ECE 145 Senior Design Project: Biomimetic Locomotion

FINAL DOCUMENT, Winter 2004

Thai Phan
Khoa Hoang

Table of Contents

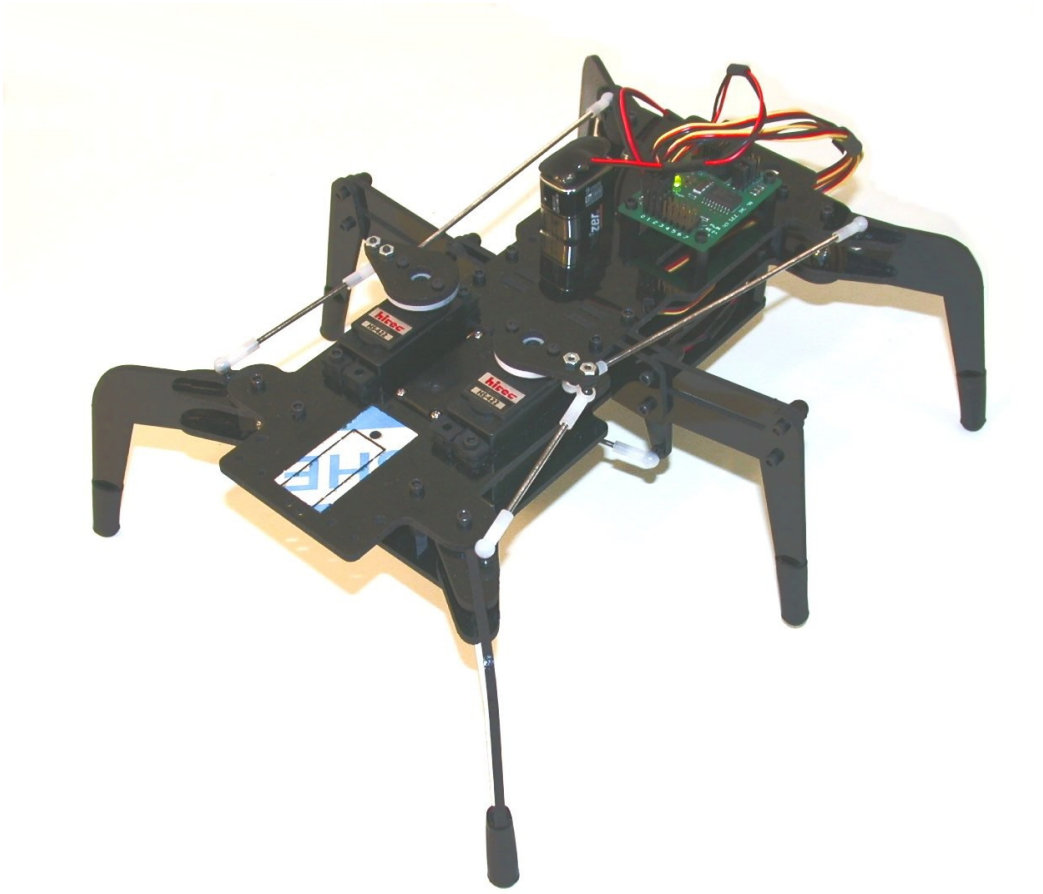
Introduction

Our senior design project is a six-legged robot. The robot is not a true robot, as it is non-autonomous. Our goal is to animate the six legs of the robot so that it can move forward, backward, left and right. Six-legged locomotion is a form of biomimetic locomotion. Biomimetic means to mimic life. Thus, biomimetic motion is motion with legs instead of wheels. The robot is not equipped with sensors, and will be unable to analyze its surroundings and respond to stimuli. The only input it gets from the outside world is through a keypad that controls its direction of movement.

