MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Using Bluetooth/wireless designing and making a Lynxmotion A-Pod robot controller

Group 3 - ES	
Group Members	Phan Anh Dũng Cường – Team Leader – SE60774 Nguyễn Minh Quân – Team Member – 60344
Supervisor	Mr. Trần Khánh Ninh
Ext Supervisor	N/A
Capstone Project code	APOD

- Ho Chi Minh City, 09/2013 -

This page intentionally left blank.

Acknowledgements

We would like to express our appreciation to Mr. Trần Khánh Ninh - our research supervisor – for his support and guideline throughout this project, from the beginning. His experience and advices has been a great resource for us.

We also wish to thanks various people including our family, our friends for their contribution and support for this project.

Table of Contents

A. INTRODUCTION

I. PROJECT INFORMATION

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

II. 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

Table A-1 Team Members

III. APOD INTRODUCTION

1. 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".

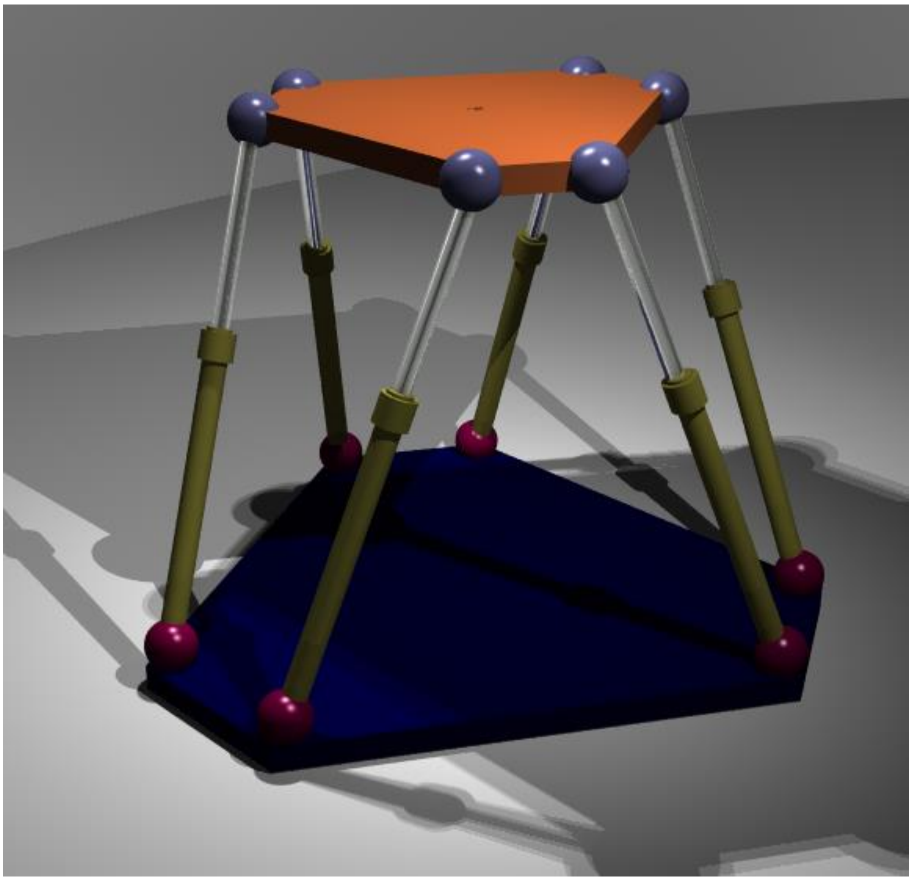


Figure A-1 Stewart platforms

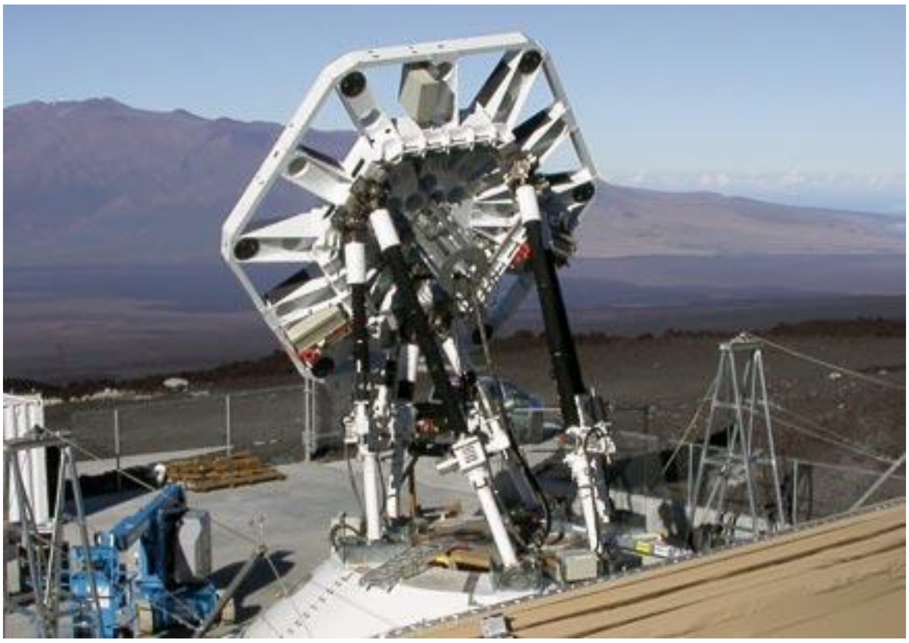


Figure A-2 Stewart platforms application

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 A-3 CH3-R Walking Robot

Figure A-4 AH2 Walking Robot

2. Lynxmotion APOD

Figure A-5 Lynxmotion APOD

- 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.

IV. EXISTING SOLUTIONS

Figure A-6 SSC-32 Board

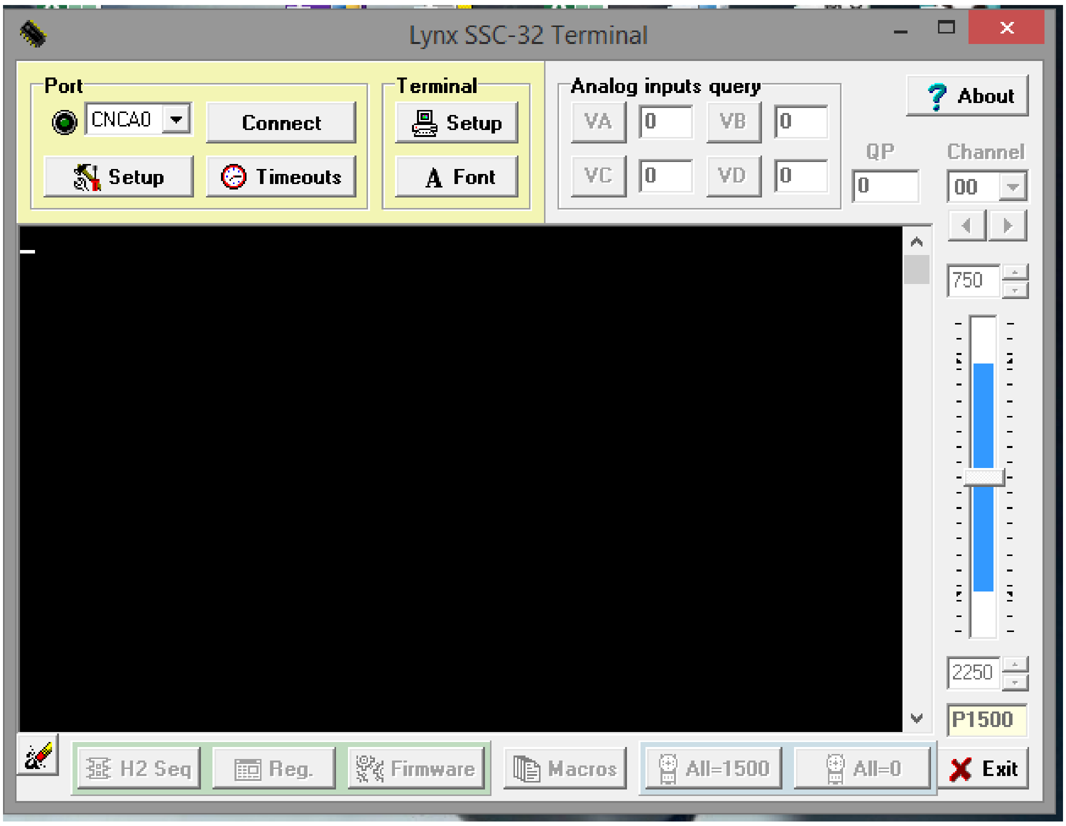


Figure A-7 LynxTerm Application

V. PROJECT INTRODUCTION

This project focus on programming on microprocessor unit (MCU) to developing, control the APOD 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 obstacles.

Abstracts purpose: Design the controller board motor servo using microcontroller to control Lynxmotion APOD 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.

VI. APPROACHES

- Building SSC32 firmware with ATMega328p microcontroller to control 25 servos.
- 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 (Human Machine Interface) program will be provided for user to control the APOD with ease.

VII. OTHER

1. Why we need APOD robot?

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

2. Product

- HMI Program on PC (Possible extend: Android/IOS).
- Servo Control Module Software (SSC32 firmware).
- Central Processing Module Software (STM32F4 Discovery).

B. SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)

I. PROBLEM DEFINITION

1. Name Of This Capstone Project

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

2. 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.

3. Problem Overview

3.1. 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 obstacles for APOD movement.
  - Domain or mechanical knowledge is required to perform.
  - On-sight performance: APOD must stay insight of user viewpoint.

3.2. The Propose System

The system consists of 3 module:

- Servos Control Module (SSC-32): Directly control servo ‘s movement and reading distance value from sensor.
- 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.

3.3. Boundaries of The System

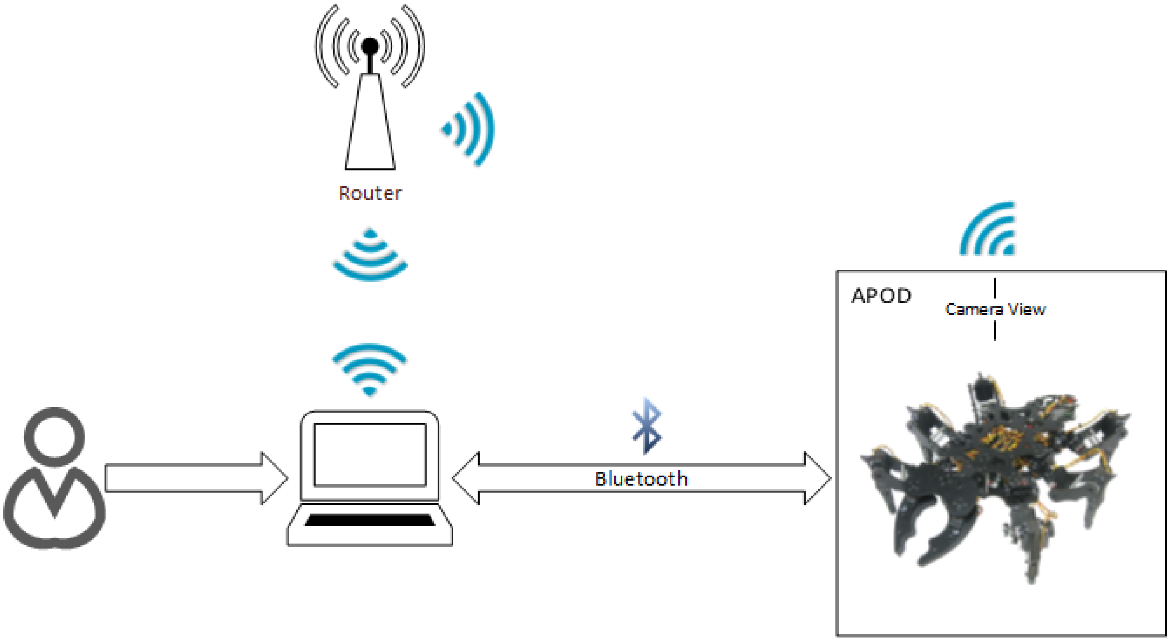


Figure B-1 System Boundary Overview

- 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.

3.4. Developing Environment

3.4.1 Hardware environment:

- Lyxnmotion APOD
- IP camera, Bluetooth device, Distance sensor.
- Laptops with appropriate configuration for embedded development.

3.4.2 Software environment:

- Developing environment CodeVisionAVR, Atmel Studio 6.1 for SSC32 board programming.
- Developing environment Keil uVision5 for STM32F4 board programming
- Developing environment Visual studio 2012 with C# language for desktop application programming.

-

II. PROJECT ORGANIZATION

1. *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.

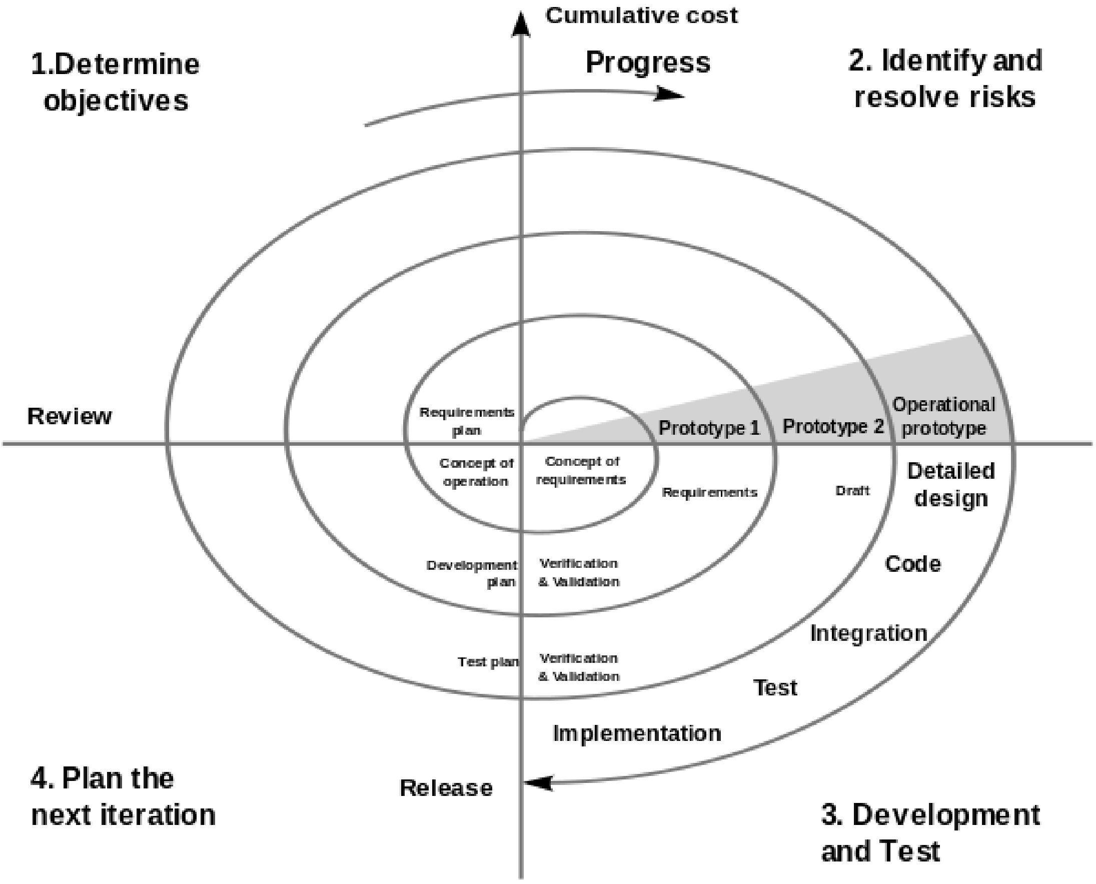


Figure B-2 Sprial model

2. Roles And Responsibilities

No.	Full name	Team Role	Responsibilities
1	Trần Khánh Ninh	Supervisor	Define business Support in technical issues
2	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

Table B-1 Roles and Responsibilities

3. Tool And Techniques

3.1. For Development

3.1.1 Hardware:

- Lynxmotion APOD
- IP camera, Bluetooth device, Distance sensor.
- Laptops with appropriate configuration for embedded development.

