**Implementing, Testing and Debugging a Mini-Hexapod**

Bajwa Sandeep, Carnley James, Dani Jay, Johnson Aaron, Kachhadia Bhumika,

Patel Dhairya, Patel Maulikkumar, Patel Pratikbhai, Rahman Md.,

Raikar Poonam, Sands Dean, Srirangam Narasimha, Venkata Susheel

Coordinator: Dr. Stefan Andrei

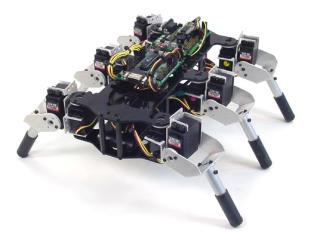
Lamar University

Department of Computer Science

Beaumont, Texas

**Abstract:** The synchronization between hardware and software components can be illustrated using an excellent technique of solving a programming logic problem with a programmable electronic device. This can be achieved using an embedded system which is designed to perform one or few dedicated functions often with real-time computing constraints. In this paper, we implement and examine the development process of an embedded system, namely the Mini-Hexapod. Utilizing the innovative solutions to common robotics problems provided by the Lynxmotion, we physically simulate the problem by using a powerful Bot Board/SSC-32/BA28 and PS2 Controller attached to a Mini-Hexapod to carry out the actual motions to solve the problem. Our team experienced certain challenges during the implementation phase of the robot to maintain precise and graceful motions, which we had overcome by allowing synchronization between legs of the Mini-Hexapod as an actual spider and not as a human being. The actual movement of the mini-hexapod is a bit slow as compared to the other robots such as Mini-Rover; however Mini-Hexapod can efficiently cope up with this drawback upon solving a much more challenging “Steps Climbing” problem.

1. **Introduction**

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. Embedded systems are controlled by one or more main processing cores that are typically either a microcontroller or a digital signal processor (DSP). The key characteristic is however being dedicated to handle a particular task, which may require very powerful processors. For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites.

The Mini hexapod measures 10.5" long and 12" wide and supports the Bot Board and SSC-32. It has two DOF (degrees of freedom) leg design along with a mounting hardware to hold a Bot Board and SSC-32. The chassis is also available in black Lexan. It includes 6 ball bearings for leg attachment which makes it easier to move on uneven surfaces. It also has the capability of climbing stairs. In this project, we have implemented the Mini-Hexapod as an autonomous robot.

The Mini-Hexapod can be compared with a Mini-Rover in many aspects. Hexapod is a robot with two actuated degrees of freedom which enables it to have smooth motion on flat surfaces. Even though the speed of rover is more as compared to hexapod, hexapod has the capability of climbing stairs and traversing obstacles, and adaptability toward uphill, downhill and slope surfaces. It can also move on uneven surfaces. However, rover doesn’t have the above mentioned capabilities to a large extent.

Lynxmotion was founded in 1995 due to a fascination with robotics and a frustration with the available robot-based kits. They have a nice selection of intelligently designed, precision engineered robot kits and components. They have many innovative solutions to common robotics problems. In addition to lots of products, they also have lots of information on doing hobby robotics. Tech support is also provided by them. They also publish the assembly guides on their website. Also their two robots namely Johnny 5 and AH3-R robots were used in the lab of the new Knight Rider movie.

**2. Related Work**



The Phoenix Hexapod Robot has a more natural looking, more articulate leg and body design. It has three DOF (degree of freedom) leg design which means that this robot can walk in any direction. The robot has been designed to use 18 Hitec HS-645 servos for the legs. It uses the Bot Board/SSC-32/BAP28 which is then controlled by PS2 robot controller.

The AH3-R robot's round body symmetry makes this a very unique hexapod walker. The three DOF (degree of freedom) leg design provides the flexibility required to walk in any direction. The robot uses 18 HS-645 servos for the legs. The combo kit contains the SSC-32, Bot Board, and BASIC Atom Pro 28. The current program allows the robot to walk with variable speed, in any direction (translation), or turn in place (rotation), or any combination of the two. The leg lift and ride height is adjustable as well as the gait walking speed. There are preset walking modes for tall grass, tile floor, low profile and normal operation. The body can be rotated in every axis. There are also preprogrammed poses and a sleep mode. All of these are accessible from the controller.

The BH3 robot offers the most advanced leg design available today. The three DOF (degree of freedom) leg design make this robot to walk in any direction. The robot uses 18 Hitec HS-475 / HS-485 servos for the legs. The robot is made from ultra-tough laser-cut Lexan structural components, custom injection molded components, and high-quality aluminum Servo Erector Set brackets.

**3. Background**

Similar to all the other embedded systems, the Mini-Hexapod has two basic components: Hardware and Software components. Hardware component consists of SSC-32/BA28 controller and servos which are assembled together to perform the moves specified by the software component. Software component is programmable and is responsible for controlling the movement of the Mini-Hexapod.