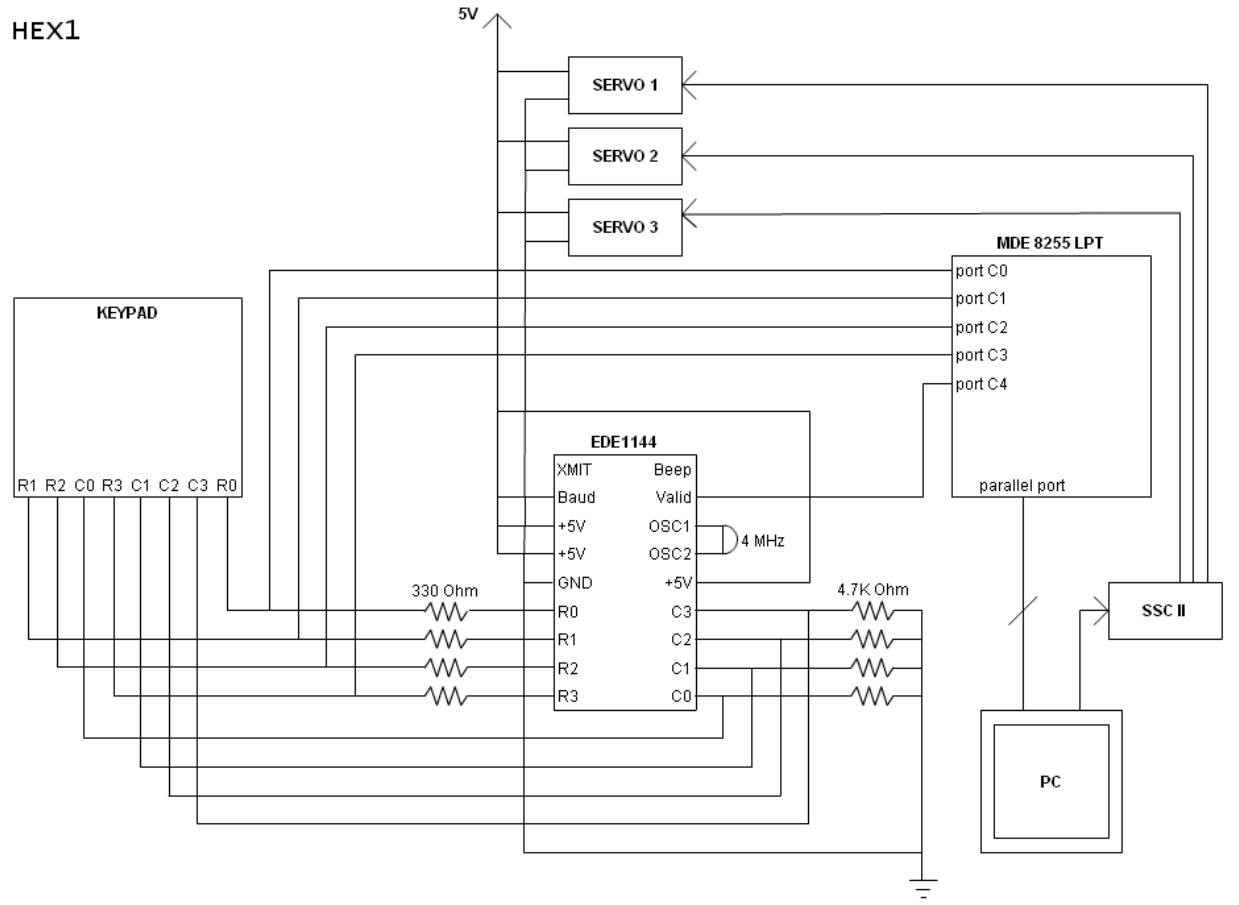
Three servos power the legs of the robot. All servos rotate in an oscillatory fashion between 45° and 135°. Rods connect the legs to the servos. Each servo is connected to two legs.

The serial servo controller is also mounted on the chassis. It provides the pulses that are necessary to control the servos. We control the serial servo controller by sending bytes to it from the serial port of our computer. The serial port is configured using a dynamic link library called ssc05.dll. The program running on our computer accepts input from the keypad, which is connected to the keypad encoder, which is connected to the Intel 8255 chip, which has connections to the computer's parallel port. The controls are as follows: 2 is forward, 4 is pivot left, 5 is stop, 6 is pivot right, 8 is reverse, and # is to exit the main program.