3.1.2 Software:

- CodeVisionAVR.
- Atmel Studio 6.1.
- Keil uVision5
- Visual Studio 2012.
- Proteus 7

3.2. 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

3.3. For Communication

- Gmail report
- Skype chat system
- CMS system

III. PROJECT MANAGEMENT PLAN

1. *Tasks*

1.1. *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 mm
Dependencies and Constrains	N/A
Risks	Some members may be absent

Table B-2 Initiating and Planning

1.2. *Software Requirement Analysis*

Description	Analyzing software requirements based on available APOD robot's hardware to create software requirements specification document
Output	Software Requirement Specification document
Deliverables	Software Requirement Specification (SRS) document file
Effort (man-month)	1 mm
Dependencies and Constrains	N/A
Risks	- Lack of knowledge about hardware

Table B-3 Software Requirement Analysis

1.3. *Creating Software Design Description*

Description	Designing the controller for APOD robot based on actual requirements
Output	Architecture design, circuits diagram, board diagram, algorithms and design specification
Deliverables	Software Design Description(SDD) document
Effort (man-month)	1 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 B-4 Creating Software Design Description

1.4. Coding

Description	Designing the controller for APOD robot based on actual requirements
Output	Architecture design, circuits diagram, board diagram, algorithms and design specification
Deliverables	SDD document
Effort (man-month)	3 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 B-5 Coding

1.5. 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 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 B-6 Testing

1.6. Deployment

Description	Deploying APOD Controller system
Output	Software packages, user manual
Deliverables	Software packages, user manual
Effort (man-month)	0.5 mm
Dependencies and Constrains	Completion of all other tasks
Risks	Hardware's errors

Table B-7 Deployment

2. Task sheet

ID		Task Mode	Task Name	Duration	Start	Finish
1			**Lynxmotion APOD robot controller**	**63 days**	**Mon 9/9/13**	**Wed 12/4/13**
2			Start	0 days	Mon 9/9/13	Mon 9/9/13
3			**Initiating**	**2 days**	**Mon 9/9/13**	**Tue 9/10/13**
4			Register Capstone Project	1 day	Mon 9/9/13	Mon 9/9/13
5			Kick-off meeting	1 day	Tue 9/10/13	Tue 9/10/13
6			**Planning**	**4 days**	**Wed 9/11/13**	**Mon 9/16/13**
7			Create Project Plan	1 day	Wed 9/11/13	Wed 9/11/13
8			Approve Project Plan	1 day	Thu 9/12/13	Thu 9/12/13
9			Retrieve and Set up device	2 days	Fri 9/13/13	Mon 9/16/13
10			**Executing**	**55 days**	**Tue 9/17/13**	**Mon 12/2/13**
11			**Software Requirement Analysis (SRS)**	**5 days**	**Tue 9/17/13**	**Mon 9/23/13**
12			Obtain requirement	1 day	Tue 9/17/13	Tue 9/17/13
13			Gather and analysize requirements	1 day	Wed 9/18/13	Wed 9/18/13
14			Preparation & Study overview requirements	1 day	Thu 9/19/13	Thu 9/19/13
15			**Brainstorming**	**2 days**	**Fri 9/20/13**	**Mon 9/23/13**
16			Requirement	1 day	Fri 9/20/13	Fri 9/20/13
17			Design & conventions	1 day	Mon 9/23/13	Mon 9/23/13
18			**Software Design Description (SDD)**	**16 days**	**Tue 9/24/13**	**Tue 10/15/13**
19			Evaluate hardware	2 days	Tue 9/24/13	Wed 9/25/13
20			Define Specific functionality	2 days	Thu 9/26/13	Fri 9/27/13
21			Determine Controlling Solution	2 days	Mon 9/30/13	Tue 10/1/13
22			Determine Communication Solution	2 days	Wed 10/2/13	Thu 10/3/13
23			Determine PC Controlling Software Solution	2 days	Fri 10/4/13	Mon 10/7/13
24			Intergration Solution	2 days	Tue 10/8/13	Wed 10/9/13
25			Create Architecture design	1 day	Thu 10/10/13	Thu 10/10/13
26			Create SDD document	1 day	Fri 10/11/13	Fri 10/11/13
27			Review SDD	1 day	Mon 10/14/1	Mon 10/14/13
28			Fix SDD	1 day	Tue 10/15/13	Tue 10/15/13
29			**Implementation**	**23 days**	**Wed 10/16/1**	**Fri 11/15/13**

Figure B-3 Project plan Page 1

ID		Task Mode	Task Name	Duration	Start	Finish
30			**SSC32 Board**	**14 days**	**Wed 10/16/1**	**Mon 11/4/13**
31			Define function and workflow	1 day	Wed 10/16/13	Wed 10/16/13
32			Codding command reciever	2 days	Thu 10/17/13	Fri 10/18/13
33			Coding pulse generator	2 days	Mon 10/21/1	Tue 10/22/13
34			Coding pulse modifier	3 days	Wed 10/23/1	Fri 10/25/13
35			**Coding servo controller**	**5 days**	**Mon 10/28/1**	**Fri 11/1/13**
36			Single/group move	2 days	Mon 10/28/1	Tue 10/29/13
37			Speed control	2 days	Wed 10/30/1	Thu 10/31/13
38			Synchronize move	1 day	Fri 11/1/13	Fri 11/1/13
39			Coding Distance sensor	1 day	Mon 11/4/13	Mon 11/4/13
40			**STM324F Board**	**17 days**	**Wed 10/16/1**	**Thu 11/7/13**
41			**IP Camera intergration**	**2 days**	**Wed 10/16/1**	**Thu 10/17/13**
42			Coding view transmition	1 day	Wed 10/16/13	Wed 10/16/13
43			Coding view display	1 day	Thu 10/17/13	Thu 10/17/13
44			Bluetooh communication	1 day	Fri 10/18/13	Fri 10/18/13
45			Coding command frame	1 day	Mon 10/21/1	Mon 10/21/13
46			**SSC32 Control**	**3 days**	**Tue 11/5/13**	**Thu 11/7/13**
47			Coding UART frame	1 day	Tue 11/5/13	Tue 11/5/13
48			Apply moving algorihm	2 days	Wed 11/6/13	Thu 11/7/13
49			Apply automation sequences	2 days	Fri 11/8/13	Mon 11/11/13
50			Intergrate SSC32	1 day	Tue 11/12/13	Tue 11/12/13
51			Intergrate STM324F	1 day	Wed 11/13/1	Wed 11/13/13
52			**Coding APOD Driver**	**2 days**	**Thu 11/14/13**	**Fri 11/15/13**
53			C# interface	1 day	Thu 11/14/13	Thu 11/14/13
54			Apply command frame	1 day	Fri 11/15/13	Fri 11/15/13
55			**Testing**	**11 days**	**Mon 11/18/1**	**Mon 12/2/13**
56			Create Unit test cases	1 day	Mon 11/18/1	Mon 11/18/13
57			Create intergration test cases	1 day	Tue 11/19/13	Tue 11/19/13
58			Perform unit test	1 day	Wed 11/20/1	Wed 11/20/13
59			Perform intergration test	2 days	Thu 11/21/13	Fri 11/22/13
60			Fixbug	2 days	Mon 11/25/1	Tue 11/26/13

Figure B-4 Project plan Page 2

ID		Task Mode	Task Name	Duration	Start	Finish
61			Create Software Test Documentation (STD)	2 days	Wed 11/27/13	Thu 11/28/13
62			Review STD	1 day	Fri 11/29/13	Fri 11/29/13
63			Fix STD	1 day	Mon 12/2/13	Mon 12/2/13
64			**Termination**	**2 days**	**Tue 12/3/13**	**Wed 12/4/13**
65			Project Package, Evaluations	1 day	Tue 12/3/13	Tue 12/3/13
66			Post-mortem Meeting	1 day	Wed 12/4/13	Wed 12/4/13

Figure B-5 Project plan Page 3

IV. CONVENTION

1. Coding style

The following rules follow:

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

2. Source File Organization

2.1. C/C++

Each module must have 2 file [File name].h and [File name].c following below description:

2.1.1 [File name].h:

- Macro declaration.
- External declaration of global variable for usage from different module
- Prototype of function/subroutine used by referenced module

2.1.2 [File name].c:

- Including [File name].h
- Declaration of external Module/Library supporting the implementation
- Declaration of external variable in [File name].h
- Declaration of variable used only within the scope of this module
- Implementation of all prototype declared in [File name].h

2.2. C#

- Each module support for 1 function will be places in 1 folder (namespace)
- External Dynamic-link library (DLL) will be place in 1 different folder from the solution with a text file listed all of the DLL needed.

C. SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

I. USER REQUIREMENT SPECIFICATIONS

- APOD should be able to walk freely in any direction.
- Controlling via a Serial connection like Bluetooth, or directly through a PS2 controller.
- APOD should be able to grab small things like a Coke.
- APOD should be able to recording video.
- APOD 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, APOD should move within 2 seconds

II. SYSTEM REQUIREMENT SPECIFICATIONS

1. *Hardware requirement*

- 25 HS-645MG servo
- 1 SSC-32 servo controller
- 1 HC-SR04 supersonic sensor
- 1 STM32F4 ARM
- Laptops with Bluetooth Device
- 1 PS2 controller
- Few serial cables, adapters , jumpers…

2. *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 APOD ‘s currently speed, object stand in way and distance.
- Screen : center of menu controller, display view from camera.

3. *External interface*

- PS2 Controller

4. *Functional requirement*

- Direction control : user can direct the APOD to turn left, right, go forward or backward.
- Camera view control : user can see the view sending by the APOD camera.
- Grab control : user can direct the APOD to grab things.
- Remote control : enable remote controlling, help the user control the APOD through a programmed Bluetooth device.

5. *Non-functional requirement*

- The APOD 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.
- View from camera should be refresh every 0.5 second.
- The APOD 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

III. USE-CASE DIAGRAM AND USE-CASE SPECIFICATIONS

1. *Use-case Diagram*

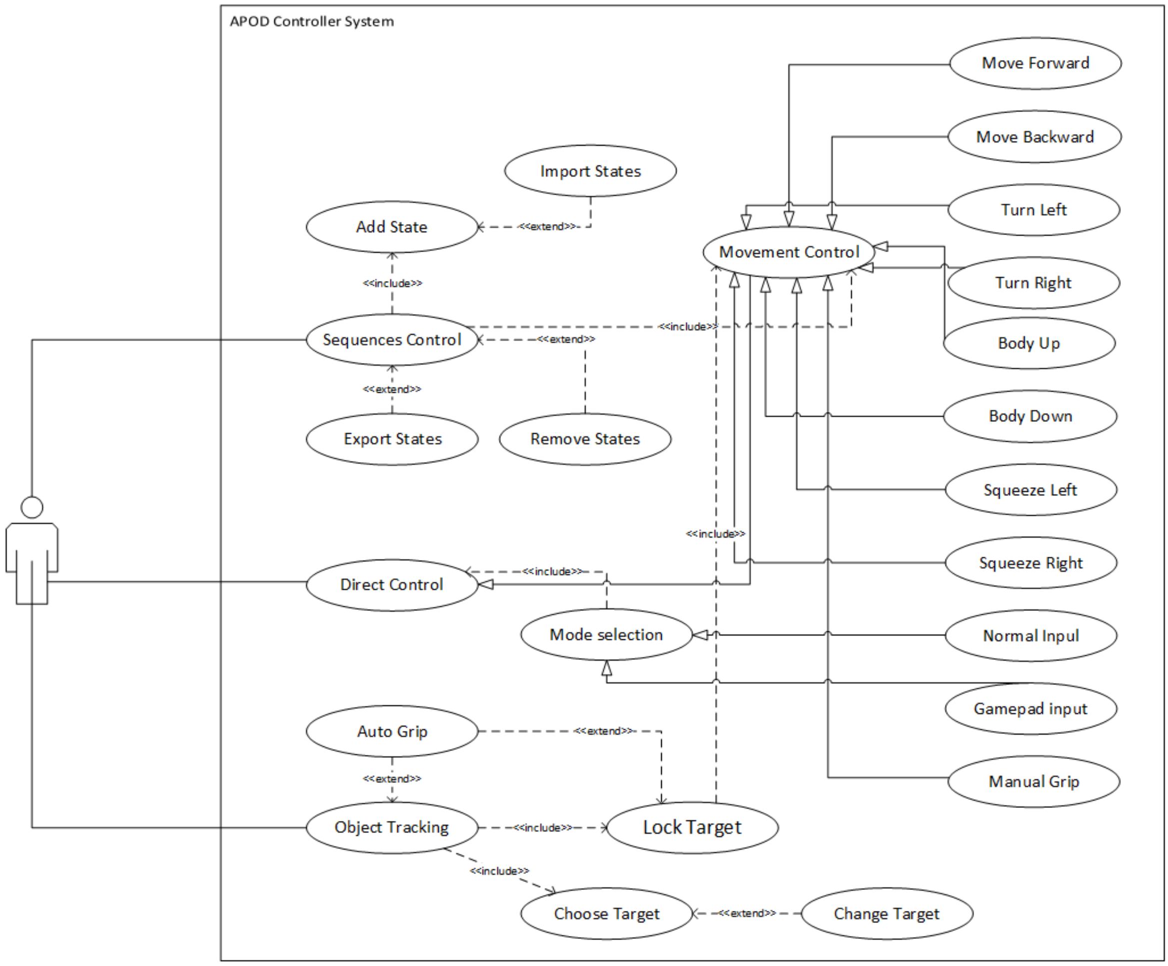


Figure C-1: Use cases diagram

2. *Use-case Specifications*

2.1. *Sequences Control*

2.1.1 *Add State*

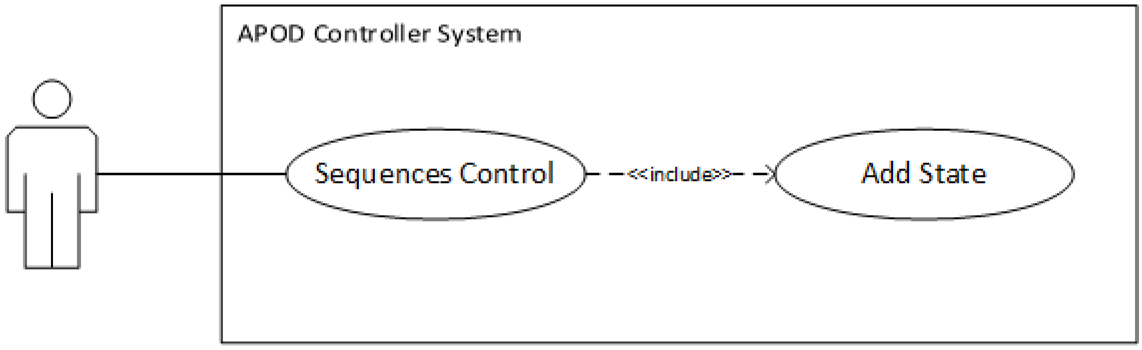


Figure C-2 Add State Use-case diagram

ADD STATE USE CASE SPECIFICATION			
Use-case No.	UC001	**Use-case Version**	1.0
Use-case Name	Add State		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Add a state (move) to sequence