**3.1. Software Specification**

BasicMicro Atom IDE

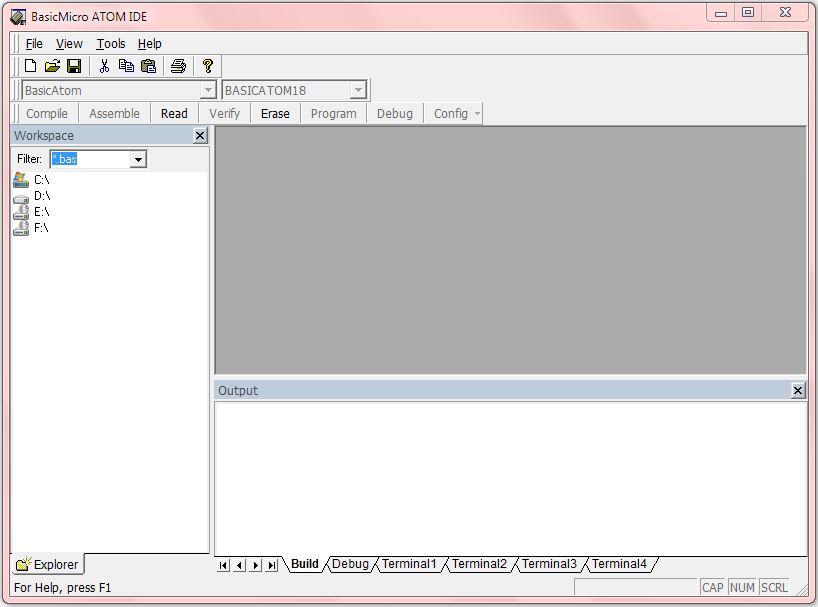


Figure 3.1.1 BasicMicro Atom IDE [v5.3.1.0](http://www.basicmicro.com/downloads/software/BasicATOMSetup_5310.exe)

BasicMicro-IDE is a Windows based Integrated Development Environment (IDE) for micro-controllers. Originally designed for use with Micro-C, Micro-IDE is fully user-configurable to convert command line compilers, assemblers and utilities into Windows applications. It is a Multi-file Editor with tabs to create and modify C or Assembly source code. It has a Built-in Loader to download programs to target microcontroller board following build. It also includes Built-in Terminal to interact with the target microcontroller board. In addition, it has many features such as customizable settings and command-line options for building projects, ASCII chart and so on. The BasicMicro IDE supports many external toolkits such as C compilers, BASIC compiler and assemblers.

Lynx SSC-32 Terminal

The Lynx SSC-32 Terminal is a software component that makes it easy to quickly test all functionality of the SSC-32 Servo Controllers of the Mini-Hexapod.

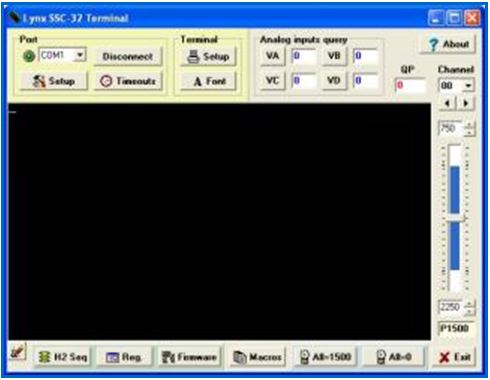


Figure 3.1.2 Lynx SSC-32 Terminal v1.10

The Lynx Terminal software also consists of hexapod sequencer which is used to configure the servos of the Mini-Hexapod.

