|  |  |
| --- | --- |
| **Logo_FPT_University_doc** | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Designing and making  A Lynxmotion A-POD robot controller |
|  |
| |  |  | | --- | --- | | **Hexapod Team** | | | **Group Members** | * Phan Anh Dũng Cường * Nguyễn Minh Quân * Cao Đình Nguyên Khoa | | **Supervisor** | M.Si. Trần Khánh Ninh | | **Ext Supervisor** |  | | **Capstone Project code** | APOD | |
|  |

|  |
| --- |
| - Ho Chi Minh City, 09/2013 - |

# INTRODUCTION

## PROJECT INFORMATION

* Project name: **Lynxmotion A-Pod  robot controller**
* Project code: **APOD**
* Product type: **Embedded robot controller**
* Timeline: **from September 2013 to December 2013**

## TEAM MEMBER

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Role** | **Contact** |
| **1** | Phan Anh Dũng Cường | Leader | * Phone: 0972708318 * Email: cuongpadse60774@fpt.edu.vn |
| **2** | Nguyễn Minh Quân | Member | * Phone: 0915699635 * Email: quannm60344@fpt.edu.vn |
| **3** | Cao Đình Nguyên Khoa | Member | * Phone: * Email: khoacnd60344@fpt.edu.vn |
|  |  |  |  |

Table ‑

## APOD INTRODUCTION

### **HEXAPOD**

The trademarked name “hexapod” (by Geodetic Technology) was originally for Stewart platforms used in machine tools. However, the term is now used for 6-jack platforms outside of the machine tool area, since it simply means "six legs".

In the scope of this document, “hexapod” will represent a structure with six legs which can be used for variety of Robotics application.

Variety of robotics hexapod:



Figure ‑

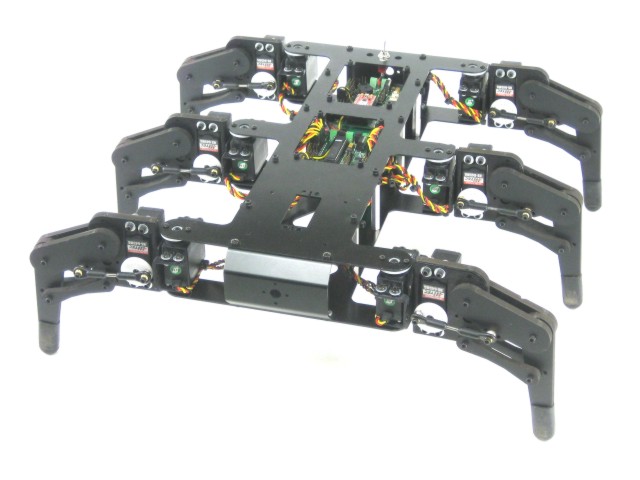


Figure ‑

### **LYNXMOTION APOD**



Figure ‑

* Insect inspired body.
* Assembly with 25 servos (motors) and PVC materials to form 6 legs, mandible grippers (with force sensor feature) and a tail.
* The three DOF (degree of freedom) leg design help the APOD moving in a variety of directions.
* Servos are directly controlled by SSC-32 board which receive command via RS232 interface from Bot board II / PC or any other devices communicate using RS232.