Goal:

New state is added to sequence

Triggers

User click [Add] button

Preconditions:

- Sequence Player feature is selected
- Interval (if required) stay within valid range

Post Conditions:

NONE **Main Success Scenario:** - User select move to add to sequence - Enter interval value (if available) - Click [Add] **Alternative Scenario:** NONE **Exceptions:** NONE **Relationships:** NONE

Table C-1 Add state use-case specification

2.1.2 Remove states

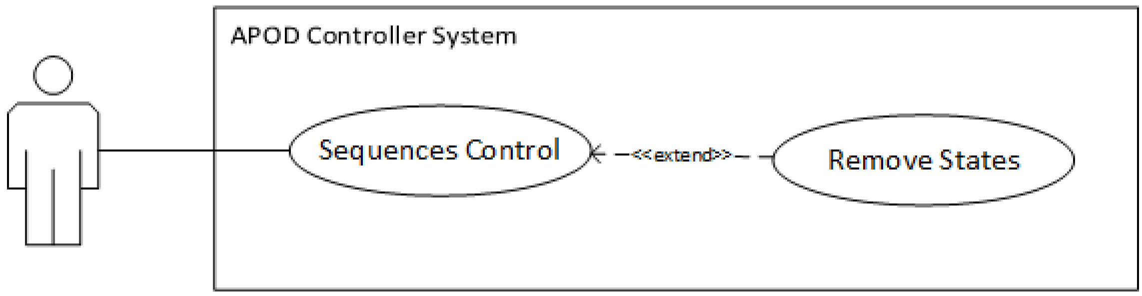


Figure C-3 Remove States Use-case diagram

REMOVE STATES USE CASE SPECIFICATION			
Use-case No.	UC002	**Use-case Version**	1.0
Use-case Name	Remove States		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal
Actor: User **Summary:**			

Remove selected states from current sequence

Goal:

All selected state to be removed

Triggers

User click [Remove] Button

Preconditions:

- Sequence Player feature is selected

Post Conditions:

NONE

Main Success Scenario:

- User select all move to be removed from sequence
- Click [Remove]

Alternative Scenario:

NONE

Exceptions:

NONE

Relationships:

NONE

Table C-2 Remove states use-case specification

2.1.3 Import States

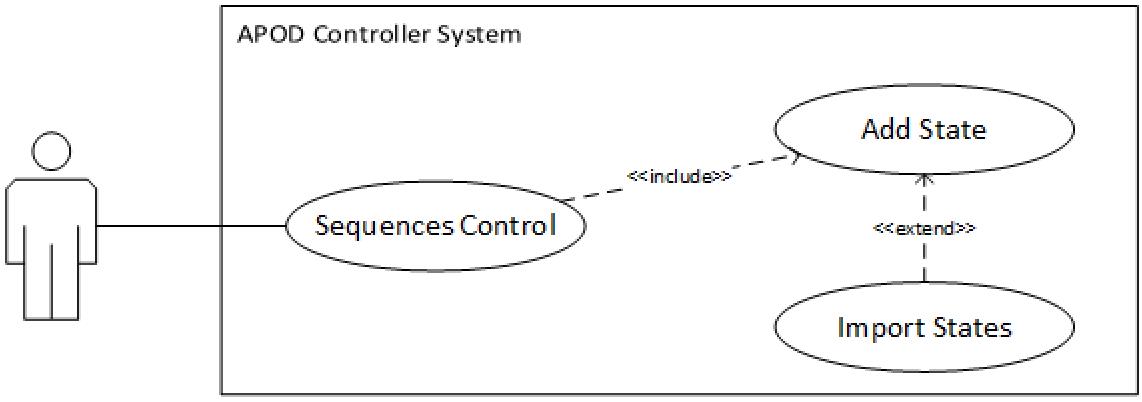


Figure C-4 Import States Use-case diagram

<table>
<tr><td colspan="4">IMPORT STATES USE CASE SPECIFICATION</td></tr>
<tr><td>Use-case No.</td><td>UC003</td><td>Use-case Version</td><td>1.0</td></tr>
<tr><td>Use-case Name</td><td colspan="3">Import States</td></tr>
<tr><td>Author</td><td colspan="3">CuongPAD</td></tr>
<tr><td>Date</td><td>05/10/2013</td><td>Priority</td><td>Normal</td></tr>
</table>

Actor:

User

Summary:

Import existing sequence from file

Goal:

Current sequence is replace by new one from file

<u>Triggers</u>

User click [Import] Button

Preconditions:

- Sequence Player feature is selected

Post Conditions:

NONE

Main Success Scenario:

- Click [Import] button
- Browse to location contain sequence file (*.xml)
- Select file & click [Ok]
- Validate new sequence content
- Replace current sequence if valid

Alternative Scenario:

NONE

Exceptions:

- IO Exception

Relationships:

NONE

Table C-3 Import States use-case specification

2.1.4 Export States

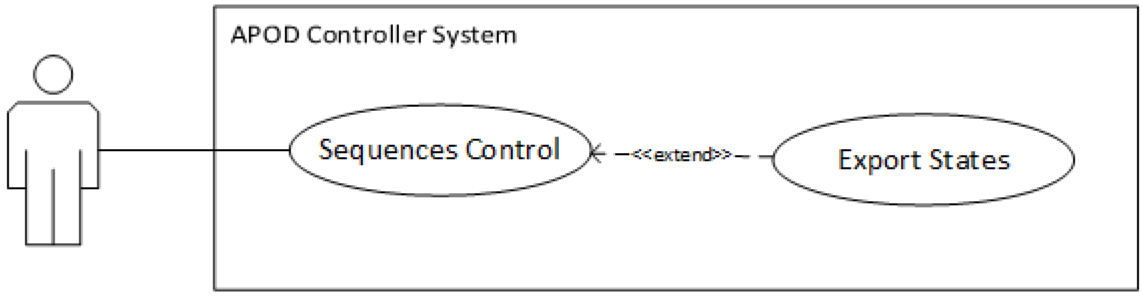


Figure C-5 Export States Use-case diagram

EXPORT STATES USE CASE SPECIFICATION			
Use-case No.	UC004	**Use-case Version**	1.0
Use-case Name	Export States		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal
Actor: User **Summary:** Export current sequence to file **Goal:** New Sequence file **Triggers** User click [Export] button **Preconditions:** - Sequence Player feature is selected **Post Conditions:** NONE			

Main Success Scenario:

- Click [Export] button
- Browse to location contain sequence file (*.xml)
- Click [Ok]

Alternative Scenario:

NONE

Exceptions:

- IO Exception

Relationships:

NONE

Table C-4 Export states use-case specification

2.2. Direct Control

2.2.1 Mode Selection

a) Normal Input

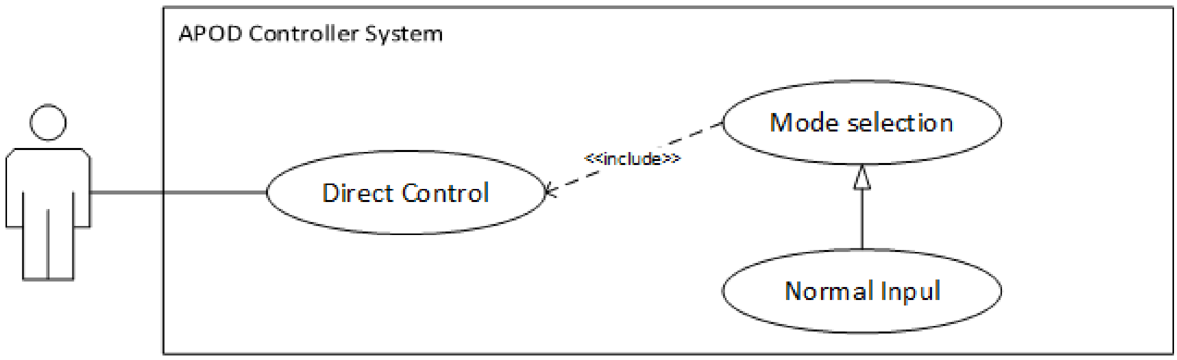


Figure C-6 Normal input Use-case diagram

NORMAL INPUT USE CASE SPECIFICATION			
Use-case No.	UC005	**Use-case Version**	1.0
Use-case Name	Normal Input Selection		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal
Actor:			

User

Summary:

Using normal input method (keyboard or Mouse)

Goal:

Normal input is selected

Triggers

User click [Normal Mode] button

Preconditions:

- Live control feature is selected

Post Conditions:

NONE

Main Success Scenario:

- User click [Normal Mode] button
- Configuration is checked
- Establish Bluetooth connection

Alternative Scenario:

NONE

Exceptions:

- IO Exception

Relationships:

NONE

Table C-5 Normal input use-case specification

b) Gamepad Input

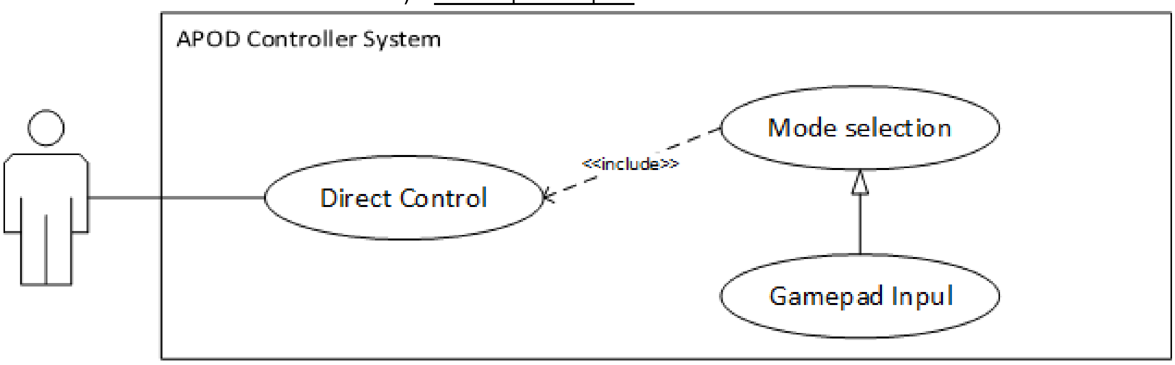


Figure C-7 Gamepad Input Use-case diagram

GAMEPAD INPUT USE CASE SPECIFICATION			
Use-case No.	UC006	**Use-case Version**	1.0
Use-case Name	Gamepad Input Selection		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Using Gamepad input method (Gamepad)

Goal:

Gamepad input is selected

Triggers

User click [Gamepad Mode] button

Preconditions:

- Live control feature is selected
- Gamepad is connected

Post Conditions:

NONE

Main Success Scenario:

- User click [Gamepad Mode] button
- Configuration is checked
- Gamepad device connection is checked
- Establish Bluetooth connection

Alternative Scenario:

NONE

Exceptions:

- IO Exception

Relationships:

NONE

Table C-6 Gamepad input use-case specification

2.2.2 Movement Control

a) Move Forward

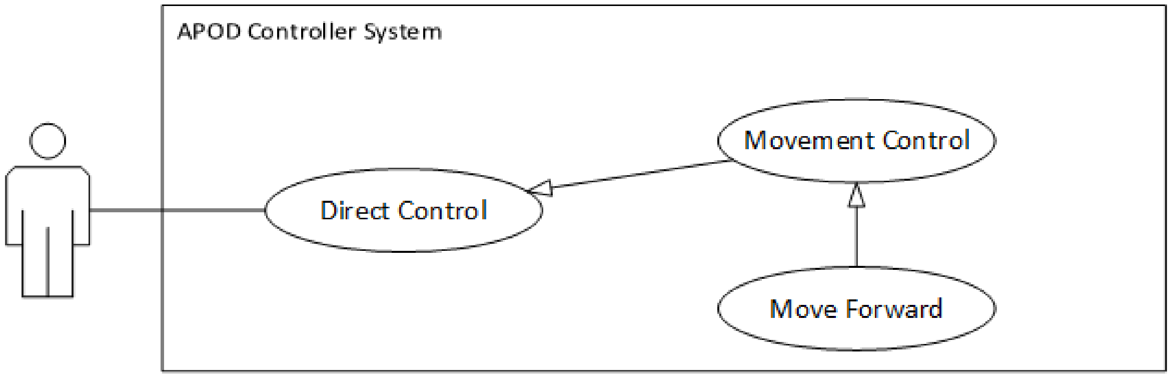


Figure C-8 Move Forward Use-case diagram

MOVE FORWARD USE CASE SPECIFICATION				
Use-case No.	UC007	**Use-case Version**	1.0	
Use-case Name	Move Forward			
Author	CuongPAD			
Date	05/10/2013	**Priority**	Normal	
Actor: User **Summary:** Command APOD to move forward **Goal:** APOD walking in forward direction **Triggers** User click [Forward] button **Preconditions:** - Live control feature is selected - Bluetooth connection is established				

- Normal input mode or Gamepad input mode is selected - [L1] and [L2] is not pressed. **Post Conditions:** - APOD stop when button is released **Main Success Scenario:** - User click on [Forward] button (⛊) **Alternative Scenario:** - Alt 1: press "W" key on keyboard (if Normal input is selected) - Alt 2: press (⛊) key on gamepad (if Gamepad input is selected) **Exceptions:** - IO Exception **Relationships:** NONE

Table C-7 Move Forward use-case specification

b) Move Backward

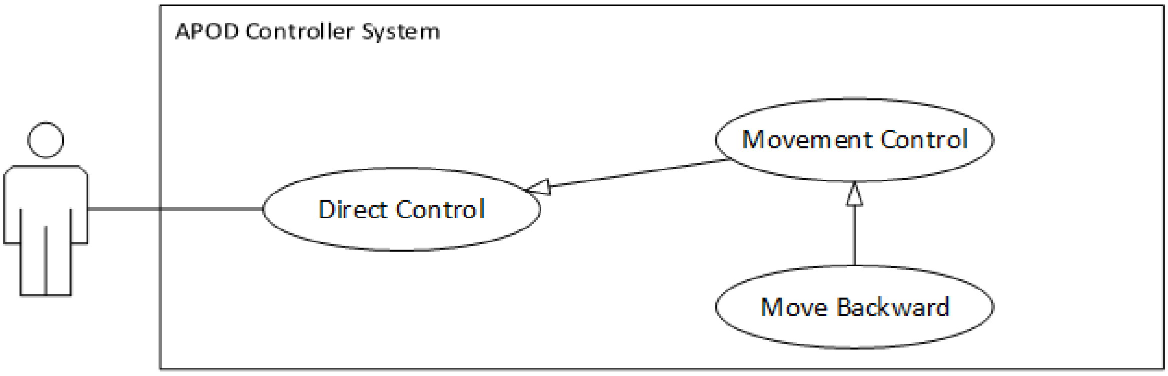


Figure C-9 Move Backward Use-case diagram

MOVE BACKWARD USE CASE SPECIFICATION			
Use-case No.	UC008	**Use-case Version**	1.0
Use-case Name	Move Backward		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Command APOD to move Backward

Goal:

APOD walking in Backward direction

<u>Triggers</u>

User click [Backward] button

Preconditions:

- Live control feature is selected
- Bluetooth connection is established
- Normal input mode or Gamepad input mode is selected
- [L1] and [L2] is not pressed.