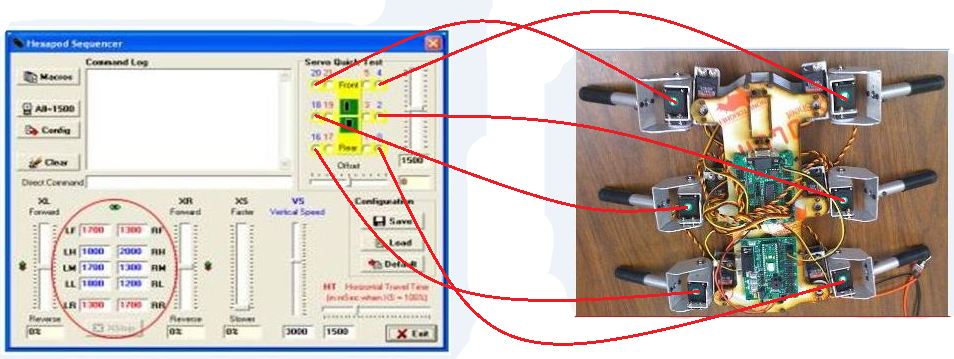


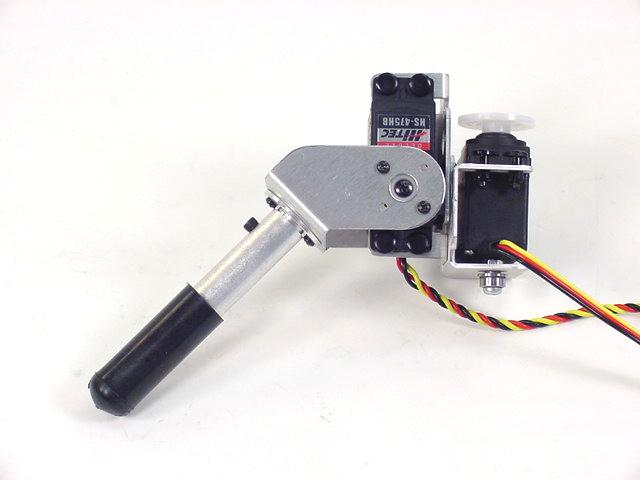
Figure 3.1.3 Configuring the Mini-Hexapod using the Lynx Sequencer software

**3.2. Hardware Specification**

The Mini-Hexapod includes a powerful Board/SSC-32/BA28 and a PS2 controller.

Mini-Hexapod Body Kit:

The Body kit of mini-hexapod contains laser-cut Lexan panels and aluminum standoffs make a nice Hexapod robot chassis for use with the SES legs. It has mounting hardware to hold a Bot Board and SSC-32 and includes 6 ball bearings for leg attachment.



The 2DOF (degree of freedom) leg design:

This type of a leg design is popular with Mini Hexapod. The all aluminum construction even incorporates ball bearings in each joint. These 2DOF legs are made from the highest quality of aluminum alloy with a brushed finishing.



The HS-422 (57 oz. in.) Standard Servo:

Size = Standard

Servo Erector Set = [ASB-xx Standard Scale line](http://www.lynxmotion.com/Category.aspx?CategoryID=87)

Range = 180°

Voltage = 4.8 - 6.0vdc

Torque = 57 oz.-in.

Weight = 1.66 oz.

Speed = 0.16s / 60 degrees

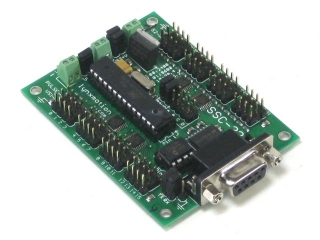
The Bot Board II:

The Bot Board II is the best carrier board for the Basic Atom, or any other 24 or 28 pin microcontrollers. It has an onboard speaker, three buttons and LED's, a Sony PS2 controller port, a reset button, logic and servo power inputs, an I/O bus with power and ground, and a 5vdc 250mA regulator. Up to 20 servos can be plugged in directly.



The BASIC Atom 28 Pin:

The powerful Basic Atom is faster and has more memory than a BS2. This chip is easy to program and reliable to use. It can be plugged into the Bot Board for complete access to all of the I/O pins. It is BS2 Pin Compatible and includes O'Scope and In Circuit Debugger (ICD).

The SSC-32 Servo Controller:

This is the best servo controller available which has 32 channels of 1uS resolution servo control. It has features such as Bidirectional communication with Query commands, Synchronized or "Group" moves and 12 built in Servo Hexapod Gait Sequencer, MiniSSC-II emulation.



The DB9 Serial Data Cable - 6':

The DB9 Serial Data Cable is used to connect to a Bot Board II or SSC-32 servo controller with the computer in order to configure the Mini-Hexapod.

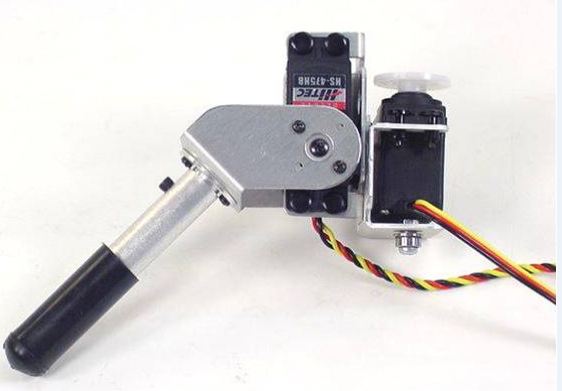
**4. Design and Implementation**

This project has both hardware and software components.