## EXISTING SOLUTIONS

* Flowbotics Studio: $39.9

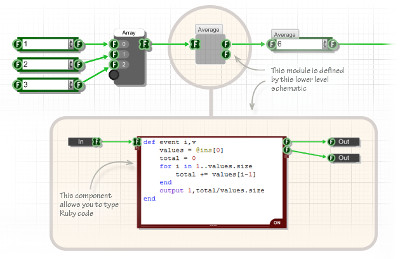


Figure ‑

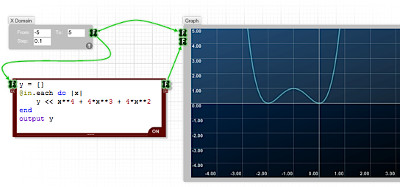


Figure ‑

* Lynxmotion Visual Sequences: $39.94

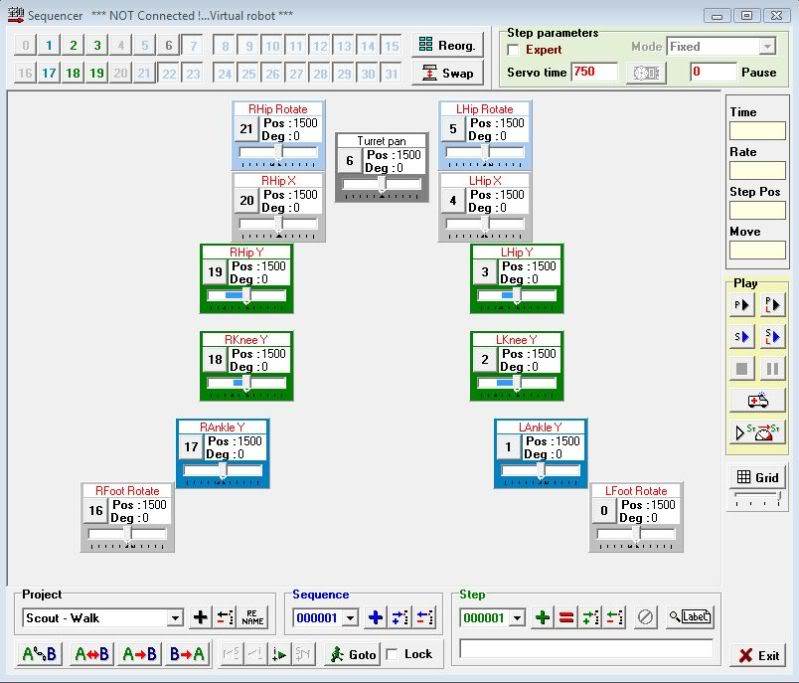


Figure ‑

## PROJECT INTRODUCTION

This project focus on programming on microprocessor unit (MCU) to developing, control the A-Pod robot. The BASIC is let him move forward, backward, right, left. Thus, we concern on control over Bluetooth. Through Camera put on robot control it/ or automatic function, connect with Sensor to discover obstructions.

Abstracts purpose: Design the controller board motor servo using microcontroller to control Lynxmotion A-Pod  robot  with available mechanical. Implement the hardware and software on PC to control Lynxmotion. The system requires Microcontroller ARM LPC,AT91SAM,PIC,AVR. Finding critical points and find best way algorithms.

## APPROACHES

* Building SSC32 firmware with ATMega328p microcontroller to control 25 servos.
* Using Bot Board II as universal sensor reader to serve automation purposes (additional sensors will be needed).
* Additional STM32F4 Discovery using ARM will be used as a central processing module which communicate with other device via Bluetooth.
* IP Camera to provide an insight view for user.
* A friendly HMI program will be provided for user to control the APOD with ease.

## OTHER

### **WHY WE NEED APOD ROBOT?**

The main purpose of A-Pod Robot is to be used at home with variable simple tasks. The Project ‘s scope is to control A-Pod from a distance via a HCI program on PC (or mobile devices). A-Pod can perform task without the present of human and report back working information when needed.

### **PRODUCT**

* HMI Program on PC (Possible extend: Android/IOS).
* Servo Control Module Software (SSC32 firmware).
* Sensor Service Module Software (Bot Board II firmware)
* Central Processing Module Software (STM32F4 Discovery).

# SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)

## PROBLEM DEFINITION

### **NAME OF THIS CAPSTONE PROJECT**

* + Project name: **Lynxmotion A-Pod  robot controller**
  + Project code: **APOD**
  + Product type: **Embedded robot controller**

### **PROBLEM ABSTRACT**

As mentioned above, this project will focus on implement a new control system for APOD using Bluetooth/wireless. The main purpose of new controller system is to make the APOD more flexible. In order to achieve that purpose, the final product should have those characteristics:

* + Friendly HMI interface.
  + Easy to use, easy to learn.
  + Different user experience: comfort and high interaction.
  + Adaptable to new mechanical constraint, prepared for complete automation.