Post Conditions:

- APOD stop when button is released

Main Success Scenario:

- User click on [Backward] button ()

Alternative Scenario:

- Alt 1: press “S” key on keyboard (if Normal input is selected)
- Alt 2: press () key on gamepad (if Gamepad input is selected)

Exceptions:

- IO Exception

Relationships:

NONE

Table C-8 Move Backward use-case specification

c) Turn Left

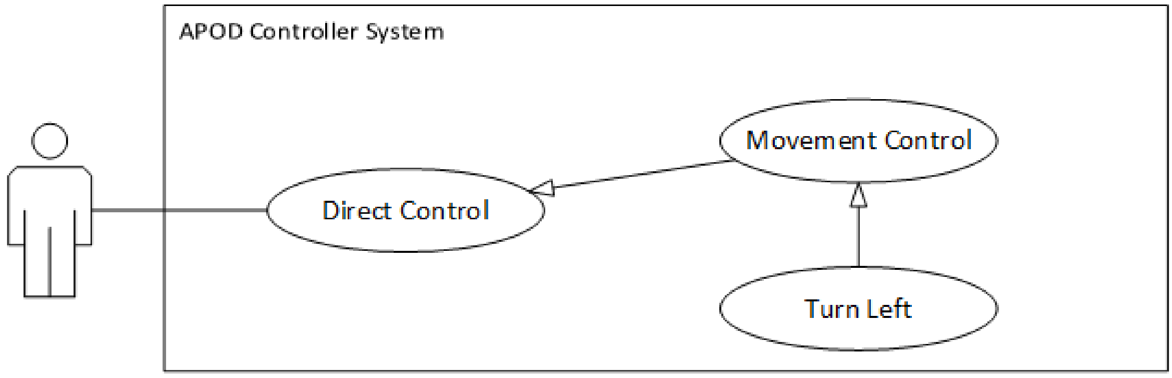


Figure C-10 Turn Left Use-case diagram

TURN LEFT USE CASE SPECIFICATION			
Use-case No.	UC009	**Use-case Version**	1.0
Use-case Name	Turn Left		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Command APOD to turn to the left

Goal:

APOD turn in Left direction

Triggers

User click [Turn Left] button

Preconditions:

- Live control feature is selected
- Bluetooth connection is established
- Normal input mode or Gamepad input mode is selected
- [L1] and [L2] is not pressed.

Post Conditions:

- APOD stop when button is released **Main Success Scenario:** - User click on [Turn Left] button (⯃) **Alternative Scenario:** - Alt 1: press "A" key on keyboard (if Normal input is selected) - Alt 2: press (⯃) key on gamepad (if Gamepad input is selected) **Exceptions:** - IO Exception **Relationships:** NONE

Table C-9 Turn left use-case specification

d) <u>Turn Right</u>

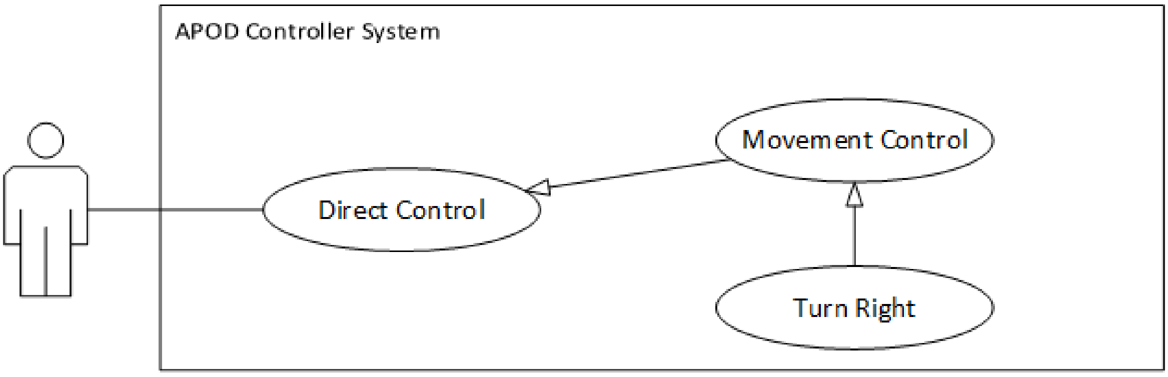


Figure C-11 Turn Right Use-case diagram

TURN RIGHT USE CASE SPECIFICATION			
Use-case No.	UC010	**Use-case Version**	1.0
Use-case Name	Turn Right		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal
Actor: User			

Summary:

Command APOD to turn to the Right

Goal:

APOD turn in Right direction

<u>Triggers</u>

User click [Turn Right] button

Preconditions:

- Live control feature is selected
- Bluetooth connection is established
- Normal input mode or Gamepad input mode is selected
- [L1] and [L2] is not pressed.

Post Conditions:

- APOD stop when button is released

Main Success Scenario:

- User click on [Turn Right] button ()

Alternative Scenario:

- Alt 1: press "D" key on keyboard (if Normal input is selected)
- Alt 2: press () key on gamepad (if Gamepad input is selected)

Exceptions:

- IO Exception

Relationships:

NONE

Table C-10 Turn right use-case specification

e) Body Lift

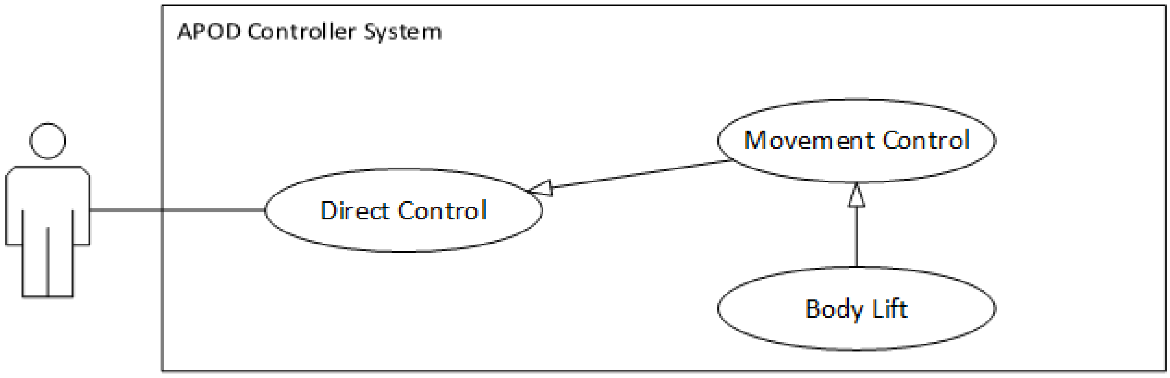


Figure C-12 Body Lift Use-case diagram

BODY LIFT USE CASE SPECIFICATION			
Use-case No.	UC011	**Use-case Version**	1.0
Use-case Name	Body Lift		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Command APOD to Lift up

Goal:

APOD Lift body up

Triggers

User click [R1] button

Preconditions:

- Live control feature is selected
- Bluetooth connection is established
- Normal input mode or Gamepad input mode is selected

Post Conditions:

NONE

Main Success Scenario:

- User click on [R1] button ()

Alternative Scenario:

- Alt 1: press "U" key on keyboard (if Normal input is selected)
- Alt 2: press () key on gamepad (if Gamepad input is selected)

Exceptions:

- IO Exception

Relationships:

NONE

Table C-11 Body Lift use-case specification

f) Body Drop

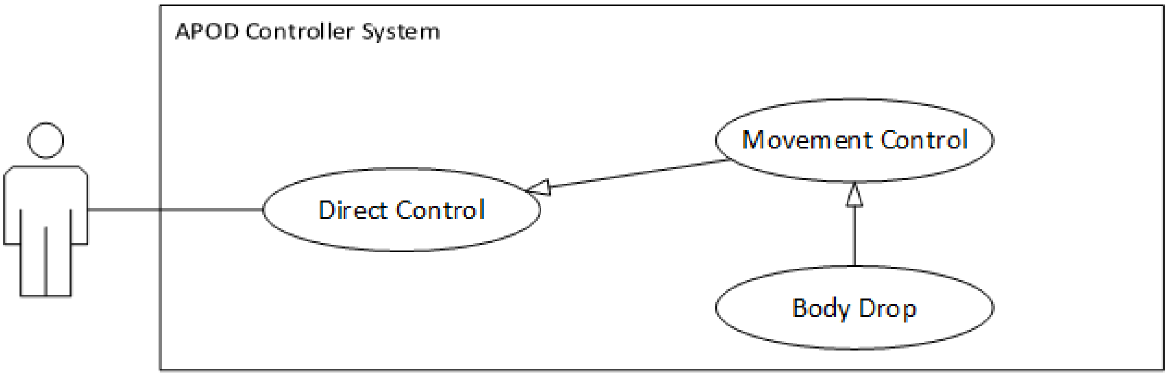


Figure C-13 Body Drop Use-case diagram

BODY DROP USE CASE SPECIFICATION			
Use-case No.	UC012	**Use-case Version**	1.0
Use-case Name	Body Drop		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Command APOD to Drop down

Goal:

APOD Drop body Down

Triggers

User click [R2] button

Preconditions:

- Live control feature is selected
- Bluetooth connection is established
- Normal input mode or Gamepad input mode is selected

Post Conditions:

NONE

Main Success Scenario:

- User click on [R1] button (R2)

Alternative Scenario:

- Alt 1: press “O” key on keyboard (if Normal input is selected)
- Alt 2: press (R2) key on gamepad (if Gamepad input is selected)

Exceptions:

- IO Exception

Relationships:

NONE

Table C-12 Body Drop use-case specification

g) Squeeze Left

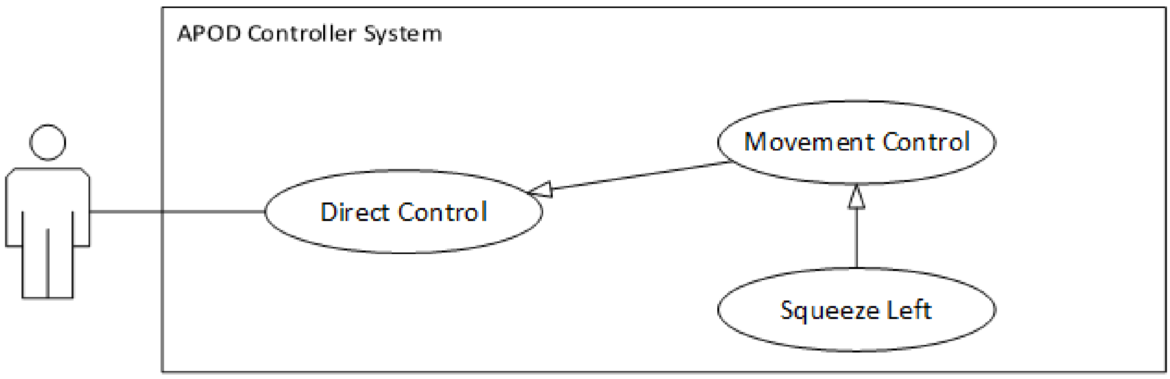


Figure C-14 Squeeze Left Use-case diagram

SQUEEZE LEFT USE CASE SPECIFICATION

Use-case No.	UC013	**Use-case Version**	1.0
Use-case Name	Squeeze Left		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Command APOD to squeeze to the left

Goal:

APOD Drop Squeeze left

Triggers

User click [Turn Left] button

Preconditions:

- Live control feature is selected
- Bluetooth connection is established
- Normal input mode or Gamepad input mode is selected
- [L2] button is pressed

Post Conditions:

NONE

Main Success Scenario:

- User press [L2] button (L2)
- User click on [Turn Left] button ()

Alternative Scenario:

- Alt 1: press "A" key on keyboard while pressing "E" key (if Normal input is selected)
- Alt 2: press () key on gamepad while pressing L2 (if Gamepad input is selected)

Exceptions: - IO Exception **Relationships:** NONE

Table C-13 Squeeze Left use-case specification

h) Squeeze Right

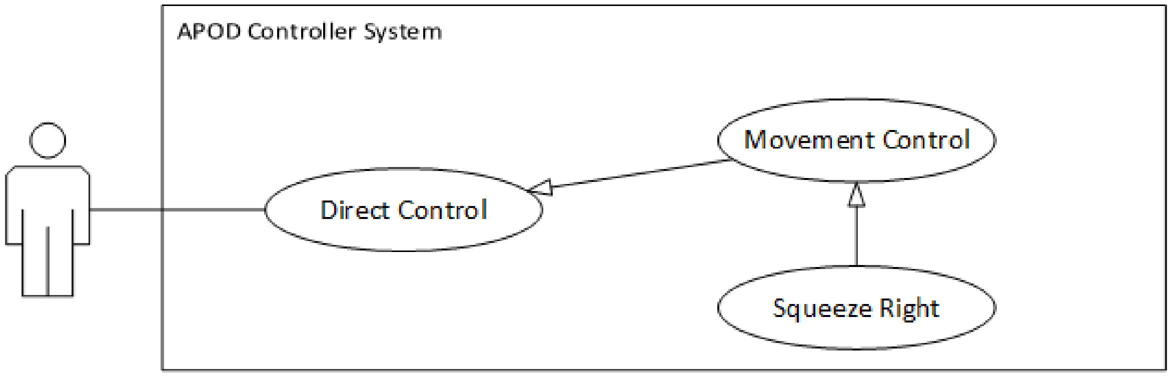


Figure C-15 Squeeze Right Use-case diagram

SQUEEZE RIGHT USE CASE SPECIFICATION			
Use-case No.	UC014	**Use-case Version**	1.0
Use-case Name	Squeeze Right		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal
Actor: User **Summary:** Command APOD to squeeze to the right **Goal:** APOD Drop Squeeze right **Triggers** User click [Turn Right] button			

Preconditions:

- Live control feature is selected
- Bluetooth connection is established
- Normal input mode or Gamepad input mode is selected
- [L2] button is pressed

Post Conditions:

NONE

Main Success Scenario:

- User press [L2] button (L2)
- User click on [Turn Left] button ( )

Alternative Scenario:

- Alt 1: press “D” key on keyboard while pressing “E” key (if Normal input is selected)
- Alt 2: press ( ) key on gamepad while pressing L2 (if Gamepad input is selected)

Exceptions:

- IO Exception

Relationships:

NONE

Table C-14 Squeeze Right use-case specification

i) Manual Grip/Release

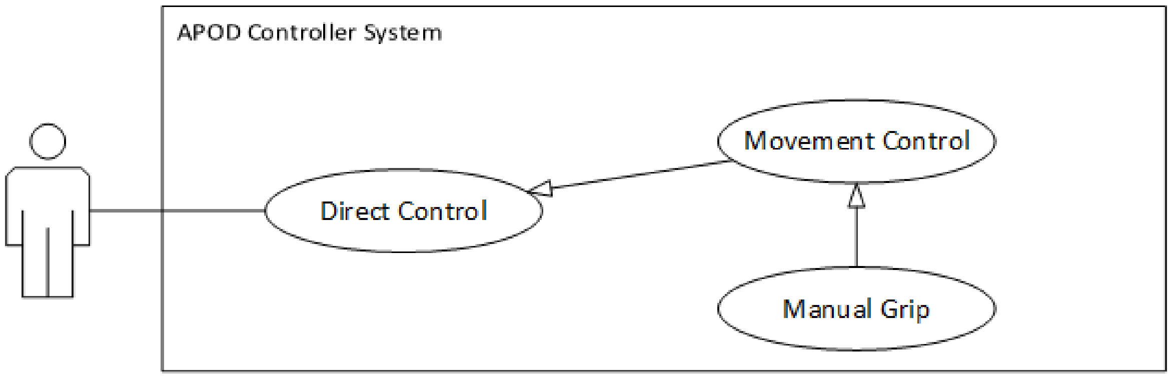


Figure C-16 Manual Grip Use-case diagram

MANUAL GRIP/RELEASE USE CASE SPECIFICATION

Use-case No.	UC015	**Use-case Version**	1.0
Use-case Name	Manual Grip		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Command APOD to Grip/Release Mandibles

Goal:

APOD Mandibles Grip/Release

<u>Triggers</u>

User click [Grip/Release] button

Preconditions:

- Live control feature is selected
- Bluetooth connection is established
- Normal input mode or Gamepad input mode is selected
- [L1] and [L2] is not pressed.