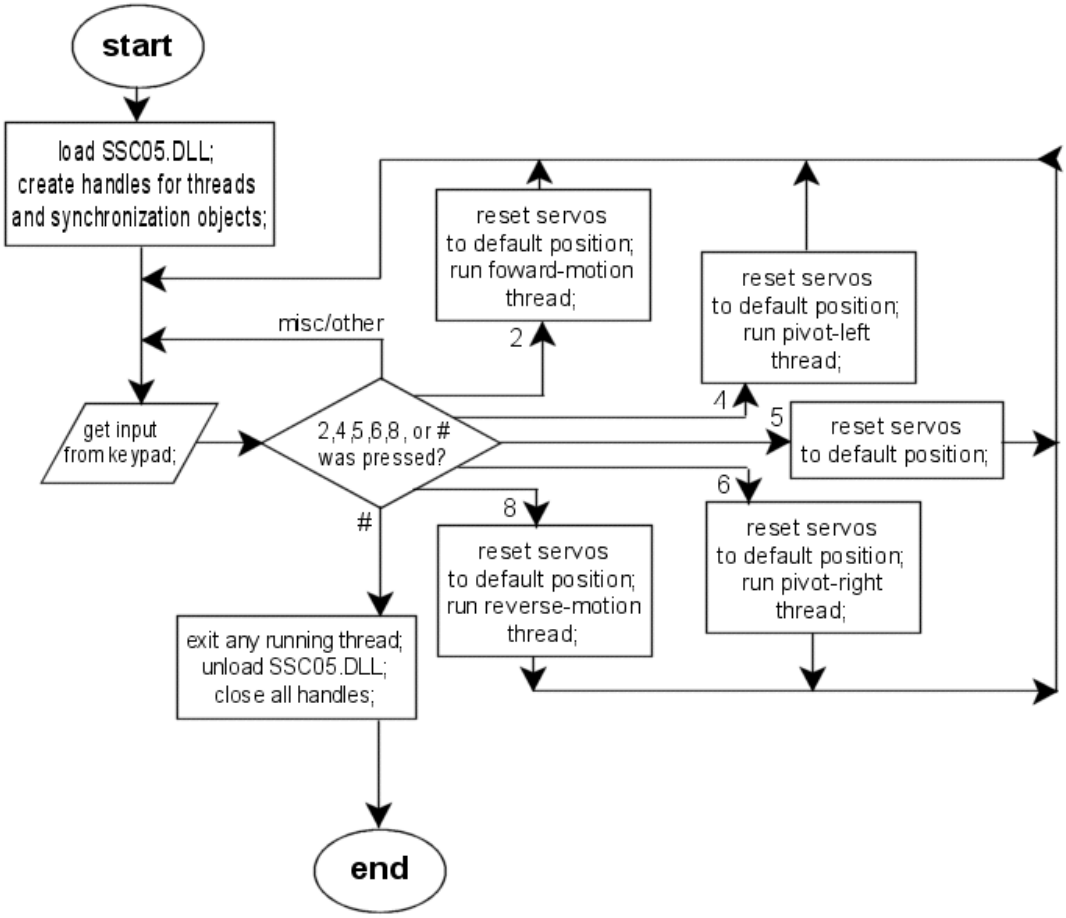
Project Image



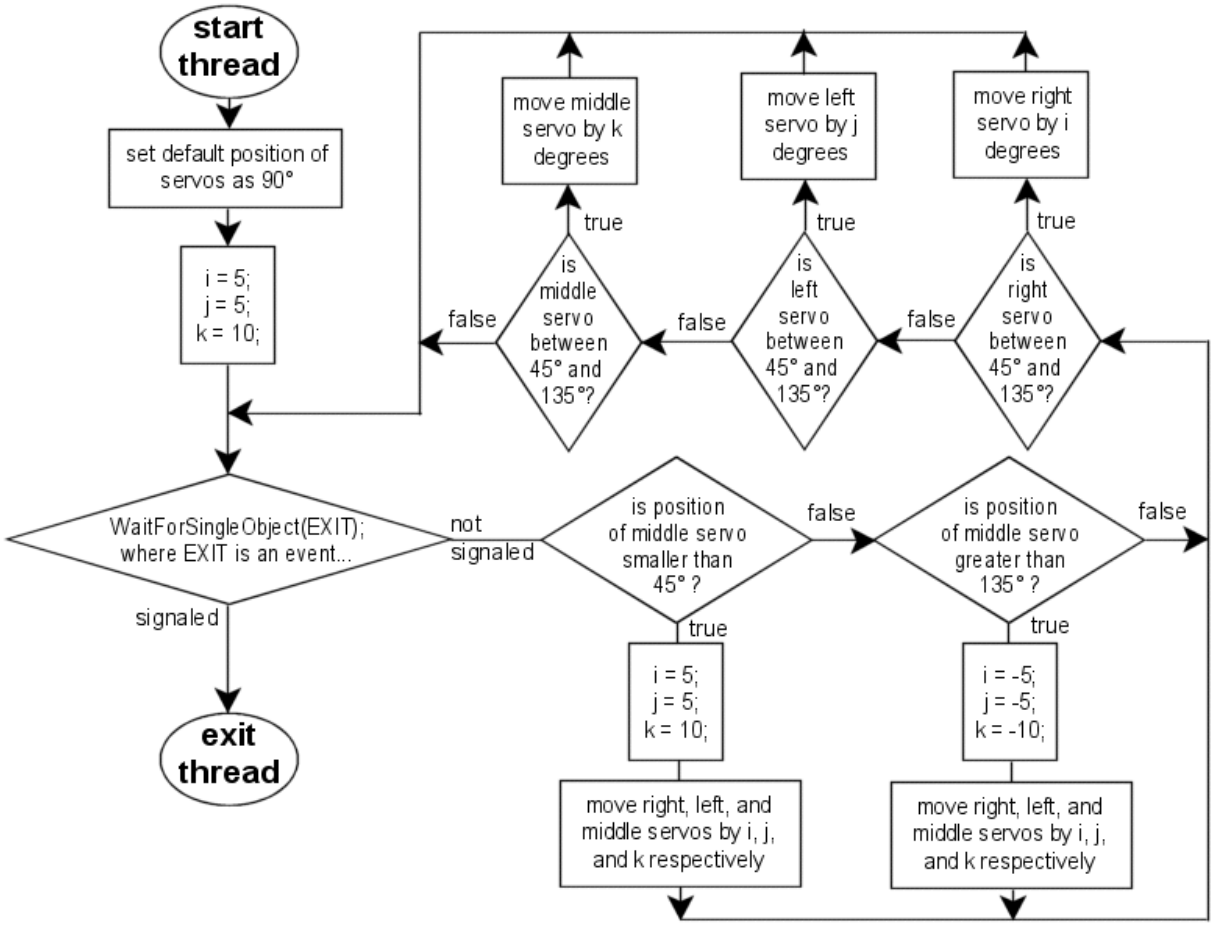
Hardware Drawing



Main Process Flow Chart



Forward-motion Thread Flow Chart



Source Code

```
#include <iostream.h>
#include <windows.h>
#include <stdio.h>
#include <iomanip.h>
#include <conio.h>
#include "mdedriverdll.h"

////// DLL loading declarations and definitions /////////
//////////////////////////////////////////////////////////

#define MAXMODULE 50
typedef void (WINAPI*ofunc)(long port, long baud);
typedef void (WINAPI*mfunc)(long no, long no2100);
typedef void (WINAPI*cfunc)();
ofunc SSC_OPEN;
mfunc SSC_MOVE;
cfunc SSC_CLOSE;

//////////////////////////////////////////////////////////
////// Handles to threads and semaphores /////////////////

DWORD id[4];
HANDLE thread1,thread2,thread3,thread4, hExitThread;

//////////////////////////////////////////////////////////
////// SERVO position constants //////////////////////////

const long cwSERVO_0  = 100;
const long nSERVO_0   = 130; //right
const long ccwSERVO_0 = 160;

const long cwSERVO_1  = 90;
const long nSERVO_1   = 130; //left
const long ccwSERVO_1 = 150;

const long cwSERVO_2  = 60;
const long nSERVO_2   = 130;
const long ccwSERVO_2 = 190;

//////////////////////////////////////////////////////////
////// Loop counter/////////////// //////////////////////

bool threadIsRunning = false;

void initLCD() {
    MDEOutPC(0x378, 0);
    MDEOutPC(0x378, 2);
    MDEOutPC(0x378, 0);
    MDEOutPA(0x378, 58);//function set 58
    Sleep(50);

    MDEOutPC(0x378, 0);
    MDEOutPC(0x378, 2);
    MDEOutPC(0x378, 0);
    MDEOutPA(0x378, 15);//initial
    Sleep(50);

    MDEOutPC(0x378, 0);
    MDEOutPC(0x378, 2);
    MDEOutPC(0x378, 0);
    MDEOutPA(0x378, 1);//initial
    Sleep(50);

    MDEOutPC(0x378, 0);
    MDEOutPC(0x378, 2);
```

```
    MDEOutPC(0x378, 0);
    MDEOutPA(0x378, 7);//initial set
    Sleep(50);

}

void display(int ivalue) {
        switch (ivalue)
        {
                case 1:
                        initLCD();

                        Sleep(50);

