp.ri.cmu.edu/projects/modsnake/modsnake.html>

[5] G. Haith," Snakebot," October 2000. [Online]. Available:

http://amesnews.arc.nasa.gov/releases/2000/00nprotectnglobalnletnOT1ntextunderscorenunhbox

[6] J.Iovive, "PIC ROBOTICS," Tab Robotics, Mc Graw Hill, 2003, page 4.

[7] Richard Barnett, Larry O'Cull and Sarah Cox, Embedded C Programming and the Microchip PIC, Delmar Learning, 2004.

[8] http://www-robot.mes.titech.as.jp/robot/snake/acmr3/acm_r3.html, About ACM-R3 Snake Like Robot, page 1.

[9] G.Haith. "Snake Robot," October 2000. Available: http://amesnews.arc.nasa.gov/releases/2000

[10] S.Hirose, Biologically Inspired Robots: Snake-Like Locomotors and Manipulators. Oxford University Press, October 1996.

[11] B.Oraw & J.Tinder, Modular Snake Robot, http://www.cs.plu.edu/~snake/, December 13, 2003.

[12] S. Hirose, Biologically Inspired Robots: Snake-Like Locomotors and Manipulators. Oxford University Press, October 1996.

[13] G. Chirikjian, "Hyper-redundant robot mechanism and their applications," November 1991, pp. 185{190.

[14] J. Ostrowski and J. Burdick, “Geometric perspectives on the mechanics and control of robotic locomotion,” October 1995, pp. 487-504.

[15] H. Choset and J. Burdick, “Sensor based planning and nonsmooth analysis.”

[16] K. L. Paap, M. Dehlwisch, and B. Klaasen, \Gmd-snake: A semi-autonomous snake-like robot," August 1997. [Online]. Available: http://set.gmd.de/RS/snake/

[17] M. Yim. Polybot. [Online]. Available: http://www2.parc.com/spl/projects/modrobots/chain/polybot/

[18] G. Haith, \Snakebot," October 2000. [Online]. Available: http://amesnews.arc.nasa.gov/releases/2000/00nprotectnglobalnletnOT1ntextunderscorenunhboxn

[19] G. S. Miller, Snake Robots for Search and Rescue. Massachusetts Institute of Techonology, 2002.

[20] Oh Lay Shan, FINAL YEAR PROJECT REPORT, “ Design and Implementation of An Intelligent Hexapod”, 2005.

[21] Home of the EAGLE Layout Editor Available : http://www.cadsoft.de/

[22] MicroCode Studio 2 Available : http://www.brothersoft.com/Software_Developer_Compilers_and_IDEs_MicroCode_Studio_539.html

[23] WARP 13 Available: http://www.newfoundelectronics.com/

[24] Modular Snake Robot Available: http://www.cs.plu.edu/~snake/projects/capstone/papers

APPENDICES

APPENDIX A

CONTINUOUS ROTATION SERVO

PARALLAX
599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office: (916) 624-8333
Fax: (916) 624-8003
General: info@parallax.com
Technical: support@parallax.com
Web Site: www.parallax.com
Educational: http://www.parallax.com/html_pages/edu/

Continuous Rotation Servo (#900-00008)

General Information

The Parallax Continuous Rotation servo is ideal for robotic products that need a geared wheel drive or other projects that require a 360 degree rotation geared motor. The Parallax Continuous Rotation servo output gear shaft is a standard Futaba configuration. The servo can be adjusted with a small Phillips screw driver if the unit becomes out adjustment on its center set point. Servo is custom manufactured for Parallax by Futaba.

Technical Specifications

> Power 5vdc max

> Average Speed 60 rpm
Note: with 5vdc and no torque

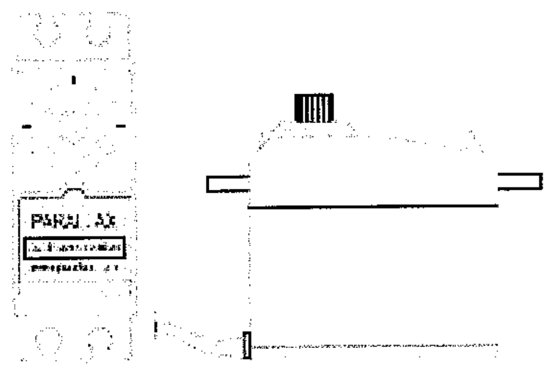
> Weight 45.0 grams/1.59oz

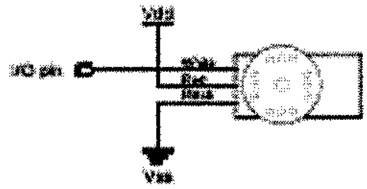
> Torque 3.40 kg-cm/47oz-in

> Size mm (L x W x H)
40.5x20.0x38.0

> Size in (L x W x H)
1.60x.79x1.50

> Manual adjustment port





The servo is controled by pulsing of its signal line. If you are using a Basic Stamp® microcontroller this is done with the 'pulsout' command. Below is PBASIC code that will help you with basic control of a servo. The code below will show center and then rotate the servo to the left and to the right then stop.