Post Conditions:

- APOD stop when button is released

Main Success Scenario:

- User click on [Grip] button (△) or [Release] Button (⊗)

Alternative Scenario:

- Alt 1: press "I" key on keyboard (if Normal input is selected)
- Alt 2: press (△) or (⊗) key on gamepad (if Gamepad input is selected)

Exceptions:

- IO Exception

Relationships:

NONE

Table C-15 Manual Grip use-case specification

2.3. Object tracking

2.3.1 Auto Grip

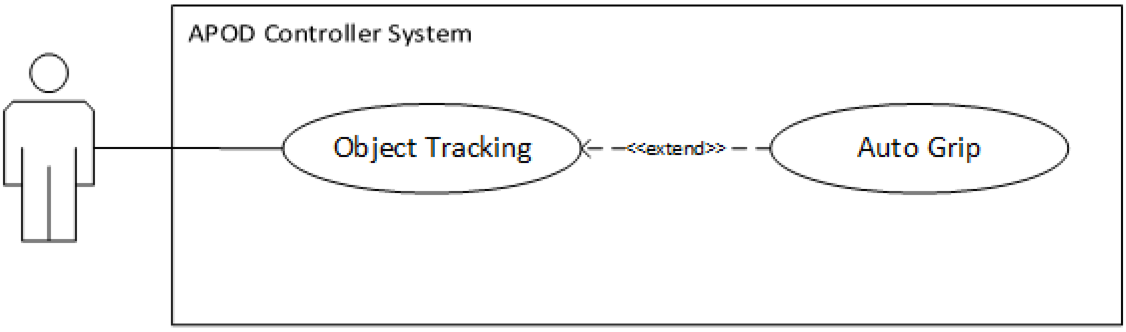


Figure C-17 Auto Grip Use-case diagram

AUTO GRIP USE CASE SPECIFICATION			
Use-case No.	UC016	**Use-case Version**	1.0
Use-case Name	Auto Grip		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal

Actor:

User

Summary:

Auto detect object in range and move forward to grab it.

Goal:

Object is grabbed

Triggers

User click [AUTO GRIP] button

Preconditions:

- Live control feature is selected
- Bluetooth connection is established

Post Conditions: NONE **Main Success Scenario:** - User click [AUTO GRIP] button - Auto grip command is sent to APOD **Alternative Scenario:** NONE **Exceptions:** - IO Exception **Relationships:** NONE

Table C-16 Auto Grip use-case specification

2.3.2 Choose Target

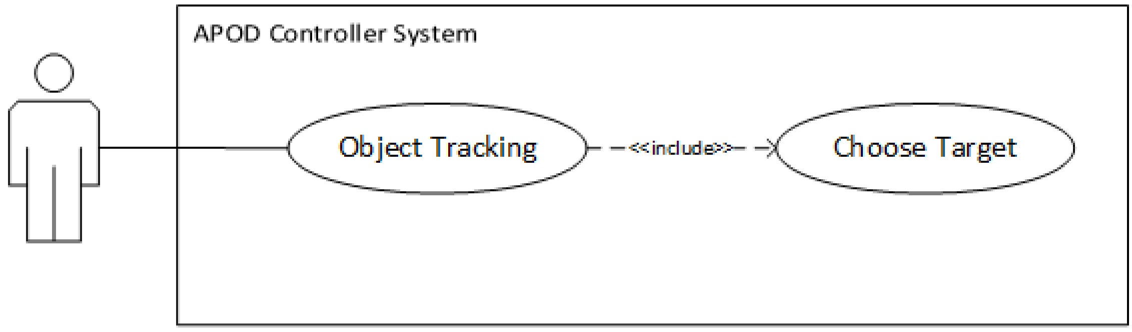


Figure C-18 Choose Target Use-case diagram

CHOOSE TARGET USE CASE SPECIFICATION			
Use-case No.	UC017	**Use-case Version**	1.0
Use-case Name	Choose Target		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal
Actor: User			

Summary:

Enter tracking mode, choosing target to track

Goal:

New tracking template is selected

Triggers

User click [Object Tracking] button

Preconditions:

- Configuration is initiated
- Bluetooth connection is established
- IP camera stream is working

Post Conditions:

NONE

Main Success Scenario:

- [Object Tracking] button is clicked
- Object extractor is showed
- User select tracking template from capture frame.
- User click [OK]

Alternative Scenario:

NONE

Exceptions:

- IO Exception
- Time Out Exception

Relationships:

NONE

Table C-17 Choose Target use-case specification

2.3.3 Change Target

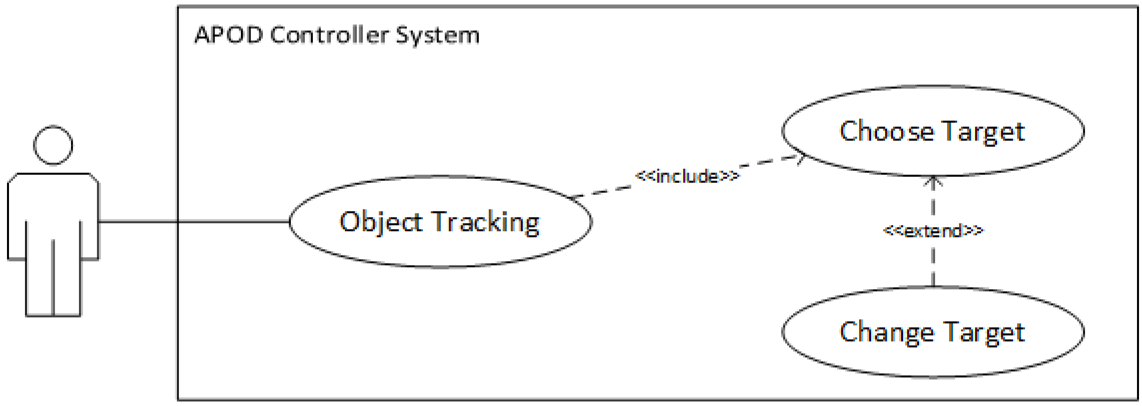


Figure C-19 Change Target Use-case diagram

<table>
<tr><td colspan="4">ADD POINT USE CASE SPECIFICATION</td></tr>
<tr><td>Use-case No.</td><td>UC018</td><td>Use-case Version</td><td>1.0</td></tr>
<tr><td>Use-case Name</td><td colspan="3">Change Target</td></tr>
<tr><td>Author</td><td colspan="3">CuongPAD</td></tr>
<tr><td>Date</td><td>05/10/2013</td><td>Priority</td><td>Normal</td></tr>
<tr><td colspan="4">Actor:
User
Summary:
Change current tracking template
Goal:
New tracking template is selected
<u>Triggers</u>
User click [Change target] button
Preconditions:
- Configuration is initiated
- Bluetooth connection is established
- IP camera stream is working
- Tracking mode is selected
Post Conditions:</td></tr>
</table>

NONE **Main Success Scenario:** - [Change Target] button is clicked - Object extractor is showed - User select tracking template from capture frame. - User click [OK] **Alternative Scenario:** NONE **Exceptions:** - IO Exception - Time Out Exception **Relationships:** NONE

Table C-18 Change Target use-case specification

2.3.4 Lock Target

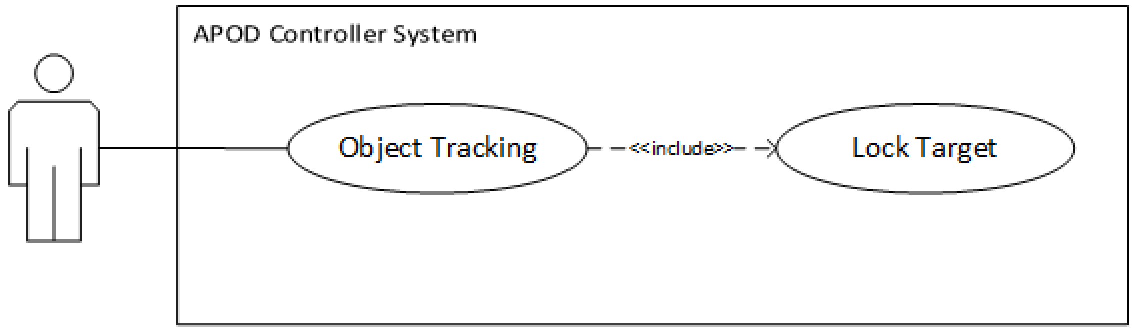


Figure C-20 Lock Target Use-case diagram

ADD POINT USE CASE SPECIFICATION			
Use-case No.	UC019	**Use-case Version**	1.0
Use-case Name	Lock Target		
Author	CuongPAD		
Date	05/10/2013	**Priority**	Normal
Actor:			

User

Summary:

Start tracking target

Goal:

Reach target (within 60 cm)

Triggers

User click [Lock] button

Preconditions:

- Configuration is initiated
- Bluetooth connection is established
- IP camera stream is working
- Tracking template was set

Post Conditions:

NONE

Main Success Scenario:

- User click on [Lock] button
- APOD keep moving forward (with adjustment) until target is reached
- User decide to Grab object or not.

Alternative Scenario:

NONE

Exceptions:

- IO Exception

Relationships:

NONE

Table C-19 Lock Target use-case specification

IV. SOFTWARE SYSTEM ATTRIBUTES

1. Reliability

- Easy to upgrade firmware.
- The APOD 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.

2. Availability

- In case of electrical incident, the APOD system will be shut down and reset automatically.
- Controller has the flexibility that allows changes in hardware design.

3. Maintainability

- Each module of the APOD controller system can be easily upgrade.
- APOD embedded controller can be recycled and integrated into other control system.

1. Security

- N/A

2. Portability

- The APOD controller is depend on hardware so that APOD system do not have the portability attribute. However, the APOD 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.

3. Performance

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

D. SOFTWARE DESIGN DESCRIPTION (SDD)

I. DESIGN OVERVIEW

1. Introduction

This section is created to introduce and give a brief overview of the system design.

The followings information are given in this section:

- System Architecture Design : the system overall structure
- Components Description : detailed info of hardware components
- Class Diagram : entity & attributes
- Sequence Diagram : describe the flow of events
- User Interface Design