The hardware component design mainly consists of assembling the Mini-Hexapod. There are three main parts in assembling Mini-Hexapod which are as follows:

4.1 Assembling the 2DOF Legs

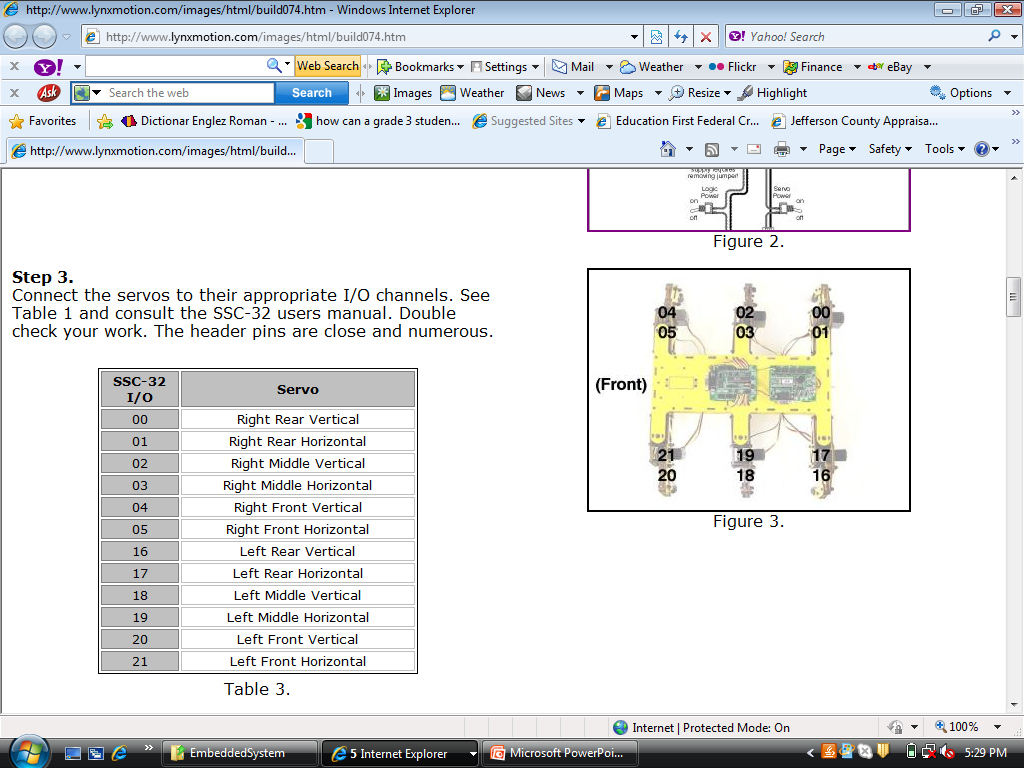
* Attach the Multi-Purpose brackets together.
* Attach the ball bearing that comes with the offset "C" bracket to the Multi-Purpose bracket.
* Attach a tubing connector hub to the offset "C" bracket.
* Attach the offset "C" bracket to the Multi-Purpose bracket.
* Connect a 3.0" tube to the hub and attach a rubber foot to the end of the tube.
* Install the servos such that the HS-422 servo goes in the horizontal position, and the HS-475HB servo goes in the vertical position.
* Now the leg is ready to be attached to a body.