Basic Stamp 1 code

```
SYMBOL Temp = W0                'Work space for FOR NEXT
SYMBOL Servo pin  = 0           'I/O pin that is connected to servo

  FOR temp = 0 TO 200  '
  PULSOUT Servo pin,150
  PAUSE 20
  NEXT

  FOR temp = 0 TO 200
    PULSOUT Servo_pin,180
    PAUSE 20
  NEXT

  FOR temp = 0 TO 200
    PULSOUT Servo_pin,120
    PAUSE 20
  NEXT
Stop
```

BASIC Stamp 2, 2e,2pe code

```
Temp          VAR          Word    'Work space for FOR NEXT
Servo pin     CON          0       'I/O pin that is connected to servo

  FOR temp = 0 TO 200
    PULSOUT servo pin,750
    PAUSE 20
  NEXT

  FOR temp = 0 TO 200
    PULSOUT servo_pin,900
    PAUSE 20
  NEXT

  FOR temp = 0 TO 200
    PULSOUT servo_pin,600
    PAUSE 20
  NEXT
STOP
```

BASIC Stamp 2sx,2p24/40 code

```
Temp          VAR          Word    'Work space for FOR NEXT
Servo pin     CON          0       'I/O pin that is connected to servo

  for temp = 0 to 200
   pulsout Servo_pin,1875
   pause 20
  next

  for temp = 0 to 200
    pulsout Servo_pin,2350
    pause 20
  next

  for temp = 0 to 200
    pulsout Servo_pin,1500
    pause 20
  next
STOP
```

APPENDIX B

PIC 16F84A

MICROCHIP

PIC16F84A

18-pin *Enhanced* FLASH/EEPROM 8-Bit Microcontroller

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt-on-change
  - Data EEPROM write complete

Peripheral Features:

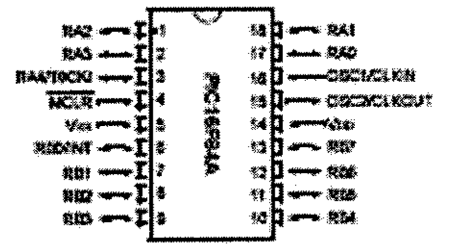
- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Special Microcontroller Features:

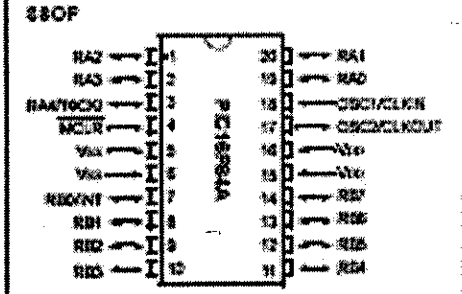
- 10,000 erase/write cycles Enhanced FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode

Pin Diagrams

PDIP, SOIC



SSOP



CMOS Enhanced FLASH/EEPROM Technology:

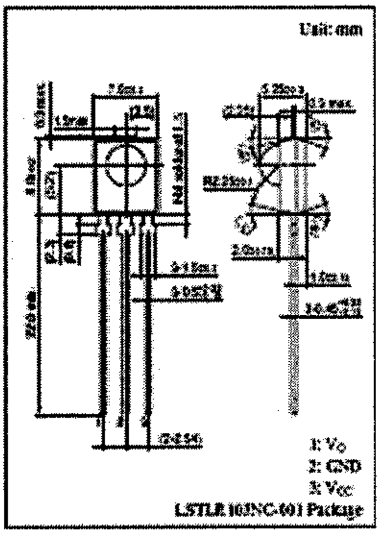
- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 5.5V
  - Industrial: 2.0V to 5.5V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 2V, 32 kHz
  - < 0.5 µA typical standby current @ 2V