2. Architecture

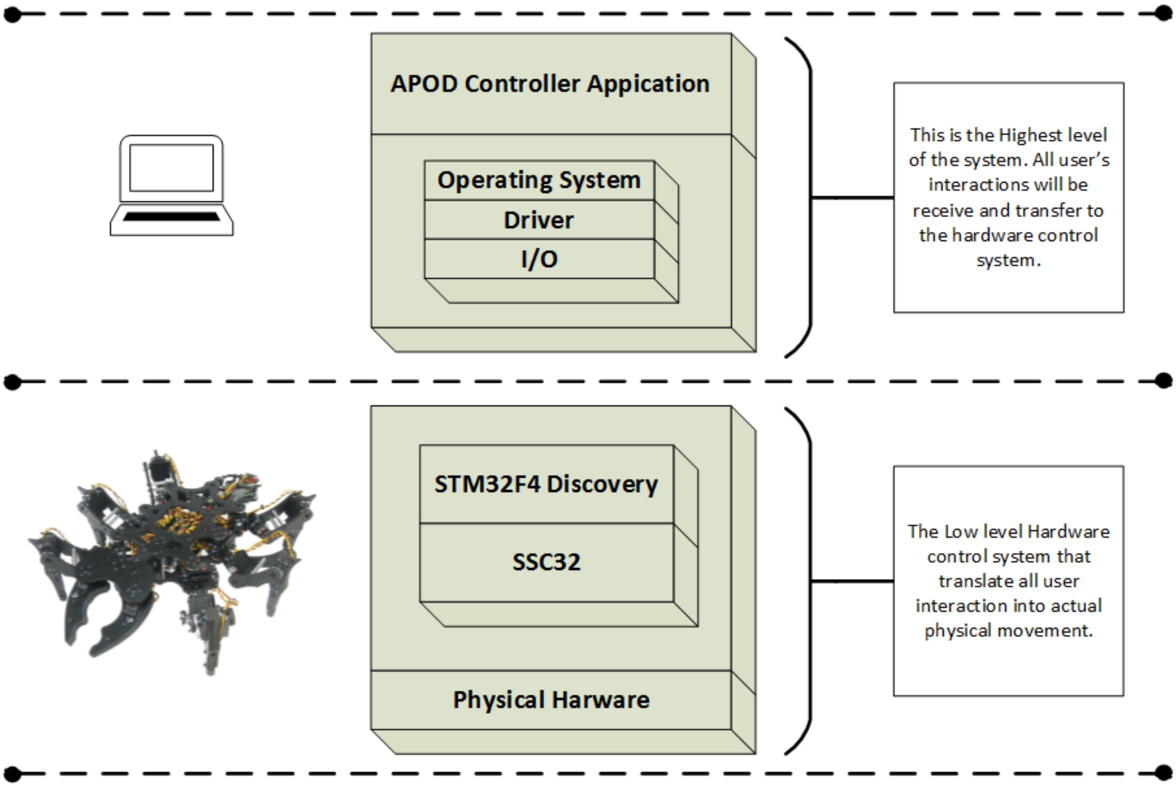


Figure D-1 APOD Controller Architecture

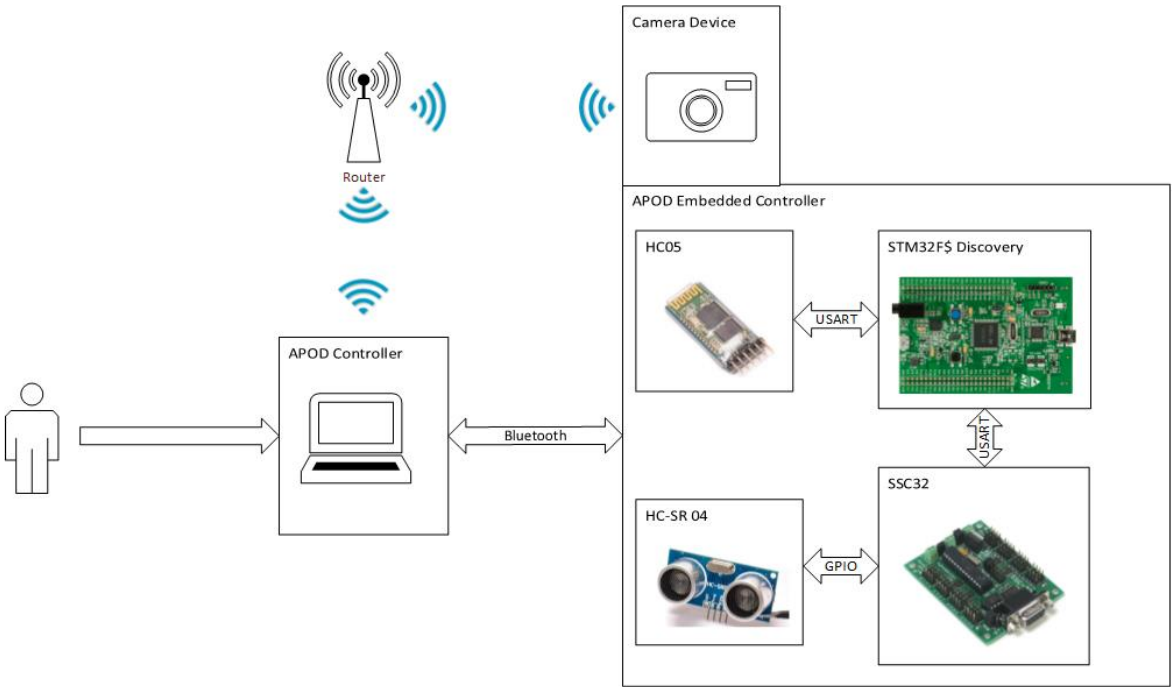


Figure D-2 APOD Communication Mechanism

3. User interface

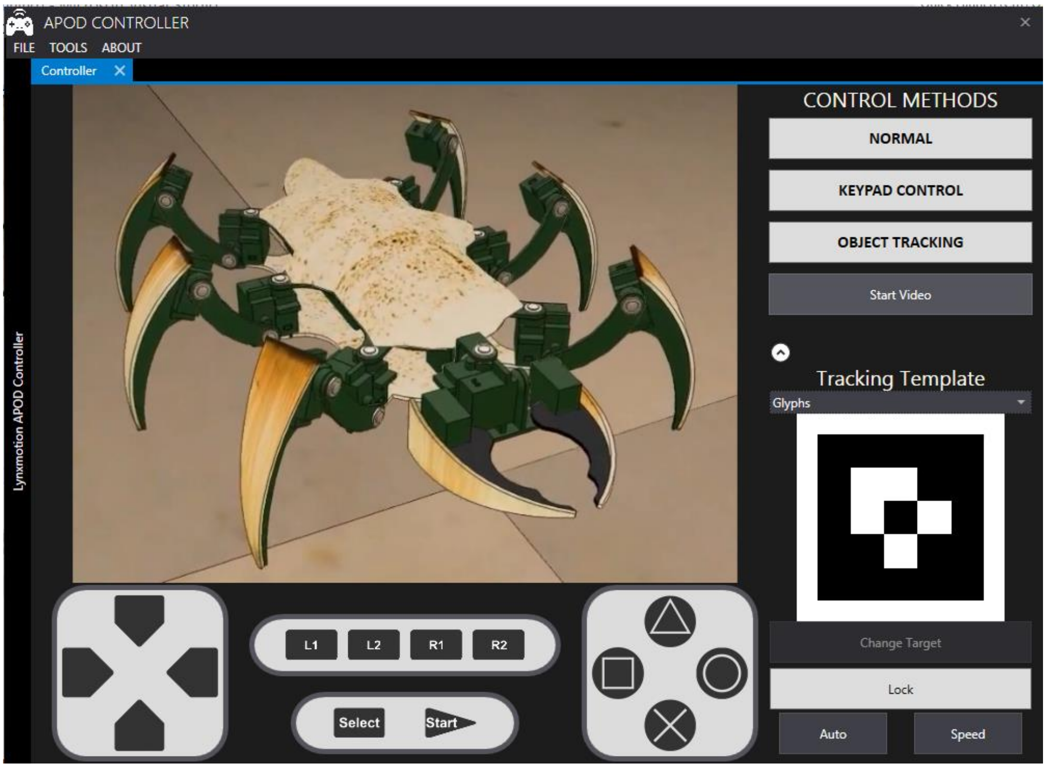


Figure D-3 Live Control User interface design

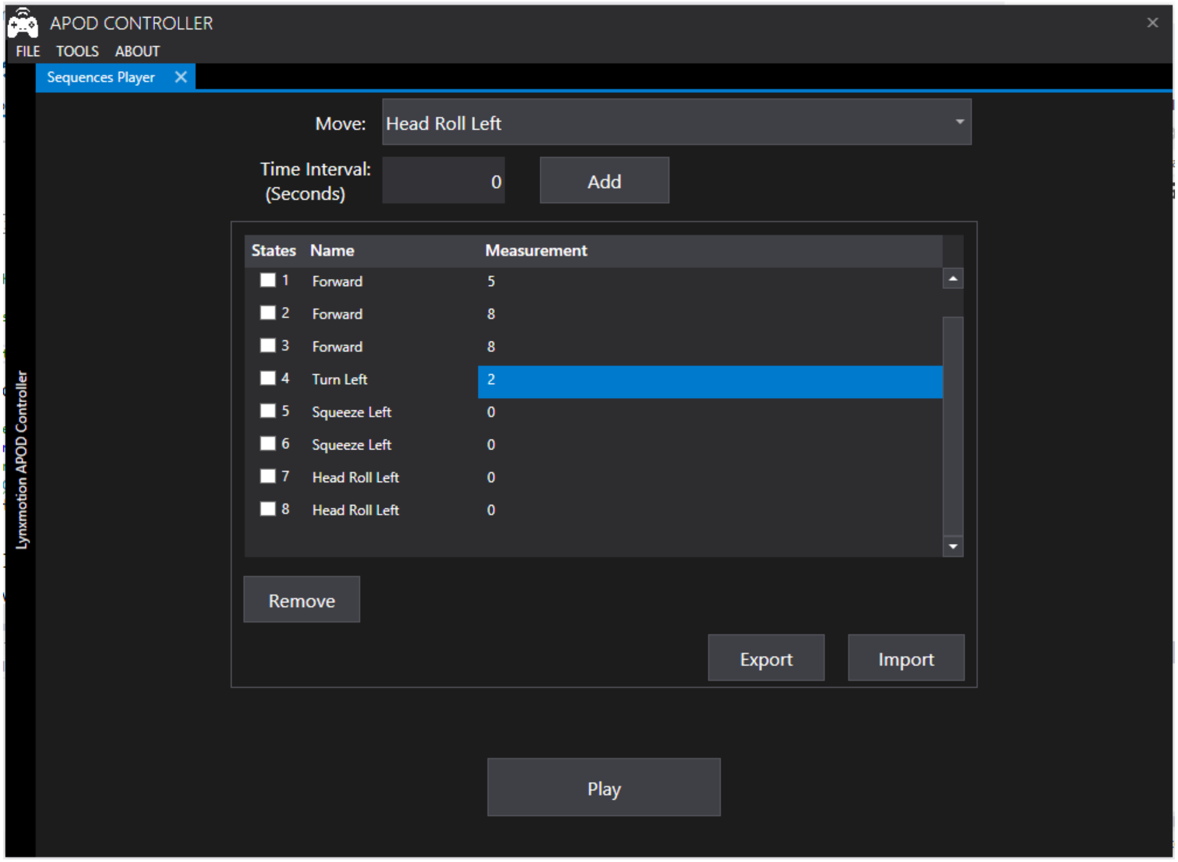


Figure D-4 Sequence Player user interface design

II. COMPONENTS

1. Central processing

1.1. Bluetooth module (HC 05)

Communication module (using Bluetooth) to transmit and receive commands from Controller devices (PC)

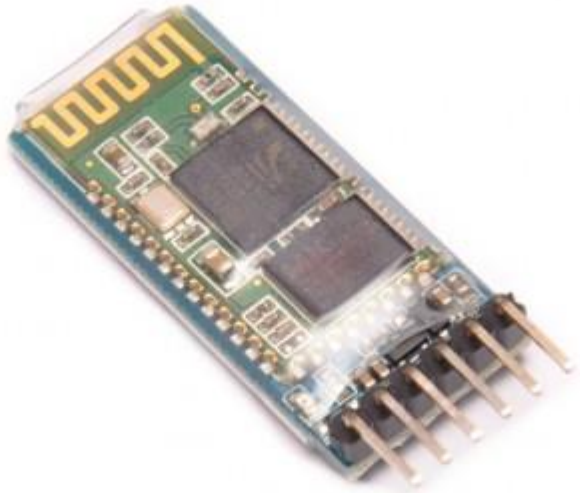


Figure D-5 Bluetooth Communication module HC05

1.2. STM32F4 Discovery

The Center Processing module to control all APOD movements or reaction to a specific event (command or obstacle).

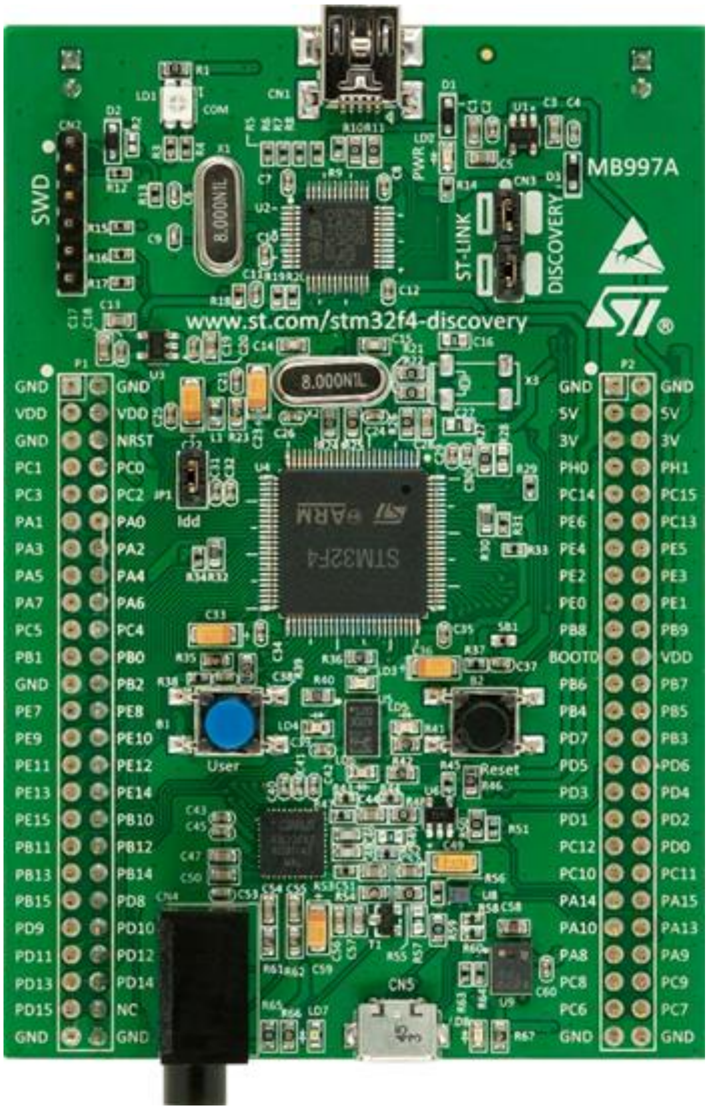


Figure D-6 STM32F4 Discovery Board

2. Servos control & Sensor reader

2.1. HG-645MG Servo

- Metal gear servo that provided the base of APOD movement.
- The details of how to control will be discussed later in this document.

Figure D-7 HC 645MG Servo

2.2. SSC32 with ATMEGA328P

- A servo controller : 32 PWM outputs to control 32 servos (25 servos in used for this project)
- Distance sensor reader: HC-SR04 (described in the section below)
- For further information about this board, followings this link of the manufacturer : http://www.lynxmotion.com/p-395-ssc-32-servo-controller.aspx

Figure D-8 SSC32 Board

2.3. Ultrasonic ranging module (HC SR04)

- Ultrasonic module using for distance measurement, discovering obstacles.

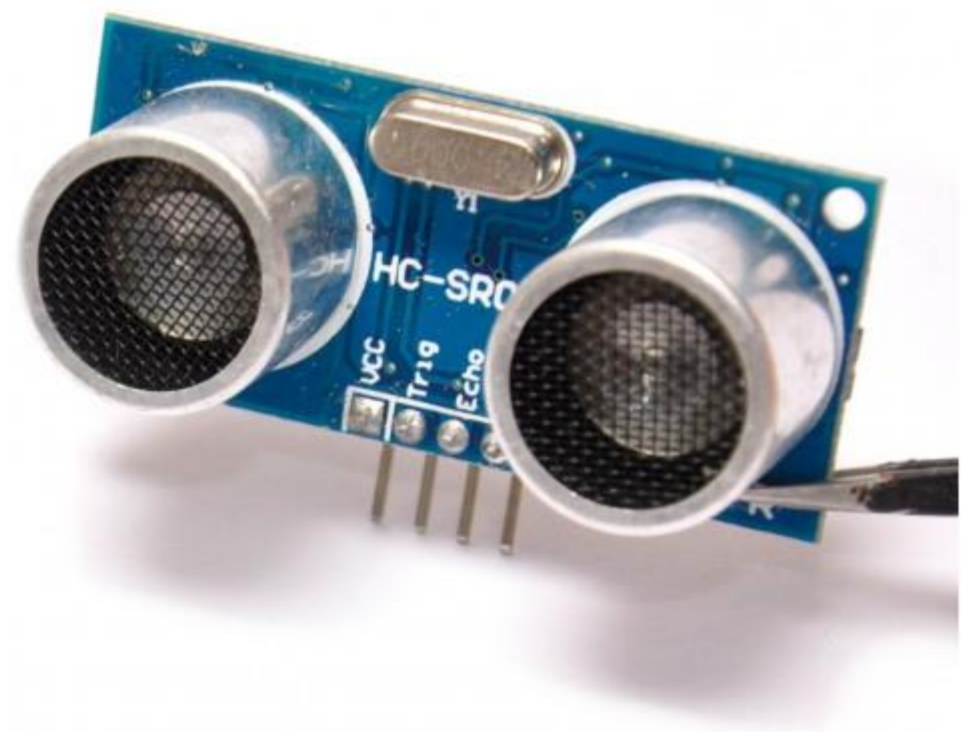


Figure D-9 HCSR04 Distance Sensor

III. IMPLEMENTATION APPROACHES

1. Algorithms and Mechanism

1.1. RC Servo Control

1.1.1 HG-645MG Specification

Motor Type:	3 Pole
Bearing Type:	Dual Ball Bearing
Speed (4.8V/6.0V):	0.24 / 0.20
Torque oz./in. (4.8V/6.0V):	107 / 133
Torque kg./cm. (4.8V/6.0V):	8.0 / 10.0
Size in Inches:	1.59 x 0.77 x 1.48
Size in Millimeters:	40.39 x 19.56 x 37.59
Weight ounces:	1.94
Weight grams:	55

Table D-1 HG-645MG Specifications

1.1.2 How does it work?

Unlike the AC/DC motor, Servo motor do not rotate continuously. Servo motor rotation range usually fall within 90° to 180° (some special servo can rotate up to 360°) and keep holding the torque at the position as long as its power supply remain or until it receive signal to move the torque.