                        MDEOutPC(0x378, 1);
                        MDEOutPA(0x378, 94); //up
                        MDEOutPC(0x378, 3);
                        MDEOutPC(0x378, 1);
                        break;
                case 4:
                        initLCD();

                        Sleep(50);

                        MDEOutPC(0x378, 1);
                        MDEOutPA(0x378, 127); //left
                        MDEOutPC(0x378, 3);
                        MDEOutPC(0x378, 1);
                        break;
                case 5:
                        initLCD();

                        Sleep(50);

                        MDEOutPC(0x378, 1);
                        MDEOutPA(0x378, 219); //stop
                        MDEOutPC(0x378, 3);
                        MDEOutPC(0x378, 1);
                        break;
                case 6:
                        initLCD();

                        Sleep(50);

                        MDEOutPC(0x378, 1);
                        MDEOutPA(0x378, 126); //right
                        MDEOutPC(0x378, 3);
                        MDEOutPC(0x378, 1);
                        break;
                case 9:
                        initLCD();

                        Sleep(50);

                        MDEOutPC(0x378, 1);
                        MDEOutPA(0x378, 118); //down
                        MDEOutPC(0x378, 3);
                        MDEOutPC(0x378, 1);
                        break;
                case 14:
                        initLCD();

                        Sleep(50);

                        MDEOutPC(0x378, 1);
                        MDEOutPA(0x378, 88); //exit
                        MDEOutPC(0x378, 3);
                        MDEOutPC(0x378, 1);
                        break;
                default:
                        //cout << "invalid caracter" << endl;
```

```
                        break;
        }
}

DWORD WINAPI moveForward(LPVOID n)
{
        long pos_0,pos_1,pos_2,i,j,k;

        pos_0 = nSERVO_0; //right
        pos_1 = nSERVO_1; //left