APPENDIX C

PNA4602M

Photodiode with amplifier functions

For infrared remote control systems



■ Features

- Extension distance is 8 m or more
- External parts not required
- Adoption of visible light cutoff resin

■ Absolute Maximum Ratings $T_a = 25°C$

Parameter	Symbol	Rating	Unit
Collector supply voltage	V_{CC}	− 0.5 to +7	V
Power dissipation	P_D	200	mW
Operating ambient temperature	T_{opr}	− 20 to +75	°C
Storage temperature	T_{stg}	−40 to +100	°C

■ Electrical-Optical Characteristics $T_a = 25°C \pm 3°C$, $V_{CC} = 5.0$ V

Parameter		Symbol	Conditions	Min	Typ	Max	Unit
Collector supply voltage		V_{CC}		4.7	5.0	5.3	V
Supply current		I_{CC}	No signal condition	1.8	2.4	3.0	mA
Load resistance		R_L		15	20	25	kΩ
Maximum reception distance *1		l_{max}		8.0	10.0		m
Output voltage low-level *2		V_{OL}	L ≤ 8.0 m, I_{OL} = 400 μA		0.35	0.50	V
Output voltage high-level		V_{OH}	No signal condition, I_{OH} = −10 μA	4.75	4.80		V
Pulse width low-level *1	PNA4601M	T_{WL1}	L = 8.0 m, 16 pulse	200	400	600	μs
		T_{WL2}	L = 0.2 m, 16 pulse, T_a = 65°C ± 3°C	100		700	
	PNA4602M	T_{WL}	L = 0.1 m to 8.0 m, 16 pulse	200	400	600	
Pulse width high-level *1	PNA4601M	T_{WH}	L = 8.0 m, 16 pulse	200	400	600	μs
	PNA4602M		L = 0.1 m to 8.0 m, 16 pulse	200	400	600	
Center frequency	PNA4601M	f_0			36.7		kHz
	PNA4602M				38.0		

Note) 1. Measuring methods are based on JAPANESE INDUSTRIAL STANDARD JIS C 7031 measuring methods for diodes.
2. *1: Burst wave form Figure 1
*2: Constant wave form Figure 2

Carrier frequency: f_0

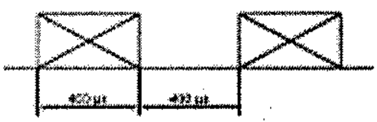


Figure 1

Carrier frequency: f_0

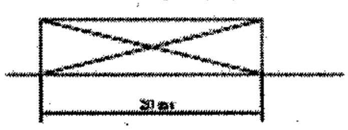
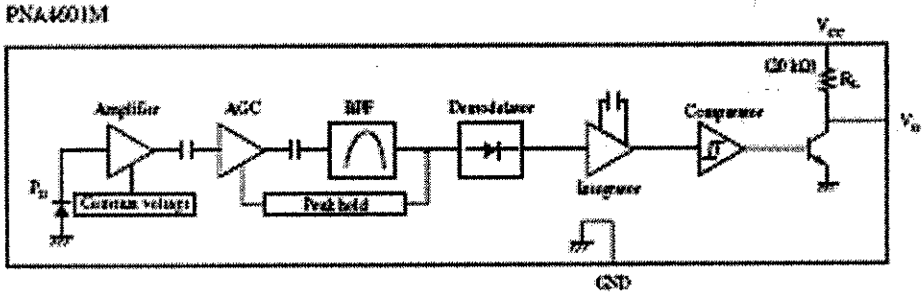