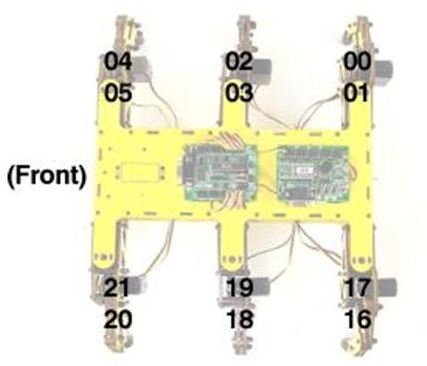


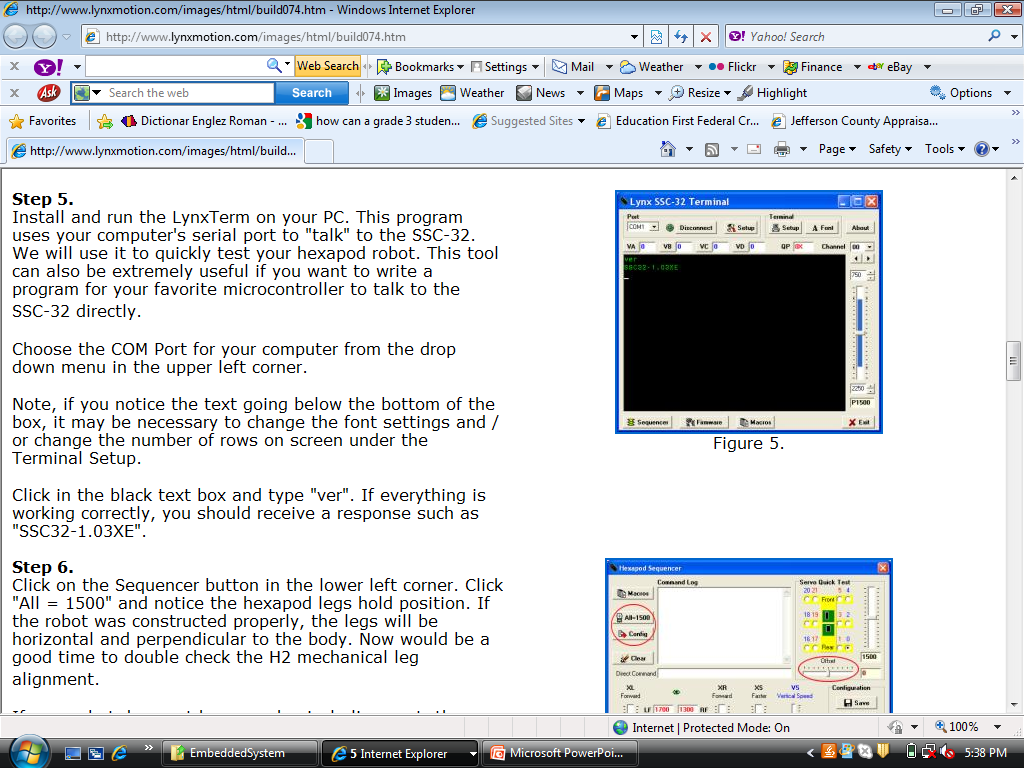
4.2 Mounting the Bot Board and SSC-32

* Install the Bot Board and SSC-32 onto the top of the robot.
* Connect power wires for servos.
* Install VS1 = VS2, VL = VS1, TX and RX, and both Baud jumpers for these tests. This will set the Baud rate to 115.2k and allow DB9 communications.
* These jumpers are used to set operating parameters for the board and also allow powering the microcontroller and support circuitry from the servo power supply.

4.3 Integrate the legs and the body of the Mini-Hexapod

* Connect the 2DOF legs to the Mini-Hexapod body.
* Connect the servos to their appropriate I/O channels by referring the adjacent table.
* The header pins are close and numerous.



* 1. Install and run the Lynx Terminal
* This program uses the computer's serial port to "talk" to the SSC-32.
* We used it quickly to test our Mini-hexapod robot.
* This tool can also be extremely useful if we want to write a program for our favorite microcontroller to talk to the SSC-32 directly.

4.4 Pictures of Assembling Mini-Hexapod in our project

Here are some pictures taken while our team was assembling the Mini-Hexapod.

The team was divided into three groups each, first team responsible for performing mechanical and electrical work, second for performing integration of the Mini-Hexapod and third for programming the Mini-Hexapod.



Fig 4.4.1 Assembling the Multi-Purpose brackets together for the 2DOF legs



Fig 4.4.2 Installing HS-422 and HS-475HB servos onto the 2DOF legs

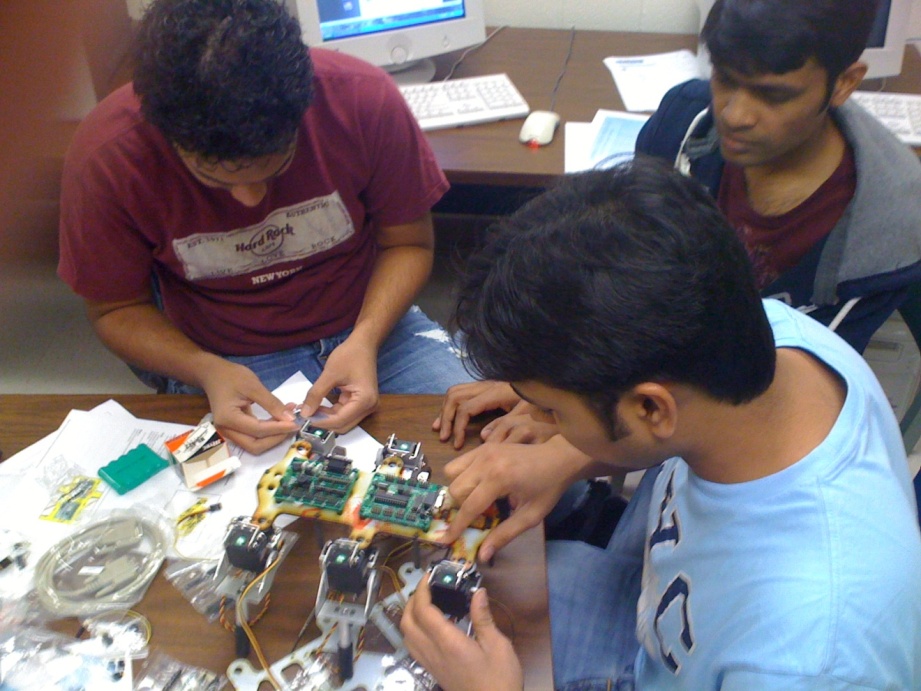


Fig 4.4.3 Attaching the 2DOF legs to the body of the Mini-Hexapod



Fig 4.4.4 Reassembling the legs with proper servo positions

One of the challenges that we faced in our project was to connect the servos of the 2DOF legs to the appropriate I/O channels of the SSC-32 board and to get correct jumper settings. Initially we found it confusing to follow the instructions for the jumper settings in the given tutorial, however we worked out an appropriate way to fix it and also located appropriate I/O channels for the connection between the servos and the SSC-32 board.