### **PROBLEM OVERVIEW**

#### The Current System

Current APOD control system can be control using either USB-to-serial cable or an PS2 interface. The SSC32 will be responsible for direct controlling of motor servos.

* Advantages:
  + Direct command.
  + No delay due to no-subsystem needed (USB-to-serial control).
  + Simple architecture
* Limitation:
  + Very limited range of control due to “wire” problem
  + A considerable amount of wire/cable can be obstructions for APOD movement.
  + Domain or mechanical knowledge is required to perform.
  + On-sight performance: APOD must stay insight of user viewpoint.

#### The Propose System

The system consists of 4 module:

* + Servos Control Module (SSC-32): Directly control servo ‘s movement.
  + Sensor Service Module (Bot Board II): receive all kind of sensor signal and value.
  + Central Processing Module (STM32F4 Discovery) plus Bluetooth module
  + HMI software on PC

Main features will be provided:

* + Basic movement: forward, backward, turn left, turn right.
  + High interactive user interface.

#### Boundaries of The System

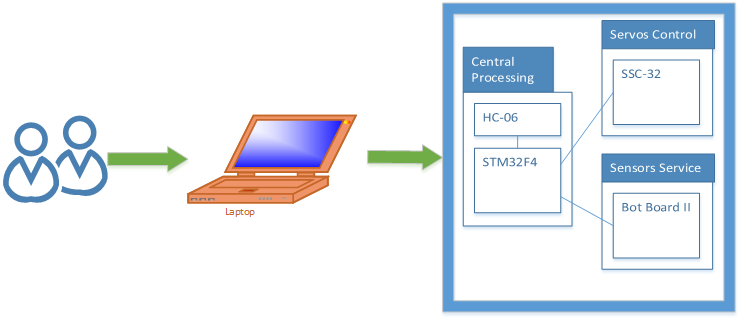


Figure ‑

* + The APOD can perform only within Bluetooth or Wireless device ‘s range
  + Any interfere while receiving command (transmitting via Bluetooth) will cause the APOD to misbehave or unpredictable movement.

#### Developing Environment

##### Hardware environment:

* Lyxnmotion APOD
* IP camera, Bluetooth device, Distance sensor.
* 4 laptops with appropriate configuration for embedded development.

##### Software environment:

* Developing environment ARV studio 4 for main board programming.
* Developing environment Visual studio 2012 with C# language for user interface programming.

## PROJECT ORGANIZATION

### **SOFTWARE PROCESS MODEL**

The process model used for developing this project is Spiral Model.

The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. Also known as the spiral lifecycle model (or spiral development), it is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping and the waterfall model. The spiral model is intended for large, expensive and complicated projects.



Figure ‑

### **ROLES AND RESPONSIBILITIES**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Full name | Team Role | Responsibilities |
| 1 | Trần Khánh Ninh | Supervisor | Define business  Support in technical issues |
| 1 | Phan Anh Dũng Cường | Team Leader | Prioritize work  Answer question and deliver direction  Facilitate productivity – maximize team performance  Complete all individual work |
| 3. | Nguyễn Minh Quân | Team Member | Commit individual product on time  Support each other to complete team work |
| 4 | Cao Đình Nguyên Khoa | Team Member | Commit individual product on time  Support each other to complete team work |

Table ‑

### **TOOL AND TECHNIQUES**

#### For Development

##### Hardware environment:

* Lynxmotion APOD
* IP camera, Bluetooth device, Distance sensor.
* 4 laptops with appropriate configuration for embedded development.

##### Software environment:

* Developing environment ARV studio 4 for main board programming.
* Developing environment Visual studio 2010 with C# language for user interface programming.
* Developing environment Keilc studio for program ARM.