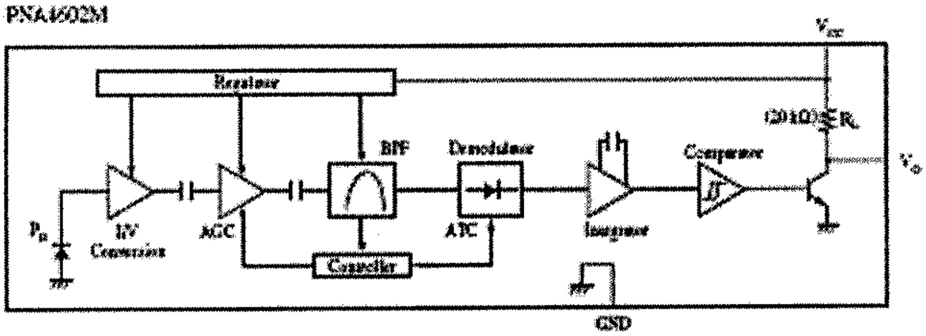
Figure 2

Block Diagram

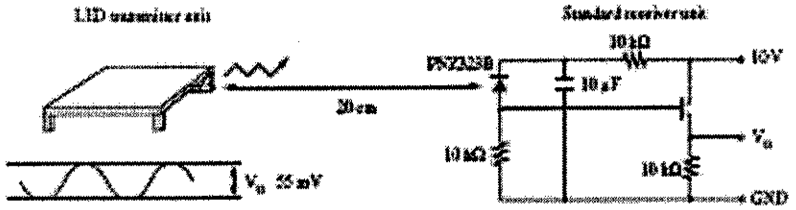
PNA4601M



PNA4602M



Panasonic Transmitter Specifications



1. The output of the LED transmitter unit is adjusted so that the output standard receiver unit, V_O may be 55 mV when transmitting waves (duty = 50%) are output from the transmitter unit, where the sensitivity to infrared emitters (S_{IR}) of PNZ323B is 0.53 μA when the irradiance H is 12.45 μW/cm^2.

2. The maximum detection distance of this specification is guaranteed by T_{WH} and T_{WL} being within the limits when constant 16 pulses are transmitted with the output of the transmitter unit corresponded to the maximum detection distance in the system above. (The maximum detection distance is measured in the darkness without disturbing noises.)

APPENDIX D

'MAIN.BAS' SOURCE CODES

```
' Author    : Mohd Hafiz B Roslan
' Compiler : MicroCode Studio
' Notes     : Complete control Program for the robot
'           : snake. Mode select push-button switch
'           : allows the infrared sensor to be easily
'           : calibrated. The robot will stop and turn
'           : if an obstacle is encountered.


' PortA set as outputs
TRISA = %00000000

' PortB set as outputs. pins 0-1 inputs
TRISB = %00000011

'-----------------------------------------------------
' INITIALIZE VARIABLES


cal_switch var PORTB.0
ir_input var PORTB.1
servo_1 var PORTB.2
servo_2 var PORTB.3
servo_3 var PORTB.7
servo_4 var PORTB.6
servo_5 var PORTB.5
servo_6 var PORTB.4
```

```
ir_count var Byte
temp var Byte
object_detect var Byte
num_samples var Byte
threshold var Byte
rand var Word
timer var Byte
temp1 var Byte
i var Byte

look_right var Byte
look_left var Byte
turn_count var Byte

servo1 var Byte
servo2 var Byte
servo3 var Byte
servo4 var Byte
servo5 var Byte
servo6 var Byte

Low led_left
Low led_right

Low servo1
Low servo2
Low servo3
Low servo4
Low servo5
Low servo6

turn_count = 0
num_samples = 40
threshold = 25
```

```
'-------------------------------------------------
' CREATE RANDON NOISES AND FLASH LED'S

For temp1 = 1 To 5
High led_left
Low led_right
Gosub randomize
pause 50

Low led_left
High led_right
Gosub randomize
pause 50
Next temp1

Low led_right

'-------------------------------------------------
' START MAIN EXECUTION

start:

If cal_switch = 1 Then
pause 50
release_calibrate:
If cal_switch = 1 Then
Goto release_calibrate
Else
sound piezo, [120, 4, 90, 2, 100, 2, 110, 4]
pause 50
Goto ir_cal
Endif
Endif
```

```
Gosub infrared

If object_detect = 1 Then
High led_left
High led_right
sound piezo, [100, 4, 90, 2]
servo1 = 180
Gosub servo
servo1 = 120
Gosub servo
turn_count = turn_count + 1
If turn_count.0 = 1 Then
Gosub slide_right
Else
Gosub slide_left
Endif

Endif

Low led_left
Low led_right

Gosub forward

Goto start
```