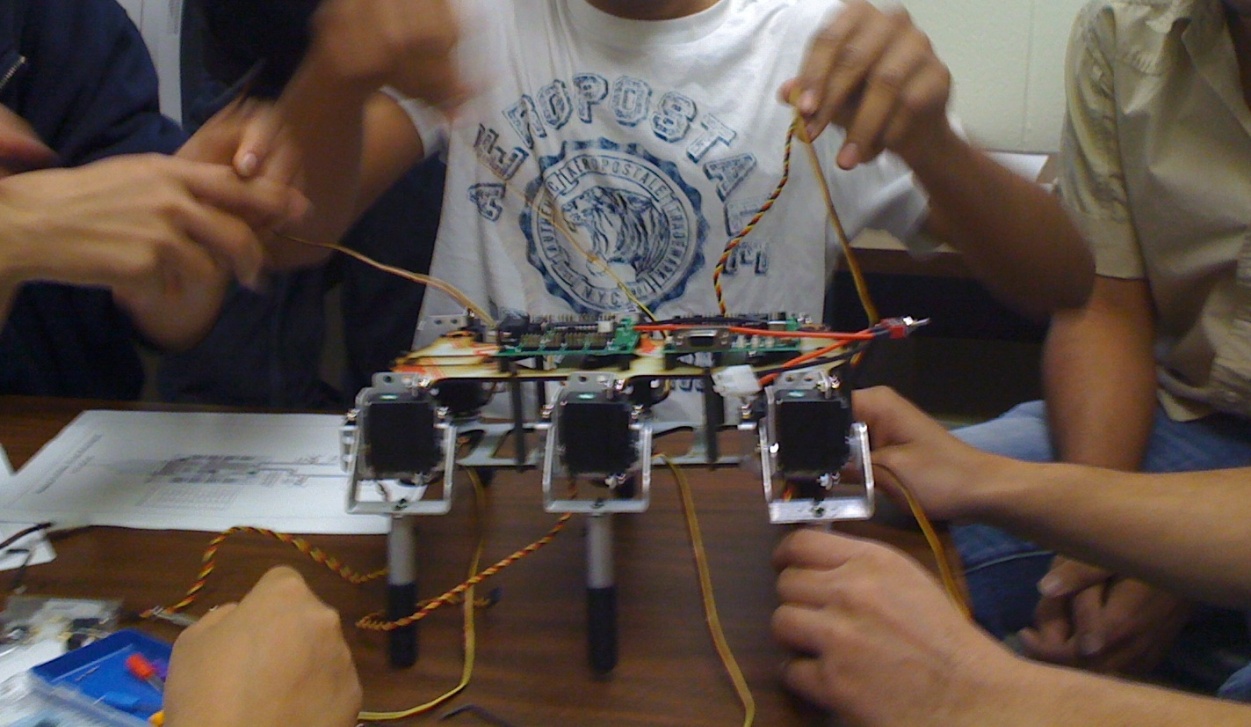


Fig 4.4.5. Connecting the servos of the 2DOF legs to the appropriate I/O channels