Most of Servos have 3 wires connector. One wire for power supply (4 – 6v). The second wire is for voltage ground. And the third wire is for receiving control signal.

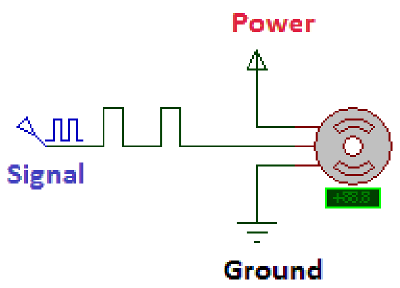


Figure D-10 RC Servo block

The whole idea to control servo is sending the appropriate signal to servo so that it will move to the intended position. The signal to control servo, basically is a PWM (Pulse Width Modulation) signal of 50 Hz (or refresh rate: 20ms). It is the width of carrier pulse that decide the position of the torque (not the number of pulse).

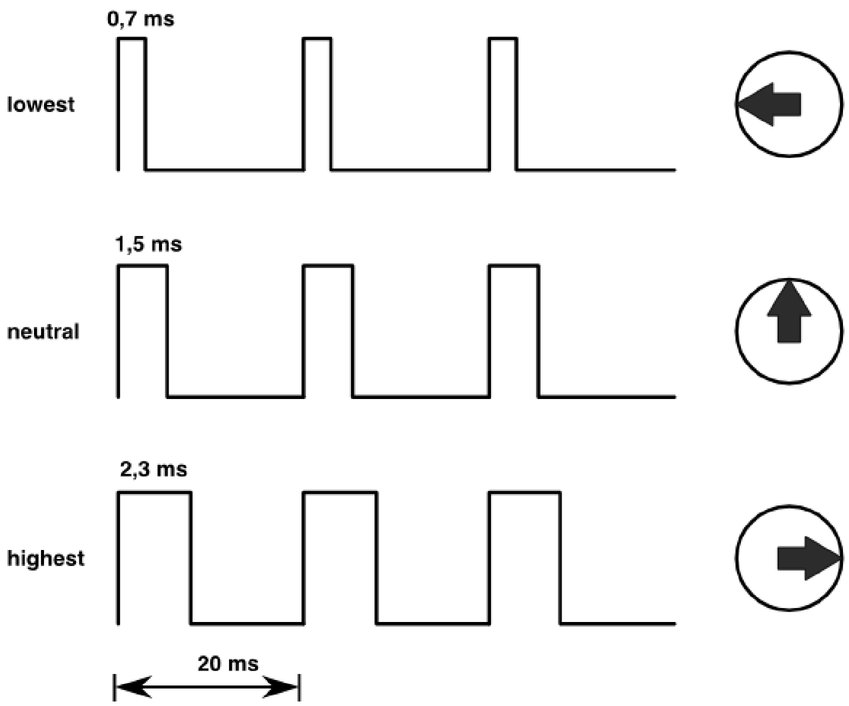


Figure D-11 Servo Signal illustration

In order for the servo to keep the position, the signal must be maintain.

In short words, the pulse width of carrier signal will act as a coordinate value of the servo's torque. Once the coordinate is changed, the servo constantly move the toque to the responding position discarding of its current position. The rotating speed will be depending on the manufacturer of the servo.

For the HG-645MG servo, the corresponding signal is showed below:

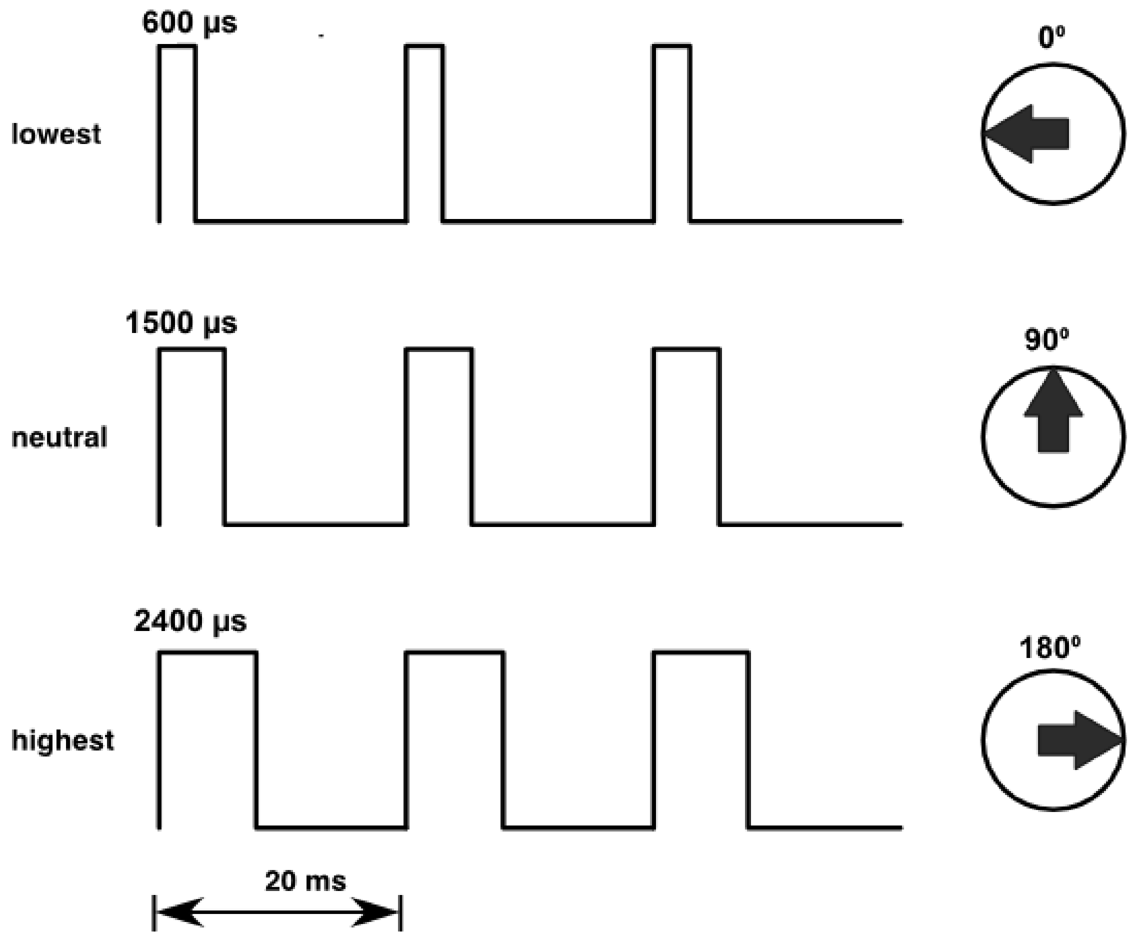


Figure D-12 HC 645MG Signal Details

1.1.3 Servo's Speed Control

As we know, the servo will always rotate at a constant speed, for example the servo HG-645MG speed is 60^o/ 0.2s with 6V supply. With such speed applied to robot's movement, it can affect the stability of the structure, especially for those in which mechanical part is not so much reliable. How do we slow it down?

The idea is not making the servo rotate a whole distance at once but make it rotate one step at a time until it reach its destination, in short it is "divide and conquer" method. For example, you want your HG-645MG servo to rotate 90^o in 1 second and the current position of your servo is 0^o. If you just apply the PWM with 1500 µs pulse width (1500 µs stand for 90^o position), the servo will move to its destination in only 0.3 second.

With the method mentioned above, we divide the whole 90° into ten 9° pieces (or 150 µs step as we call it). The signal pulse width will be modified 10 times, after each time the width will be added 150 µs more and waiting 100 ms until the next change happen. With this implementation, the servo will move 10x9° = 90° in 10x100 = 1000 ms = 1 s. The larger the divisor is, the smoother the servo will move.

For this project, the step is always fix at 20 µs and the waiting time fall in range of 10 ms to 20 ms depend on user purposes.

Let D be the travelling distance (in µs), T be the expected time for the move, δ be the waiting time for each step. We have the speed formula:

$$\delta = \frac{T \times 20}{D}$$

1.2. Sensor Reader (HC-SR04)

1.2.1 HC-SR04 Specifications

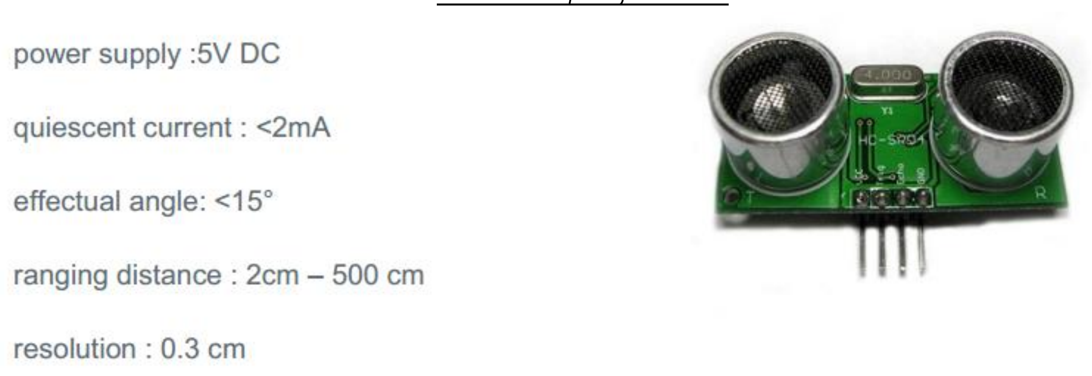


Figure D-13 HC-SR04 Specifications

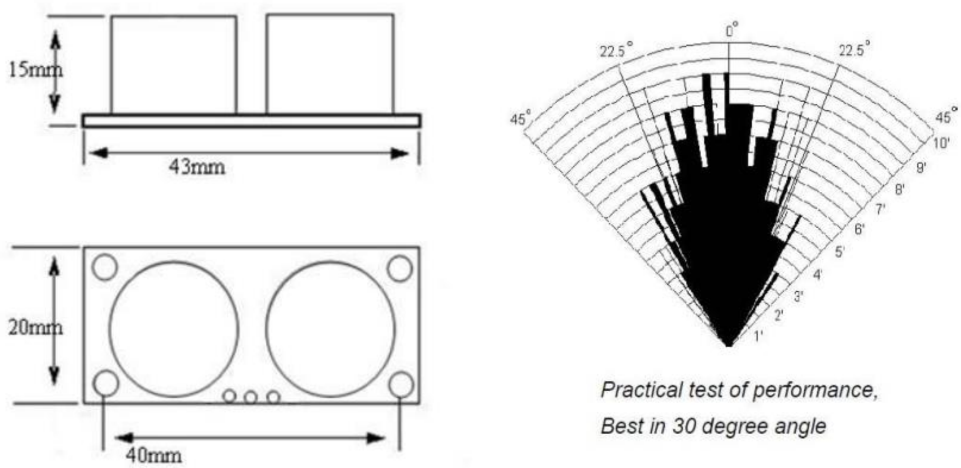


Figure D-14 HC-SR04 Best practice angle

1.2.2 How to read distance value?

The HC-SR04 has 4 pins: VCC, GND, Trigger, Echo.

- VCC is used for power supply
- GND is used for voltage ground
- Trigger and Echo will perform as follow

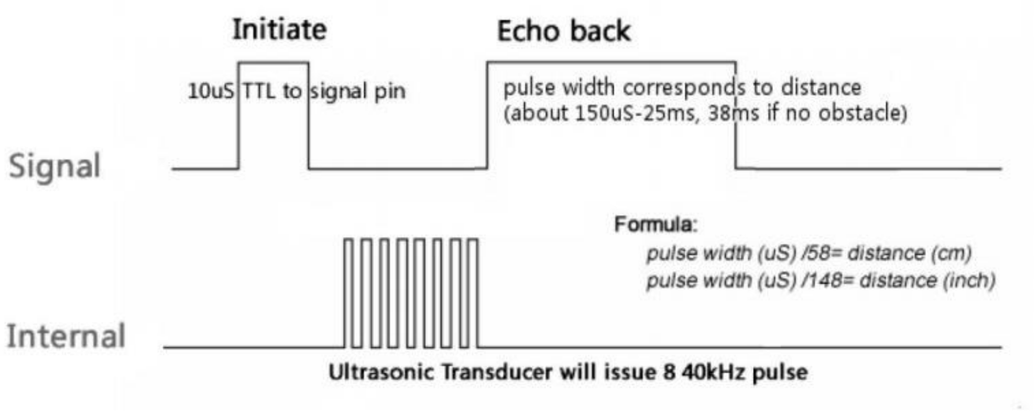


Figure D-15 HC-SR04 Operating sequence

First the Trigger pin is pull high for more than 10 µs (10 µs pulse), then waiting for the Echo output to go high and start timing the length of the Echo pin staying high. At this point, a short ultrasonic pulse is transmitted, waiting to be reflected by an obstacle. When the reflected ultrasonic pulse is receive by senor, the Echo output will go low.

The time that Echo output staying high will be used to calculate the distance by following formula given by the manufacturer:

$$distance\ (cm) = \frac{pulse\ width\ (\mu s)}{58}$$

Or

$$distance\ (inch) = \frac{pulse\ width\ (\mu s)}{148}$$

1.3. Multi-legs Control

The description below is described in the following article by Oricom Technologies – "Analysis of Multi-Legged Animal + Robot Gaits" (2003).

1.3.1 Tripod Gait

The tripod gait is the best-known hexapod gait. Hexapod six legs is divide into 2 halves for to control. Each half (or tripod) consists of 3 legs: the front and rear leg of one side and the center leg of the opposite side. More specific:

- Tripod A: left front, right center and left rear
- Tripod B: right front, left center and right rear.

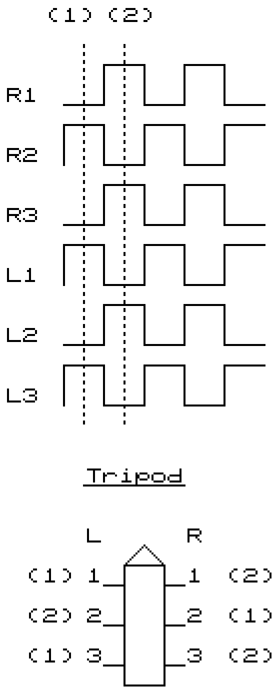


Figure D-16 Tripod Gait illustration

During walking cycle, the weight of the entire hexapod is shifted from one tripod to another. At a moment, there is always one tripod on the ground while another doing its cycle. "*Since 3 legs are on the ground at all times, this gait is both "statically" and "dynamically" stable*." - (Oricom Technologies , 2003)

1.3.2 Wave Gait

The wave gait implementation make each leg in one side do it cycle after one another leg has finished its cycles and repeat it on the other side's legs.

