CS684 Documentation

CS684 – 2010 Project

µScribe

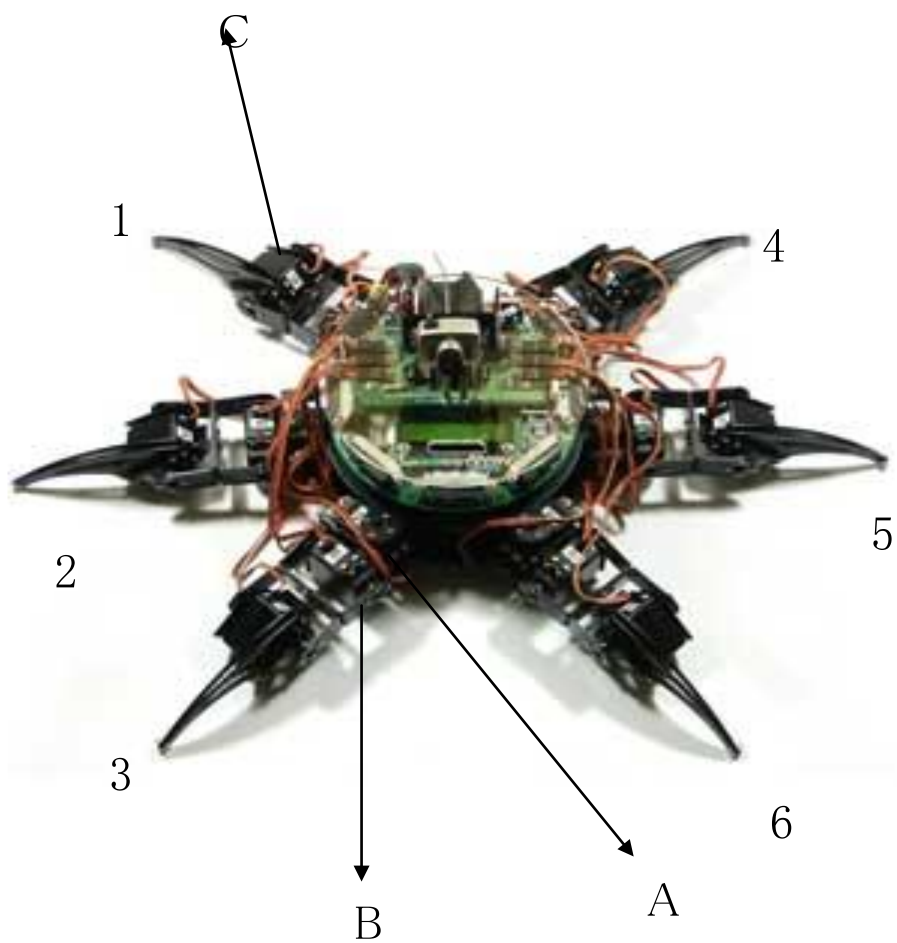
Group 10

Abhinav Maurya	**10305016**
Yogesh Kakde	**10305039**
Lokesh Rajwani	**10305066**

1. INTRODUCTION

1.1 Problem Statement

The project aims to program the Firebird V Hexapod to scribe out shapes, letters and numbers. Such an automated scribing robot can be used in many industrial applications.



Where,

A , B, C are servo motors attached to each limb.
1 to 6 are the limbs number.

1.2 Requirement Specification

- To program smooth motion of Hexapod
- To program simple natural movements
  - Rotation by any given angle, movement in any given direction
- To program simple in-place movements
  - Jerking, swaying
- To program elementary scribing movements
  - Drawing stroke, curve
- To translate shapes to a series of strokes that Hexapod can sequentially scribe

1.3 System Design

```
/********* PSEUDO CODE TO ROTATE ANTICLOCKWISE *********/


void rotate_anticlockwise()
{

 servo_calibration();  //Make all angles = 90 degrees.
 delay_250ms();


 even_A_servo(118);
 //move the A  motor of all even limbs to 118 degrees.

 even_BC_servo(45,45);
 //move the BC  motors of all even limbs to 45 degrees.

 //Now as you can observe the above two steps make the
even limbs move anticlockwise in the air.

delay_250ms();

even_BC_servo(90,90);
 //move the BC  motors of all even limbs to 45 degrees.
//This will put the even limbs on ground.

delay_250ms();


//The same steps are implemented for odd limbs.
 odd_A_servo(118);
 odd_BC_servo(45,45);
 delay_250ms();

 odd_BC_servo(90,90);
```

```
 delay_250ms();


 servo_calibration();
//Move all the motors to original position i.e. 90
degrees. As the limbs are on the ground in order to make
all angles 90 degrees the body of the robot moves in
anticlockwise direction.

}


/********* PSEUDO CODE TO ROTATE CLOCKWISE *********/
```

