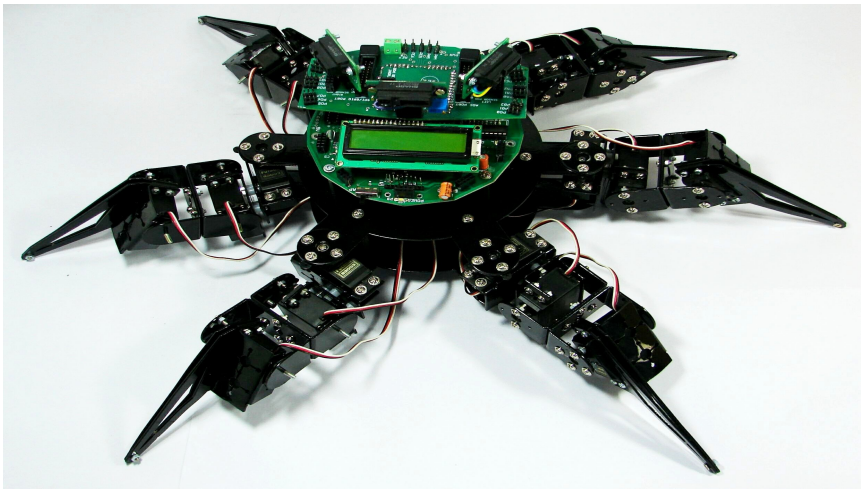
EMBEDDED SYSTEMS LAB – PROJECT SRS

FirebirdV Hexapod

Smoothening the movements of a hexapod in a RTOS Framework



Alankar Saxena – 09005021
Shivam Agarwal – 09005023
Rahul Varshneya – 09005025
Anvit Singh Tawar – 09005030

2/17/2012

This document summarizes the project on its first phase. It points out the idea and implementation details we have thought as of now regarding this project. We are sure of learning a lot by this project and we aim to develop something which can be of use at a later stage also by other related applications.

CONTENTS

1. INTRODUCTION

1.1. Problem Statement

The objective of the project is to build **'Locomotion based Hexapod Application'**. The main task is to control the locomotion of the hexapod and to make it walk both smooth and fast.

We want to model the natural walk of a spider in the hexapod by controlling the servo motors on the bot in a coordinated fashion.

Apart from this we demonstrate controlling of the hexapod locomotion with an input device (preferably keyboard) using **Zigbee module for wireless communication**.

1.2. Requirement Specification

1. Making the hexapod move smoothly in all directions.
2. Making it turn by any given angle
3. Control hexapod motion via zigbee module.

1.3. System Design

We will start with testing and experimenting with the various degrees of freedom of servo motors on the bot and analyze the various possible configurations. We will then try to simulate a smooth and fast motion of the bot (by referencing previous researches in this field including the work done on FirebirdV hexapod itself.).

Some improvements in motion can then be made by studying the biology of a spider and other related insect.

1.4. Assumptions and Limitations

1. The servo motor speed is constant due to its self-feedback mechanism.
2. Due to inherent Hexapod design we might not be able to utilize each motor's possible range of motion.
3. Intolerance to imprecision
4. Sensitive to manufacturing asymmetries
5. All the servo motors are battery driven. This leads to low battery life.
6. Speed of the hexapod is limited by the power of the servo motors.

1.5. Hardware

Apart from FirebirdV, we will use the following:

1. The hexapod consists of 6 limbs with 3 servo motors per limb, totaling to 18 servo motors which will be attached to the firebird itself.
2. Zigbee module for Wireless Communication. Here we use zigbee module for controlling the hexapod locomotion through keyboard.

2. OVERALL DESCRIPTION

2.1. System Environment

The communication with the robot is via a dedicated wireless frequency channel. The implementation of the code will be done in **RTOS Environment**.

2.2. Product Perspective

The robot (Firebird V) is controlled remotely using the ZigBee wireless interface. The user provides the input through keyboard. This input is used to call appropriate functions already coded in RTOS environment on the bot.

Some of the functionalities include:

1. Moving the bot forward using 3, 4 or 5 legs at a time.
2. Rotating the bot by some angle.

2.3. User Characteristics

The user will be responsible for moving the bot in various directions through input from the keyboard.

2.4. Constraints

1. Simulating biological movement of a spider is difficult on a mechanical hexapod due to limited degrees of freedom.
2. Controlling 18 servo motors of the bot through a single battery places restriction on the power delivered to the motors.

3. Complexity of movement (due to required coordination) makes the motion slow.

2.5. Requirement Subsets

1. Transmission of commands through zigbee.
2. Interpretation of commands.
3. Decide on action to be performed based on interpreted commands.
4. Implement the action using coordinated functioning of the various servo motors.

2.6. Implementation Details

1. The implementation in RTOS framework will be done keeping in mind the reusability of the code for related applications.
2. Code will be modular and will employ ample abstraction.
3. We will pay special attention to proper documentation of the code written.

2.7. Possible Extensions

If time permits, we will try to implement the following functionalities in our hexapod:

1. Hexapod can be made to detect obstacles automatically and avoid them by changing the path of motion.
2. The bot can be made to follow white line (with adaptive cruise control).
3. Making the bot move on an uneven terrain.