#### For Management

* Microsoft Project: Task tracking
* Tortoise SVN: Source version control
* Google Code: Connect, synchronize source code and documents
* Beyond Compare: Manage changes in source files

#### For Communication

* Gmail report
* Skype chat system
* CMS system

## PROJECT MANAGEMENT PLAN

### Task

#### Initiating and Planning:

|  |  |
| --- | --- |
| Description | Registering project, kick-off meeting and planning |
| Output | Registered project, team spirit, overview plan |
| Deliverables | Draft project plan |
| Effort (man-month[[1]](#footnote-1)) | 0.8 mm |
| Dependencies and Constrains | N/A |
| Risks | Some members may be absent |

Table ‑

#### Software Requirement Analysis

|  |  |
| --- | --- |
| Description | Analyzing software requirements based on available hexapod’s hardware to create software requirements specification document |
| Output | Software Requirement Specification document |
| Deliverables | SRS document file |
| Effort (man-month) | 1.5 mm |
| Dependencies and Constrains | N/A |
| Risks | - Lack of knowledge about hexapod’s hardware |

Table ‑

#### Creating Software Design Description

|  |  |
| --- | --- |
| Description | Designing the controller for hexapod robot based on actual requirements |
| Output | Architecture design, circuits diagram, board diagram, algorithms and design specification |
| Deliverables | SDD document |
| Effort (man-month) | 3.0 mm |
| Dependencies and Constrains | Completion of SRS |
| Risks | - Choosing inappropriate algorithms and design patterns  - The hardware is hard to maintain  - Causing high coding efforts |

Table ‑

#### Coding

|  |  |
| --- | --- |
| Description | Designing the controller for hexapod robot based on actual requirements |
| Output | Architecture design, circuits diagram, board diagram, algorithms and design specification |
| Deliverables | SDD document |
| Effort (man-month) | 3.0 mm |
| Dependencies and Constrains | Completion of SRS |
| Risks | - Choosing inappropriate algorithms and design patterns  - The hardware is hard to maintain  - Causing high coding efforts |

Table ‑

#### Testing

|  |  |
| --- | --- |
| Description | Creating test case and execute test |
| Output | Test plan, test case document, test report, all tested modules and tested system |
| Deliverables | Test plan, test report |
| Effort (man-month) | 2.2 mm |
| Dependencies and Constrains | Completion of SRS, SDD, coding |
| Risks | * Lack of professional testers in team * Unit test may not be performed thoroughly causing spending many efforts in system test phase.   - Hardware ‘s limitations and errors  - ES testing is different with IS testing |

Table ‑

#### Deployment

|  |  |
| --- | --- |
| Description | Deploying system include : hexapod robot and hexapod controller simulator |
| Output | Software packages, user manual |
| Deliverables | Software packages, user manual |
| Effort (man-month) | 0.3 mm |
| Dependencies and Constrains | Completion of all other tasks |
| Risks | - Hardware’s errors |