        pos_2 = nSERVO_2; //middle

        i = 5;
        j = 5;
        k = 10;

        while(1)
        {
                Sleep(50);
                //i,j,k
                if (pos_2<=cwSERVO_2)
                {
                        i=5;
                        j=5;
                        k=10;
                        SSC_MOVE(0,pos_0+=i);
                        SSC_MOVE(1,pos_1+=j);
                        SSC_MOVE(2,pos_2+=k/**/);
                }
                else if (pos_2>=ccwSERVO_2)
                {
                        i=-5;
                        j=-5;
                        k=-10;
                        SSC_MOVE(0,pos_0+=i);
                        SSC_MOVE(1,pos_1+=j);
                        SSC_MOVE(2,pos_2+=k/**/);
                }
                if (!(pos_0<=cwSERVO_0 || pos_0>=ccwSERVO_0))
                        SSC_MOVE(0,pos_0+=i);
                if (!(pos_1<=cwSERVO_1 || pos_1>=ccwSERVO_1))
                        SSC_MOVE(1,pos_1+=j);
                if (!(pos_2<=cwSERVO_2 || pos_2>=ccwSERVO_2))
                        SSC_MOVE(2,pos_2+=k/**/);
                if(WaitForSingleObject(hExitThread,0)==WAIT_OBJECT_0) {
                        ResetEvent(hExitThread);
                        break;
                }
        }

        return (DWORD)n;
}

DWORD WINAPI moveBackward(LPVOID n)
{
        long pos_0,pos_1,pos_2,i,j,k;

        pos_0 = nSERVO_0; //right
        pos_1 = nSERVO_1; //left
        pos_2 = nSERVO_2; //middle

        i = 5;
        j = 5;
        k = -10;


        while(1)
        {
                Sleep(50);
                //i,j,k
                if (pos_2<=cwSERVO_2)
```

```
		{
			i=-5;
			j=-5;
			k=10;
			SSC_MOVE(0,pos_0+=i);
			SSC_MOVE(1,pos_1+=j);
			SSC_MOVE(2,pos_2+=k/**/);
		}
		else if (pos_2>=ccwSERVO_2)
		{
			i=5;
			j=5;
			k=-10;
			SSC_MOVE(0,pos_0+=i);
			SSC_MOVE(1,pos_1+=j);
			SSC_MOVE(2,pos_2+=k/**/);
		}
		if (!(pos_0<=cwSERVO_0 || pos_0>=ccwSERVO_0))
			SSC_MOVE(0,pos_0+=i);
		if (!(pos_1<=cwSERVO_1 || pos_1>=ccwSERVO_1))