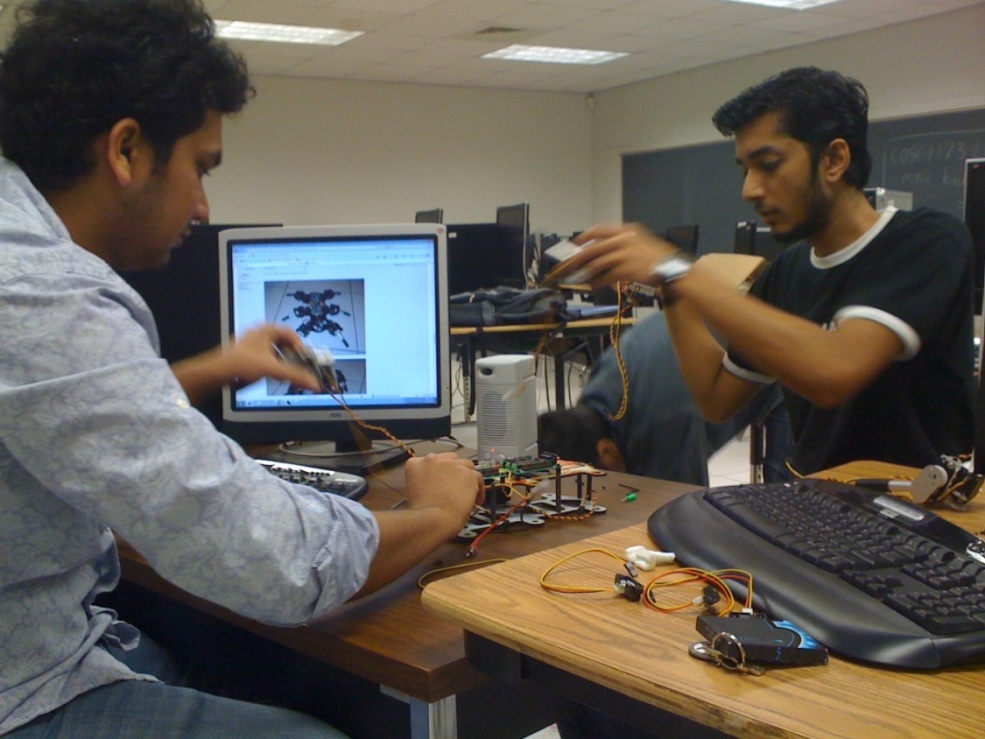


Fig 4.4.6. Installing and running the Lynx Terminal Software

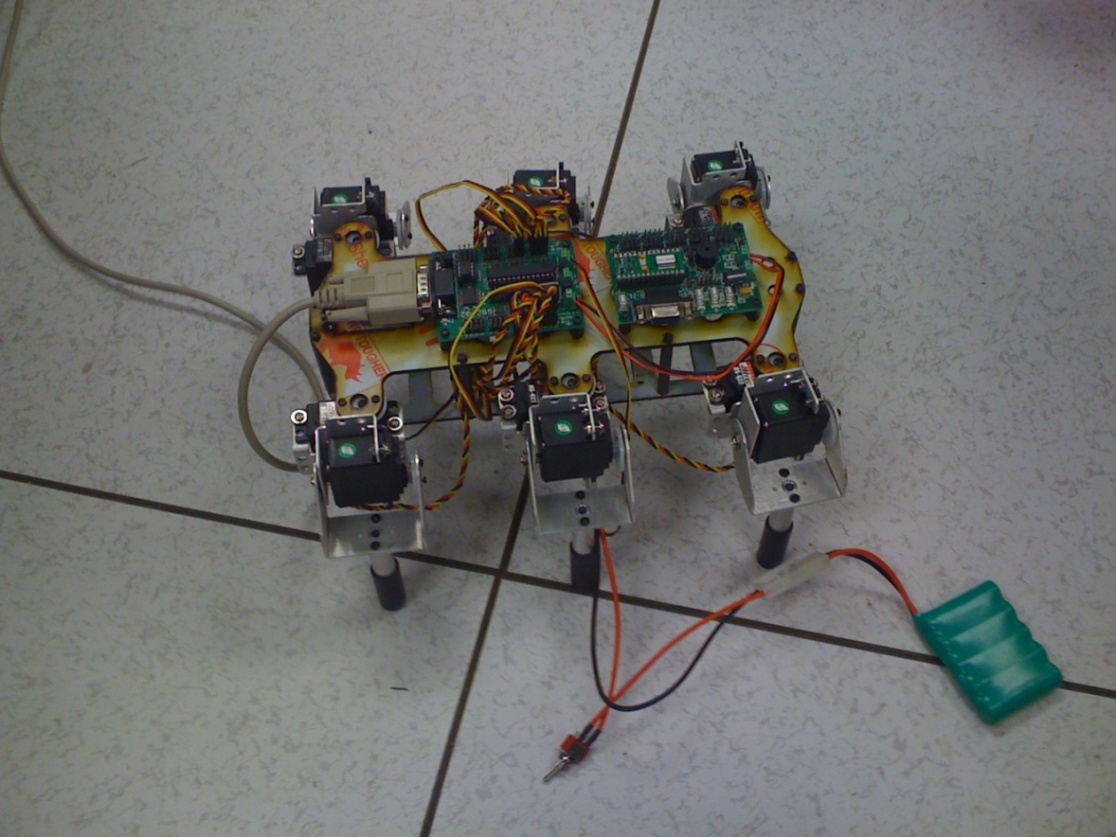


Fig 4.4.7 Configuring the Mini-Hexapod using the LynxTerm software

**5. Experimental Results**

The experimental and testing results of the Mini-Hexapod are provided in the video clips below. These videos demonstrate the motion of the Mini-Hexapod in the forward, reverse, left and right directions.

The link below demonstrates the front view of the Mini-Hexapod moving in all the directions (forward, reverse, left and right)

[****](video_2010-04-06_20.01.55.m4v)

The link below demonstrates the top view of the Mini-Hexapod moving in all the directions (forward, reverse, left and right)

[****](video_2010-04-06_20.03.43.m4v)

**6. Basic Program**

This is a Basic program used to test the configuration of the Mini-Hexapod. This code makes the hexapod walk forward.