Table ‑

### Task sheet

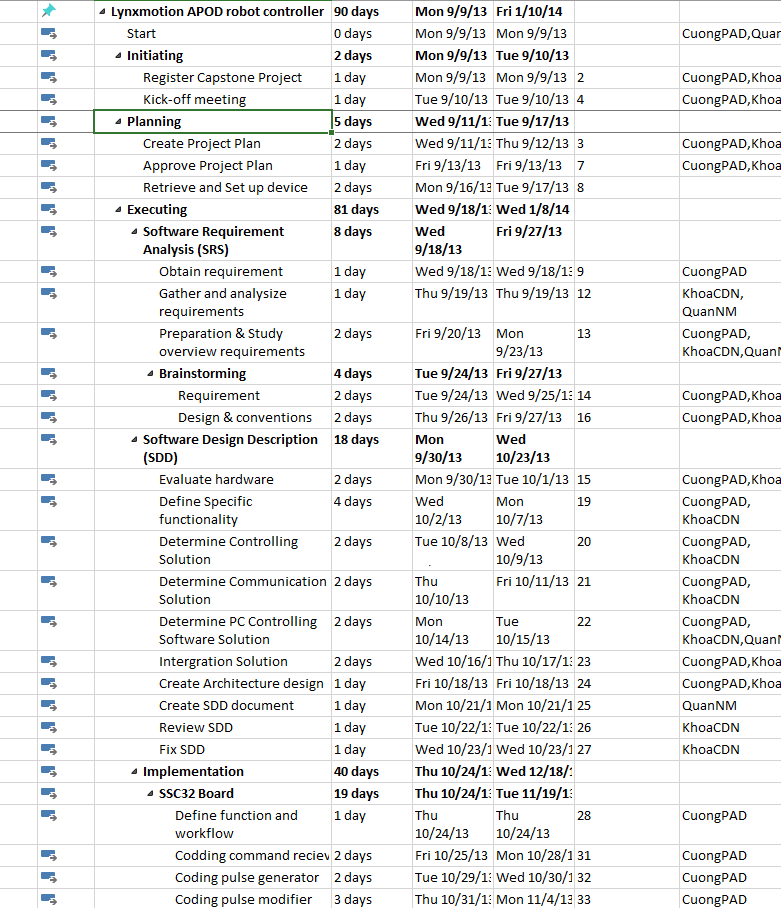


Figure 2‑1



Figure 2‑2

### All meeting minutes

<to be updated>

## CODING CONVENTION

The following rules follow:

* The standard rules for developing application using C/C++ (<http://users.ece.cmu.edu/~eno/coding/CCodingStandard.html>)

# SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

## User Requirement Specifications

* A-Pod should be able to walk freely in any direction.
* Controlling via a Serial connection like Bluetooth, or directly through a PS2 controller.
* A-Pod should be able to grab small things like a Coke.
* A-Pod should be able to recording video.
* A-Pod should be able to detect objects stand (within 30cm) in the ways and response back.
* Controller should be simple and easy.
* After receive movement signal, A-Pod should move within 2 seconds

## System Requirement Specifications

### Hardware requirement

* 25 HS-645MG servo
* 1 SSC-32 servo controller
* 1 Bot Board II with Basic Atom Pro 28 microcontroller
* 1 HC-SR04 supersonic sensor
* 1 STM32F4 ARM
* Laptops with Bluetooth Device
* 1 PS2 controller
* Few serial cables, adapters , jumpers…

### Software interface

* Menu-driven design with : button, dialog box, screen.
* Button : movement button on the right, setting like PS2 controller.
* Dialog box : on the left, display A-Pod ‘s currently speed, object stand in way and distance.
* Screen : center of menu controller, display view from camera.

### External interface

* PS2 Controller

### Functional requirement

* Direction control : user can direct the A-Pod to turn left, right, go forward or backward.
* Camera view control : user can see the view sending by the A-Pod camera.
* Grab control : user can direct the A-Pod to grab things.
* Alarm control : response to user if any object stand in the way and its distance.
* Remote control : enable remote controlling, help the user control the A-Pod through a programmed Bluetooth device.

### Non-functional requirement

* The A-Pod should response and move after receiving order within 2 seconds.
* The longest range for leg movement should be 30 centimeter, 35 degree.
* Detected range should be larger than 2 meter.
* Raise alarm sound and open Alarm Dialog Box in user interface when distance between object and the A-Pod < 20 centimeter.
* View from camera should be refresh every 0.5 second.
* The A-Pod must be compatible with the following batteries and chargers:
  + NiCad & Ni-MH Universal Smart Charger (USC-02)
  + Volt Ni-MH 2800mAh Battery Pack
* All cable, jumper, adapter should be corrected connect

## Use-case Diagram and Use-case Specifications

### Use-case Diagram

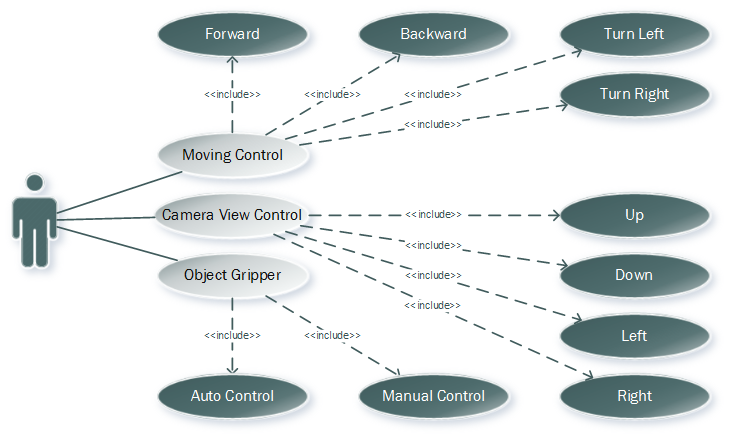
****

Figure ‑

### Use-case Specifications

#### Direction control :

* Description : Using this control, User can direct the A-Pod to move as him/her want
* Flow of events:

|  |  |  |
| --- | --- | --- |
|  | User | System |
| Normal flow | 1.User click (Left/Right/Forward/Backward) button. | 1. Left side do not detect any obstacle.  2. The A-Pod move itself to the (Left/Right/Forward/Backward) side. |
| Alternative flow | 1.User click (Left/Right/Forward/Backward) button.  2.Dialog Box appear. User choose to “Continue” or “Stop” | 1. Left side detect an obstacle  2. Response back to the User.  3. The A-Pod move itself to the (Left/Right/Forward/Backward) side if the User choose “Continue”. Stay if the User choose “Stop” |

#### Detect object control:

* Description : this control detect any object appear in the way, calculate the distance between that object and response back.
* Flow of events:

|  |  |  |
| --- | --- | --- |
|  | User Control Menu | System |
| Normal flow | 1.Receive distance from system  2. A dialog box show up : “Warning, Object in the way within 20cm !!!” | 1.System detect an object and response its distance back to user  2. Distance <20 cm, raise alarm sound, send signal to User Control Menu |
| Alternative flow | 1. Distance : 500 cm ( no Object) | 1.System detect nothing |

#### View control:

* Description : this control display the View from the A-Pod camera.
* Flow of events:

|  |  |  |
| --- | --- | --- |
|  | User | System |
| Normal flow | 1.User click View button  2.Screen display the A-Pod’s camera view | 1. Camera start recording and transmitting data back to the User |
| Alternative flow |  |  |

## Software System Attributes

### Reliability

* Easy to upgrade firmware.
* The hexapod controller can be replaced easily by loaded into chip if the controller has problem.
* The APOD system is guarantee by quality testing in:
  + Stability constraints.
  + Functionality.
  + Reliability.
* It’s mostly depending on hardware reliability.
* Small error margin when moving.

### Availability

* In case of electrical incident, the hexapod system will be shut down and reset automatically.
* Controller has the flexibility that allows changes in hardware design.
* Hexapod Controller can actuate the 6-legs forces to properly position the mobile plate given a desired trajectory.

### Security

* N/A

### Maintainability

* N/A

### Portability

* The hexapod controller is depend on hardware so that hexapod system do not have the portability attribute. However, the hexapod controller can easily be loaded into chips and use in different motors.
* The development environment and the language constructs used ensure portability as much as possible to avoid the limitation of software’s changes each time the hardware is upgraded or replaced.

### Performance

* Hexapod controller can control multi-motors in the same time.
* The current hardware can move correctly

## Entity Relationship Diagram

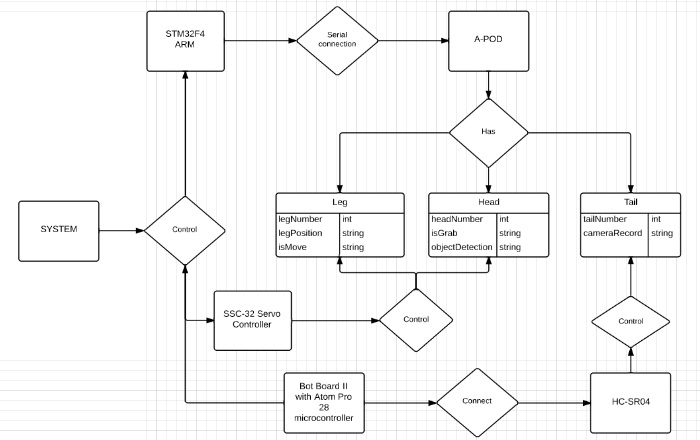


Figure ‑

# SOFTWARE DESIGN DESCRIPTION (SDD)

# SOFTWARE TEST DOCUMENTATION (STD)

# SOFTWARE USER’S MANUAL (SUM)

# APPENDIX

1. 1 man-month equals to 22 man-day [↑](#footnote-ref-1)