			SSC_MOVE(1,pos_1+=j);
		if (!(pos_2<=cwSERVO_2 || pos_2>=ccwSERVO_2))
			SSC_MOVE(2,pos_2+=k/**/);
		if(WaitForSingleObject(hExitThread,0)==WAIT_OBJECT_0) {
			ResetEvent(hExitThread);
			break;
		}
	}

	return (DWORD)n;
}

DWORD WINAPI moveRight(LPVOID n)
{
	long pos_0,pos_1,pos_2,i,j,k;

	pos_0 = nSERVO_0; //right
	pos_1 = nSERVO_1; //left
	pos_2 = nSERVO_2; //middle

	i = -5;
	j = 5;
	k = 10;


	while(1)
	{
		Sleep(50);
		//i,j,k
		if (pos_2<=cwSERVO_2)
		{
			i=-5;
			j=5;
			k=10;
			SSC_MOVE(0,pos_0+=i);
			SSC_MOVE(1,pos_1+=j);
			SSC_MOVE(2,pos_2+=k/**/);
		}
		else if (pos_2>=ccwSERVO_2)
		{
			i=5;
			j=-5;
			k=-10;
			SSC_MOVE(0,pos_0+=i);
			SSC_MOVE(1,pos_1+=j);
			SSC_MOVE(2,pos_2+=k/**/);
		}
		if (!(pos_0<=cwSERVO_0 || pos_0>=ccwSERVO_0))
			SSC_MOVE(0,pos_0+=i);
		if (!(pos_1<=cwSERVO_1 || pos_1>=ccwSERVO_1))
			SSC_MOVE(1,pos_1+=j);
```

```
		if (!(pos_2<=cwSERVO_2 || pos_2>=ccwSERVO_2))
			SSC_MOVE(2,pos_2+=k/**/);
		if(WaitForSingleObject(hExitThread,0)==WAIT_OBJECT_0) {
			ResetEvent(hExitThread);
			break;
		}
	}

	return (DWORD)n;
}

DWORD WINAPI moveLeft(LPVOID n)
{
	long pos_0,pos_1,pos_2,i,j,k;

	pos_0 = nSERVO_0; //right
	pos_1 = nSERVO_1; //left
	pos_2 = nSERVO_2; //middle

	i = 5;
	j = -5;
	k = 10;


	while(1)
	{
		Sleep(50);
		//i,j,k
		if (pos_2<=cwSERVO_2)
		{
			i=5;
			j=-5;
			k=10;
			SSC_MOVE(0,pos_0+=i);
			SSC_MOVE(1,pos_1+=j);
			SSC_MOVE(2,pos_2+=k/**/);
		}
		else if (pos_2>=ccwSERVO_2)
		{
			i=-5;
			j=5;
			k=-10;
			SSC_MOVE(0,pos_0+=i);
			SSC_MOVE(1,pos_1+=j);
			SSC_MOVE(2,pos_2+=k/**/);
		}
		if (!(pos_0<=cwSERVO_0 || pos_0>=ccwSERVO_0))
			SSC_MOVE(0,pos_0+=i);
		if (!(pos_1<=cwSERVO_1 || pos_1>=ccwSERVO_1))
			SSC_MOVE(1,pos_1+=j);
		if (!(pos_2<=cwSERVO_2 || pos_2>=ccwSERVO_2))
			SSC_MOVE(2,pos_2+=k/**/);
		if(WaitForSingleObject(hExitThread,0)==WAIT_OBJECT_0) {
			ResetEvent(hExitThread);
			break;
		}
	}

	return (DWORD)n;
}

void resetServos()
{
	for (int i=0 ; i<5 ; i++) {
	SSC_MOVE(0,nSERVO_0);
	SSC_MOVE(1,nSERVO_1);
	SSC_MOVE(2,nSERVO_2);
	}
}

bool test(int ivalue) {
```