APPENDIX E

SINE WAVE PATTERN MOTION BASIC CODES

```
'SUBROUTINES START HERE

'-------------------------------------------------
' SLITHER FORWARD ROUTINE IN A SINE WAVE PATTERN

forward:
servo1 = 157
servo2 = 210
servo3 = 143
servo4 = 100
servo5 = 157
servo6 = 210
Gosub servo
servo1 = 143
servo2 = 100
servo3 = 157
servo4 = 210
servo5 = 143
servo6 = 100
Gosub servo
Return

'-------------------------------------------------
' RIGHT TURN MOVEMENT ROUTINE

slide_right:

For temp1 = 1 To 3
servo1 = 150
servo2 = 210
```

```
servo3 = 150
servo4 = 100
servo5 = 150
servo6 = 210
Gosub servo
servo1 = 190
servo2 = 100
servo3 = 190
servo4 = 210
servo5 = 190
servo6 = 100
Gosub servo
Next temp1
Return

'----------------------------------------------------
' LEFT TURN MOVEMENT ROUTINE

slide_left:

For temp1 = 1 To 3
servo1 = 150
servo2 = 210
servo3 = 150
servo4 = 100
servo5 = 150
servo6 = 210
Gosub servo
servo1 = 100
servo2 = 100
servo3 = 100
servo4 = 210
servo5 = 100
servo6 = 100
```

```
Gosub servo
Next temp1
Return

'-------------------------------------------------
' RANDOM SOUND GENERATOR SUBROUTINE

randomize:

random rand
i = rand & 31 + 64
sound piezo, [i, 4]
Return

'-------------------------------------------------
' INFRARED DETECTION SUBROUTINE

infrared:

ir_count = 0
object_detect = 0

For temp = 1 To num_samples
If ir_input = 0 Then
ir_count = ir_count + 1
Endif
Next

If ir_count >= threshold Then
object_detect = 1
Endif

Return
```

```
'-------------------------------------------------
' SUBROUTINE TO CALIBRATE I.R. SENSORS

ir_cal:

If ir_input = 0 Then
High led_left
High led_right
Endif
Low led_left
Low led_right

If cal_switch = 1 Then
pause 50
button_release:
If cal_switch = 1 Then
Goto button_release
Else
sound piezo, [120, 4, 90, 2, 100, 2, 110, 4]
pause 50
Goto start
Endif
Endif

Goto ir_cal

'-------------------------------------------------
' SUBROUTINE TO SET SERVOS

servo:

For timer = 1 To 20
pulsout servo_1, servo1
pulsout servo_2, servo2
```

```
pulsout servo_3, servo3
pulsout servo_4, servo4
pulsout servo_5, servo5
pulsout servo_6, servo6
pause 12
Next timer

Return
```

APPENDIX F

PARTS LIST FOR MECHANICAL CONSTRUCTION

PARTS	QUANTITY
1/16-inch thick zinc stock	8-foot x 10-foot piece
6/32 x ½-inch machine screws	98
6/32 locking nuts	98
6/32 nylon washers	6
Parallax Continuous Rotation Servo	6

APPENDIX G

PART LIST OF MAIN CONTROLLER BOARD

PART	QUANTITY	DESCRIPTION
Semiconductor		
U1	1	78L05 5V Regulator
U2	1	PIC 16F84 flash microcontroller mounted in socket
D1	1	Red light-emitting diode
D2	1	Green light-emitting diode
Resistors		
R1	1	4.7kΩ ¼-watt resistor
R2, R3, R4	3	1kΩ 1/4 –watt resistor
Capacitors		
C1	1	0.1uF
C2,C3	2	22pf
Misc		
JP1-JP8	8	3 post header connector 2.5mm spacing
Jp9, JP10	2	1 post header connector 2.5mm spacing
5 Volt Power	3	2 post header connector 2.5mm spacing
Y1	1	4-MHZ crystal
Piezo Buzzer	1	Standard Piezoelectric element
BT1 and BT2	1	4-contact terminal block
I.C socket	1	18-pin I.C socket soldered to PC board U2

APPENDIX H

MILESTONE OF FIAL YEAR PROJECT 1

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic	■													
	-Propose Topic														
	-Topic assigned to students														
2	Preliminary Research Work		■	■											
3	Submission of Preliminary Report		●												
4	Project Work				■	■	■	■							
	-Reference/Literature														
	-Practical/Laboratory Work														
5	Submission of Progress Report						●								
6	Project work continue								■	■	■	■			
	-Practical/Laboratory Work														
7	Submission of Interim Report Final Draft										●				
8	Oral Presentation											●			
9	Submission of Interim Report												●		

APPENDIX I

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Project Work Continue		■	■											
	-Practical/Laboratory Work														
2	Submission of Progress Report 1	●													
3	Project Work Continue				■	■	■	■							
	-Practical/Laboratory Work														
4	Submission of Progress Report 2						●								
5	Project work continue								■	■	■	■			
	-Practical/Laboratory Work														
6	Submission of Dissertation Final Draft											●			
7	Oral Presentation												●		
8	Submission of Project Dissertation													●	

MILESTONE OF FIAL YEAR PROJECT 2