The same procedure will be followed for moving the hexapod in clockwise direction with one change. Instead of moving the A motor of even limbs by 28 degrees in anticlockwise direction (that is, angle given to motor is 118 degree). We move it in clockwise direction (i.e. angle given to motor is 62 degree).

```
/********* PSEUDO CODE TO MOVE FORWARD *********/

forward_3_legs ()
{

 Move all motors to original position.

  Even limbs raised by 45 degrees

  Even limbs move forward

  Put down raised limbs

  Odd limbs raised in air


 Even servos brought to original position. Which moves
chassis in forward direction

   Put down raised limbs

//The same steps are implemented with odd legs moving first.
This will adjust the asymmetries developed during first
step.

}
```

1.4 Assumptions and Dependencies

- The ATMEGA2560 can execute HEX files.
- The motions for particular shapes are already hard coded in the form of series of strokes in the program.
- A white plane surface is provided to the Hexapod to write.
- The servo motor speed is constant due to its self feedback mechanism.
- Due to inherent Hexapod design we might not be able to utilize each motor's possible range of motion.
- Intolerance to imprecision
- Sensitive to manufacturing asymmetries

1.5 Setup and any extensions implemented on the robot

A manual extension of pen is attached to the robot at the bottom.

1.6 Additional hardware used

Apart from the FirebirdV the hexapod consists of 6 limbs with 3 servo motors per limb, totaling to 18 servo motors.

2. PRESENT STATUS

2.1 A timeline based picture of project stating current status + requirements completed

Project Setup, and, Header Files for Basic Functions	27 Sept – 1 Oct
Rectilinear Motion	4 Oct – 8 Oct
Rotational Motion	11 Oct – 15 Oct
Noise Cancellation & Hardware Maintenance	18 Oct – 22 Oct
Pen Attachment	18 Oct – 22 Oct
Shape Writing Functions	25 Oct – 29 Oct
Documentation	1 Nov – 5 Nov

2.2 If there are any delays, why they occurred? How you have overcome issues?

The main obstacle in our progress was :

- Sensitive to manufacturing asymmetries
  1. All the legs do not touch the ground even if all the motors is given a 90 degree angle.
  2. Even if all the A motors given an angle 90 degrees, some motors don't remain at 90 degrees. That is the initial position we require before each motion.

- Intolerance to imprecision

Under ideal conditions the code written works perfectly well. Ideal conditions is that there are no asymmetries and no calibrations are required. But in our case we have to highly calibrate the limbs and hence it took us more time to only calibrate for basic movements.

To overcome the above problems first we tried through coding (to minimize the asymmetries). But that doesn’t work well, so we modified the hardware given to us. We covered all the limbs with the rubber tube. This slightly increased the grip between the limbs and the ground and solved one of the issues, but still the asymmetries couldn’t be avoided.

We tried to overcome the asymmetries by giving different angles to each motor so that the resultant motion will be asymmetric but it didn’t work out.

2.3 Critical steps in your projects: hardware, interfacing, algorithmic complexity, etc.

There are no special user interfaces except for switching on the Hexapod and Servo motors. The shapes to be drawn, alphabets to be scribed, etc. will be primarily hard-coded into the program itself.

Hardware Interface:

ATMega2560	Master Microcontroller in Firebird V
ATMega8	Slave Microcontroller in Firebird V
RAM and Flash Memory	Already available as a part of the microcontroller.
18 NRS-995 Servo Motor	used for chassis of the Hexapod.
Power Supply	External 7.4V, 1800mAh Lithium Polymer Battery.

Software Interface:

Compiler	AVR GCC(Specific to Atmel AVR processors)
Development Tools and IDEs	WinAVR, AVR Studio 4.0. ICC AVR7
Programming Languages	C, Esterel
Drivers	AVR MK II USB Connector Driver.

Communication Interface:

In System Programming (ISP)	AVR MK II USB Connector to Hexapod.

2.4 Individual roles and contributions

Project Setup, and, Header Files for Basic Functions	27 Sept – 1 Oct	Abhinav Maurya
Rectilinear Motion	4 Oct – 8 Oct	Abhinav Maurya, Lokesh Rajwani, Yogesh Kakde
Rotational Motion	11 Oct – 15 Oct	Abhinav Maurya, Lokesh Rajwani, Yogesh Kakde
Noise Cancellation Hardware Maintenance	18 Oct – 22 Oct	Lokesh Rajwani, Yogesh Kakde
Pen Attachment	18 Oct – 22 Oct	Abhinav Maurya, Lokesh Rajwani

Shape Writing Functions	25 Oct – 29 Oct	Lokesh Rajwani, Yogesh Kakde
Documentation	1 Nov – 5 Nov	Abhinav Maurya, Lokesh Rajwani, Yogesh Kakde