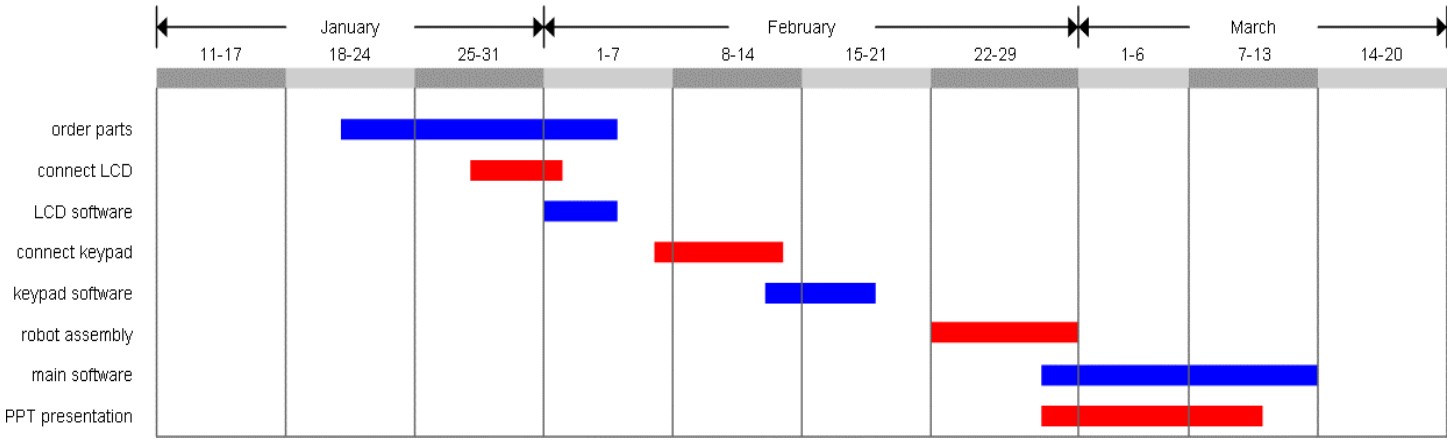
```
        display(ivalue);
        switch (ivalue)
        {
                case 1: //2
                  if (threadIsRunning)
                        SetEvent(hExitThread);
                        resetServos();
                        thread1 = CreateThread(NULL,0,moveForward,(LPVOID)0,NULL,&id[0]);
                        threadIsRunning = true;
                        break;
                case 4: //4
                  if (threadIsRunning)
                        SetEvent(hExitThread);
                        resetServos();
                        thread2 = CreateThread(NULL,0,moveLeft,(LPVOID)0,NULL,&id[1]);
                        threadIsRunning = true;
                        break;
                case 5: //5
                  if (threadIsRunning)
                        SetEvent(hExitThread);
                        threadIsRunning = false;
                        resetServos();
                        break;
                case 6: //6
                  if (threadIsRunning)
                        SetEvent(hExitThread);
                        resetServos();
                        thread3 = CreateThread(NULL,0,moveRight,(LPVOID)0,NULL,&id[2]);
                        threadIsRunning = true;
                        break;
                case 9: //8
                  if (threadIsRunning)
                        SetEvent(hExitThread);
                        resetServos();
                        thread4 = CreateThread(NULL,0,moveBackward,(LPVOID)0,NULL,&id[3]);
                        threadIsRunning = true;
                        break;
                case 14: //#
                  if (threadIsRunning)
                        SetEvent(hExitThread);
                        resetServos();
                        return true;
                        break;
                default:
                        break;
        }
        return false;
}

bool initSERVOS()
{
        char mod[MAXMODULE];
        HINSTANCE hLib=LoadLibrary("Ssc05.dll");
        if(hLib==NULL)
        {
                printf("Unable to load library!");
                getch();
                return 0;
        }
        GetModuleFileName((HMODULE)hLib, (LPTSTR)mod, MAXMODULE);
        printf("Library loaded: \n");
        SSC_OPEN=(ofunc)GetProcAddress((HMODULE)hLib, "SSC_OPEN");
        SSC_MOVE=(mfunc)GetProcAddress((HMODULE)hLib, "SSC_MOVE");
        SSC_CLOSE=(cfunc)GetProcAddress((HMODULE)hLib, "SSC_CLOSE");
        if((SSC_CLOSE==NULL) || (SSC_MOVE==NULL) || (SSC_OPEN==NULL))
        {
                printf( "Unable to load function(s)");
                FreeLibrary((HMODULE)hLib);
                return 0;
        }
        SSC_OPEN(1,9600);
```

```
        printf("ssc open done!\n");
        return 1;
}

bool exitSERVOS()
{
        SSC_CLOSE();
        printf("servo close\n");
        return 1;
}

void main()
{
        double idriverstatus;

        printf("\nOpening printer port...");
        idriverstatus = MDEOpenLPTPort(0x378);
        if(idriverstatus == 0)
                printf("Failed");
        else
                printf("Passed");

    if (initSERVOS()){
                initLCD();

                hExitThread = CreateEvent(NULL, TRUE, FALSE, "ExitThread");

                int ivalue;
                int ivalue2 = 300;
                MDEConfigPort(0x378, 0x82);

                do {
                        ivalue = MDEInPB(0x378);
                        if (ivalue < 16 && ivalue!=ivalue2) {
                                ivalue2 = ivalue;
                                if (test(ivalue2))
                                        break;
                                Sleep(500);
                        }
                } while (!kbhit());

                exitSERVOS();
                CloseHandle(thread1);
                CloseHandle(thread2);
                CloseHandle(thread3);
                CloseHandle(thread4);
                CloseHandle(hExitThread);
        }

        printf("\nClosing port...");
        idriverstatus = MDECloseDriver();
        if(idriverstatus == 0)
                printf("Failed");
        else
                printf("Passed");

        printf("\n\n");
}
```