// Declaring variables

XL var sbyte

XR var sbyte

XS var byte

XL = 100 // initializing the speed

XR = 100

XS = 100

start:

serout p15,i38400,["LF1700 RF1300 LR1300",13] // offset values

serout p15,i38400,["RR1700 LH1000 LM1700",13]

serout p15,i38400,["LL1800 RH2000 RM1300",13]

serout p15,i38400,["RL1200 VS3000 HT1500",13]

serout p15,i38400,["XL",SDEC XL,"XR",SDEC XR,13]

serout p15,i38400,["XS",DEC XS,13]

loop:  
goto loop

**6.1 Our Module**

The code below is used to make the hexapod move in all the directions i.e. forward, reverse, rotate left and right.

// declaring variables

XL var sbyte

XR var sbyte

XS var byte

Start:

// offset values

SEROUT p15,i38400,["#0PO 0 #1PO 0 #2PO 0 #3PO 0 #4PO 0 #5PO 0",13]

SEROUT p15,i38400,["#16PO 0 #17PO 0 #18PO 0 #19PO 0 #20PO 0 #21PO 0",13]

// this is where the values are sent for the legs for their initial positions

SEROUT p15,i38400,["LF1775 RF1225 LR1225",13]

SEROUT p15,i38400,["RR1775 LH1500 LM1500",13]

SEROUT p15,i38400,["LL1700 RH1500 RM1500",13]

SEROUT p15,i38400,["RL1300 VS1000 HT1700",13]

XL = 90 // speed of left legs

XR = 90  // speed of right legs

XS = 70 // vertical speed

Loop:

gosub forward // walk forward

pause 9000     // do this for 9 seconds

gosub rotateleft // turn left

pause 9000       // do this for 9 seconds

gosub rotateright // turn left

pause 9000       // do this for 9 seconds

gosub rev // turn left

pause 9000       // do this for 9 seconds

goto loop

// Forward function

forward:

// positive XL and XR values to make the robot walk forward

SEROUT p15,i38400,["XL",SDEC XL,"XR",SDEC XR,13]

SEROUT p15,i38400,["XS",DEC XS,13]

RETURN

// Reverse function

rev:

// negative XL and XR values to make the robot walk backward

SEROUT p15,i38400,["XL-",SDEC XL,"XR-",SDEC XR,13]

SEROUT p15,i38400,["XS",DEC XS,13]

RETURN

// Rotate to left function

rotateleft:

// negative XL and positive XR values to make the robot rotate right

SEROUT p15,i38400,["XL-",SDEC XL,"XR",SDEC XR,13]

SEROUT p15,i38400,["XS",DEC XS,13]

RETURN

//Rotate to right function

rotateright:

// positive XL and negative XR values to make the robot rotate right

SEROUT p15,i38400,["XL",SDEC XL,"XR-",SDEC XR,13]

SEROUT p15,i38400,["XS",SDEC XS,13]

RETURN

**7. Conclusion and Future Works**

In this project we were successful to implement an autonomous Mini-Hexapod that was able to walk in all the directions (forward, reverse, rotate left and right). However, it had certain limitations such as the hexapod was not fast enough to walk as compared to the other robots such as mini-rover. Also the Mini-Hexapod was not able to sense if there are any steps or gaps in its way, due to the lack of foot sensors.

As a future work, we plan to introduce some more of the hardware to the existing Mini-Hexapod for its better performance and more advance applications. First, we plan to introduce foot sensors that will be able to detect gaps on the surface and overcome these by taking appropriate strive to cross the gap. This can improve the performance to the great extent as the Mini-Hexapod will become more efficient in its motion. Secondly, we look forward to deal with a robotic arm on top of the Mini-Hexapod to provide more advanced applications like grabbing objects on its way and clearing the path of the Mini-Hexapod. Finally, we plan to work with a camera on to the Mini-Hexapod that will be able to provide a security application, which will allow it to scan the face of a person and make him/her authorized for a particular task.

**8. References**

1. <http://www.lynxmotion.com/Product.aspx?productID=587&CategoryID=107>
2. Dr. Stefan Andrei: Lectures notes for COSC-4301-01/COSC-5340-01: Real Time Systems. Lamar University, Department of Computer Science, Beaumont, Texas
3. E. Z. Moore and M. Buehler - Ambulatory Robotic Lab, Centre for Intelligent Machines, McGill University Montreal, QC, H3A 2A7, CANADA - “Stable Stair Climbing in a Simple Hexapod Robot”.
4. <http://en.wikipedia.org/wiki/Hexapod_(robotics)>
5. <http://spectrum.ieee.org/automaton/robotics/robotics-software/building_a_hexapod_robot_witho>
6. Jung-Min Yang - Department of Electrical Engineering, Catholic University of Daegu, Kyongsan, Republic of Korea 712-702 – “Fault-Tolerant Gait Planning for a Hexapod Robot Walking over Rough Terrain”.
7. Manuel F. Silva, J.A.T. Machado, A.M. Lopes – “Fractional Order Control of a Hexapod Robot”.