2.5 How much time devoted to project so far - man-days

20 man-days

3. DEMONSTRATION - Live demo + Video

- A video of current status of project was taken by lab staff during the prototype demonstration

4. FINAL ROADMAP OF PROJECT

1. The very first target in our project was to learn the motions of the hexapod. How with the help of 6 legs, it can move forward, backward, rotate, etc.
2. The above can be achieved with the help of moving 2 legs at a time or 3 legs at a time. We tried both the ways and found the appropriate method in the sense that it will result in fewer errors and less calibration needed. In our case, rectilinear motion is with the help of 2 legs at a time and rotational motion is with 3 legs at a time.
3. The motions were still of insufficient precision. Tried lots of ways to counter-effect the imprecision problems with hardware like, by changing the angles appropriately. For example, suppose the hexapod is programmed to move forward (in the direction between 1 and 4 numbered legs), then if it deviates to right, then we tried increasing angles of right side (4,5,6 A motors) and also tried decreasing angles of left side (1,2,3 A motors).
4. Finally we ended up with putting rubber tube over the limbs of hexapod, as expected it increased the grip and a slight precision in motions.
5. A dry run (without pen), so as to check out whether the hexapod is moving appropriately to scribe out certain letters.
6. Finally, we thought about how a pen could be attached to the center of hexapod.
7. After this we tested the hexapod, by actually making it scribe few letters like H, L, X, E, O.

5. TESTING

TEST CASE	RESULT
Character Scribing: Hexapod was programmed to draw the letter L	It scribed L as an angle symbol. That is, perfect 90 degree shape wasn't acquired. In second run, good L was scribed out.
Number Scribing: Hexapod was programmed to draw the number 1	Successfully scribed out 1.
Circle Scribing: Hexapod was programmed to draw a circle. (Same as scribing out a letter O)	Circle was approximated as a series of strokes
Scribing letters, H, E, X	Successively achieved with fair precision.

6. INNOVATION, CREATIVITY AND REUSABILITY INDEX OF THE PROJECT

6.1 Innovations in Project

- Attached rubber tubes to the limbs of the hexapod, which allowed hexapod to gain grip over the surface, thereby reducing motion errors.
- Tried both ways of motion, using 3 legs at a time OR using 2 legs at a time.

6.2 Reusability in project

- Separation of platform-specific code into a separate header file hexapod_firebirdv.h
- Ease of portability to future versions of Firebird by separation of hardware specific code involving timers, ports, etc.
- Each basic motion is programmed as a separate function
- Modular programming by having platform-dependent and platform-independent header files which enclose functions used by main()

7. IMPROVING THE PROCESS

7.1 Improvements in terms of project activities

- Constant weekly mentoring and monitoring to see project status, glitches.

7.2 Comments on the current schedule of events

- Project should start right after the robotics workshop in the beginning of semester.

7.3 The way the course activities have gone – specially the project?

- The guest lectures were interesting and inductive.
- The project was a great learning opportunity. We get to learn how things work in reality.

8. BUG REPORT

1. The letters which are scribed are not of equal size.
2. The curves or circles which are drawn are of fixed radius.

This happens due to the asymmetry of the robot. The radius of the circle cannot be changed as the calibration for that is not feasible.

9. FUTURE SCOPE

- Dynamic specification of message to be scribed
- Application of AI techniques like fuzzy logic, HMM to simplify coding of intelligent behaviour
- Esterel: Developing Esterel coding platform for the Hexapod
- Variety of movement like staircase climbing

10. CONCLUSION

The project initially was started by keeping an aim of scribing letters, numbers, and any shape and that too with in-place motion of hexapod. By in-place motion we mean that the hexapod will remain stationary at its 6 points of contacts with the ground, only the chassis will move in-place to scribe out letters. But as the work progressed, the equations of in-place movements were found to be terribly complex.

Then we switched to scribing while motion, and this was successfully achieved (although with slight imprecision). In this second method, as the hexapod moves, it scribes out letters.

Lastly, during the entire course of the project, we made an effort to keep the code reusable and useful to others. We achieved reusability by separating out the very core functions which are platform dependent in a separate header file hexapod_firebirdv.h, all the basic movements like forward, backward, rotate (both clockwise and anti-clockwise) in a second file hexapod.h. And lastly the third file main.c includes the main program and all the functions for scribing the letters like write_H(), write_X(), etc.

11. LEARNINGS

We have learnt that theoretical excellence may not always bode success in applied fields and endeavours. Theory must be skillfully applied to work in practice. Errors such as that from manufacturing processes and calibration errors must be accommodated while programming.

Appendices:

1. Readme
2. Program source code, with inline documentation (including code for header files with copyright, versioning and author information for corresponding files)
3. Final Presentation (printed 6 slides in a page)
4. Attach CD with everything in it, video included.