List of Components and Cost

Vector board 8 1/2x17	$ 27.57
Parallel LCD 2x20	$ 9.50
Wire 30 gauge (1 roll, need 5 colors)	$ 10.00
miscellaneous	$ 15.00
IEEE Hexa keypad	$ 7.95
IEEE EDE1144 encoder	$ 7.50
1 Intel 8255 Board	$ 82.00
Mini serial servo microcontroller	$ 42.00
Hexapod I robot kit (Model Number [H101-KT])	$ 99.95
Total Cost :	$ 304.47

Schedule



Troubleshooting and Testing Approach

The digital multi-meter was used extensively to check the continuity of current flow throughout our project, making sure all components were properly wired and working. Our visual studios C++ compiler and debugger proved very useful while troubleshooting our coding portion of the project. We ran our tests and simulations from the compiler making sure the servos were rotating at the right angular direction and at the right time. This also meant that we had to maintain a properly functioning robot, making sure that the screws were still on. We first tested the servos separately, notifying if they moved at all or not. Then though our use of diagrams via flash, we were able to theoretically picture the robotic movements using the servos. Once the robot was built, we had minor spasming problems with the robot because our serial servo controller couldn't not relay some of the signals provided via our program properly to our servos. So we tried and tried again, taking baby steps and eventually having the robot to impressively walk forward, backward, turning left and right.

Safety Issues

Foremost we must work in a safe and clean environment away from children and pets that can get in the way of production. We did use power tools such as drills, so gloves and googles were required while drilling holes into the vector board. The soldering iron was used to make better wire connections, of course being a very hot tool, we simply had to be more careful and avoid getting burned. Those pin sockets are sharp enough to cut yourself underneath a fingernail.

Social Impact of this Project

Robots are made to do what people can't do or won't do. Our robot can work in a nuclear waste, chemical hazard, bomb hazard or any other type of life threatening environments. They can be used to scope out terrain, crawl through tight spaces, and even explore other planets. These robots can help save people and induce a better quality of life. But robots can also destroy life. Mount them with weapons and they could be used in the military.

Lesson Learned

By selecting biomimetic locomotion as our senior design project we were tested on a wide range of engineering disciplines with direct hands on applications. For electrical engineering, we had to measure current flow, power consumption, and wiring of our components. We extended our VDD and GND ports for easier accessing. A digital multi-meter proved very useful when combining all the components. As for mechanical engineering, we learned how servos worked and provided the movements of our robot. We had to know specifics such as payload the robot can carry (11 oz), its relative speed (5 ft per sec), its weight (1 lb 2 oz), abilities to move from a flat surface onto a decline or incline (it cant do to the configuration of its legs) etc. The materials were also important such as the Lexan Polycarbonate thermo plastic used to construct the frame of the robot and that the gears of the servos were made of plastic instead of metal which enhances its life wear since metal will grind much more aggressively against each other. As for computer engineering, we coded with C++ creating a multi-threaded program using API calls and Win32 synchronization. However we did not have to generate our own pulses within our program since we opted for the serial servo control.

As for the entire scope of the class, this was our first real exposure to hands on engineering. We came up with our own project proposals, purchased our own components and were involved from beginning to end. More importantly, we are also required to market our projects thus promoting engineering entrepreneurship, which in effect is a benefit to us as engineers and to the society that uses and relies on our ingenuity.

Bibliography

Announced Specification of HS-422 Standard Deluxe Servo <http://www.superdroidrobots.com/product_info/hs422.pdf>

Cormen / Leiserson / Rivest / Stein. *Introduction to Algorithms, 2nd Edition*. Cambridge: McGraw-Hill, 2001.

Kavianpour, Alireza, Dr. 2004 <http://eee.uci.edu/04w/15550/>

Mini SSC DLL for 32-bit Windows <http://www.seetron.com/ssc_an1.htm>

Silberschatz / Galvin / Gagne. *Operating System Concepts, 6th Edition.* New York: John Wiley & Sons Inc, 2003.