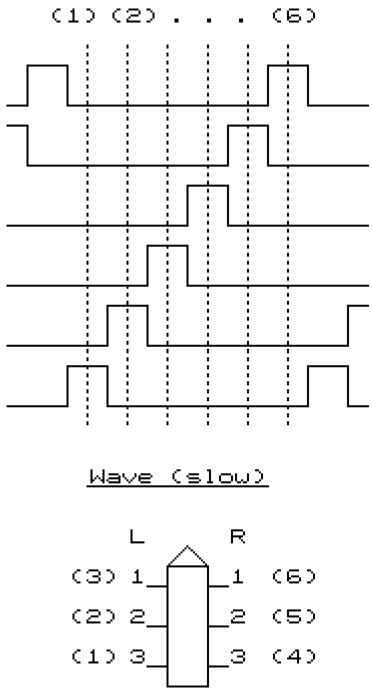


Figure D-17 Wave Gait illustration

For instance, the turn of leg taking step is as follow: Left Front -> Left Center -> Left Rear -> Right Front -> Right Center -> Right Rear

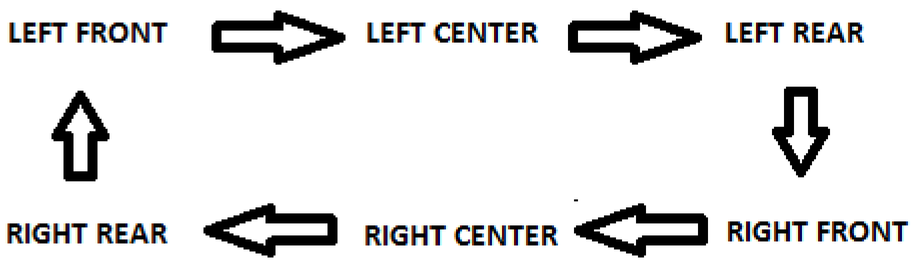


Figure D-18 Wave Gait step order

Since there is always 5 legs on the ground, this method is the most stable gait but in return, it is also the slowest gait due to its "*one step at a time*".

1.3.3 Ripple Gait

"The final stride is the Ripple Gait. At first glance the timing of this gait looks somewhat complicated, however, the key to understanding is to recognize that, on each side a local wave comprising non-overlapping lift phases is being performed, and that the 2 opposite side waves are exactly 180 degrees out of phase with one another. For instance, if L3 and R3 are considered to represent the start of each local wave, then notice that R3 starts to move exactly in the middle of the L3-L2-L1 side wave" ." (Oricom Technologies , 2003)

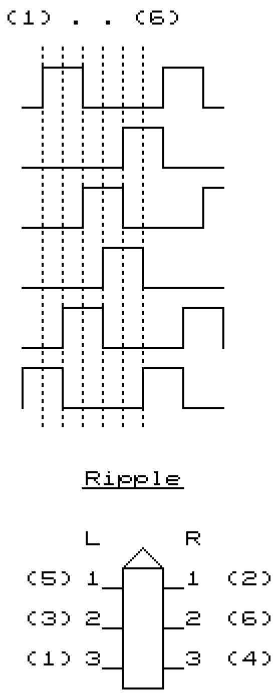


Figure D-19 Ripple Gait illustration

1.3.4 Algorithms

For this project, we will apply **the tripod gait** method to control the APOD walking movement. As we know, 6 legs of APOD will be divided into 2 tripods

We defined 2 tripod as follow:

- **Tripod A: Left front, Right Center, Left Rear**
- **Tripod B: Right front, Left Center, Right Rear**

a) Walking In Straight Line

Leg's Position:

There are 6 basic positions for the APOD's leg divive into 2 orientation.

- Vertical: High, Middle , Low
- Horizontal: Front, Center, Rear

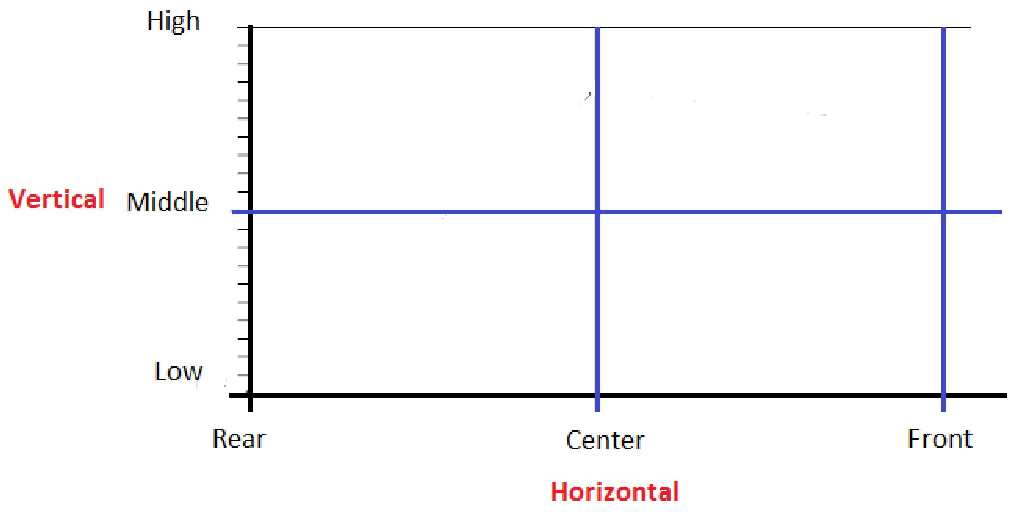


Figure D-20 APOD leg position block

Each step of the APOD leg is divide into 2 halves: swing time (legs is in the air) and standing time (leg is on the ground). The swing time (or swing) is represented by a sequence of 4 consecutive states defined by the position of the leg in both team of vertical and horizontal positions.

For example, this is the forward sequence:

State	Vertical	Horizontal
0	Low to Mid	Rear
1	Mid to High	Rear to Center
2	High to Mid	Center to Front
3	Mid to Low	Front

Table D-2 Leg swing sequence

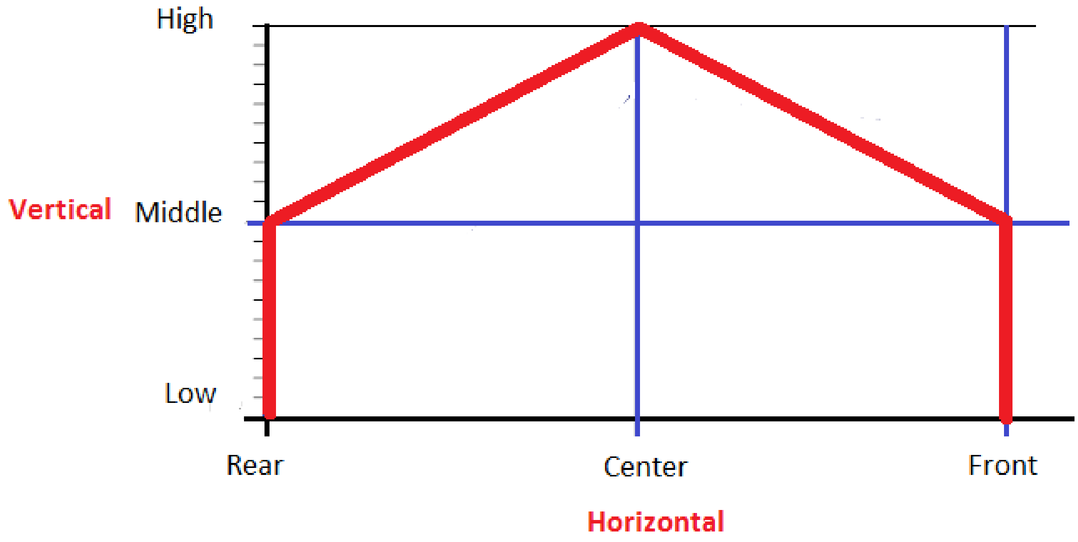


Figure D-21 APOD leg swing diagram

Depend on the step direction (forward or backward) the sequence may differ: “Front” and “Rear” may be replaing each other’s position.

After making the swing, the leg will return to its original position (compare to the body) while keeping the feet on the ground (stand time). This is the point the force was born to push the body moving in the opposite direction of the moving legs.

From the APOD view point, the whole operations of moving leg can be seen as the figure below:

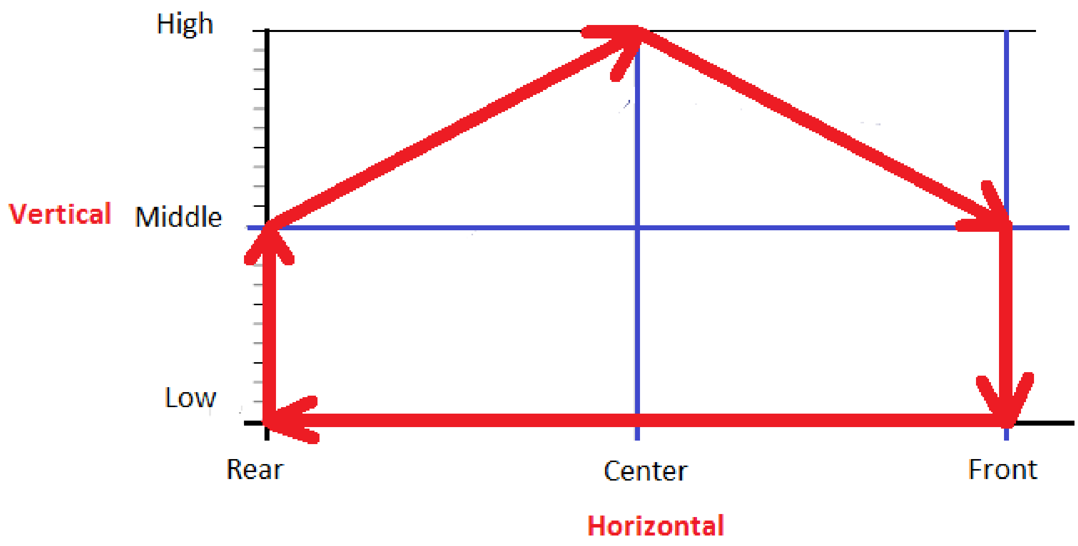


Figure D-22 APOD Leg full step diagram

The whole operation of APOD's leg can be seen as this sequence of states (Forward step) looping times to times:

State	Vertical	Horizontal
0	Low	Front to Center
1	Low	Center to Rear
2	Low	Rear
3	Low to Mid	Rear
4	Mid to High	Rear to Center
5	High to Mid	Center to Front
6	Mid to Low	Front
7	Low	Front

Table D-3 APOD Leg full forward step

Tripod Position

According to the nature of The Tripod Gait, every 3 legs in one tripod will always move and stop at the same time. In other words, the cycles of 3 legs in 1 one tripod are exactly the same phase with one another. The term tripod position is defined by the combination of its legs positions.

Let's take a view of Tripod A (Left Front, Left Rear and Right Center) from the up top viewpoint for example:

- Position: Front (close to head)

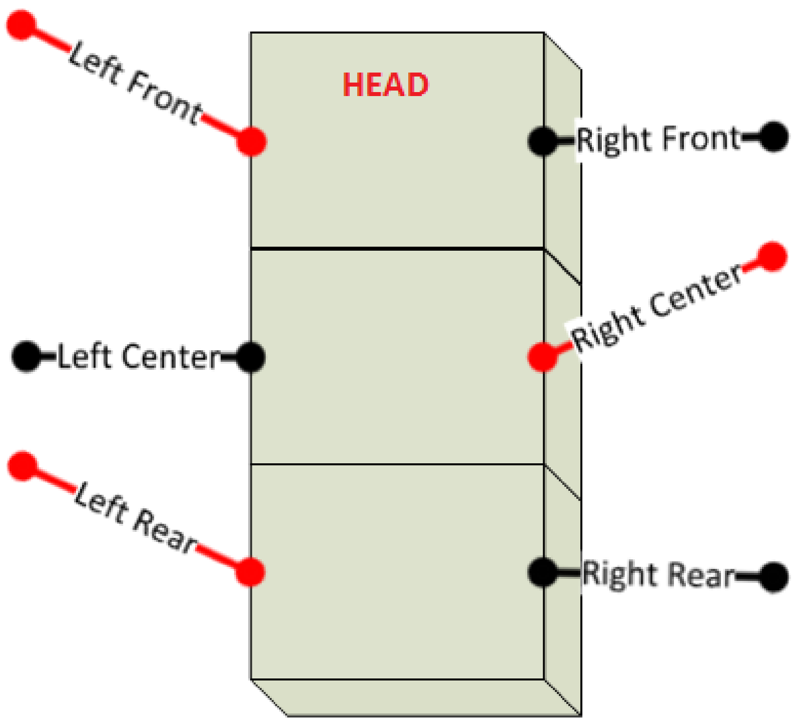


Figure D-23 Tripod A at Front position

- Position: Center or Neutral

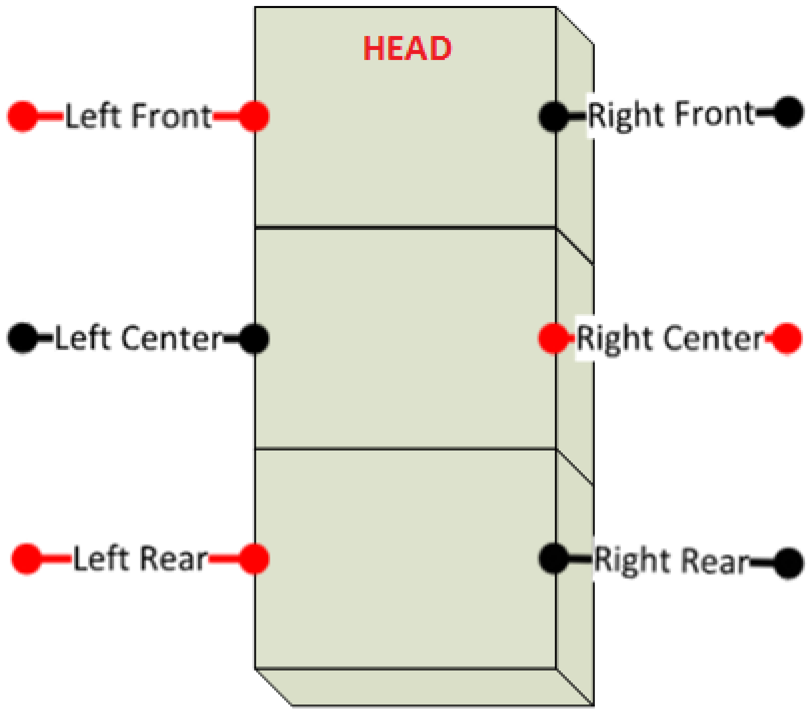


Figure D-24 Tripod A at Neutral position

- Position: Rear (far from head)

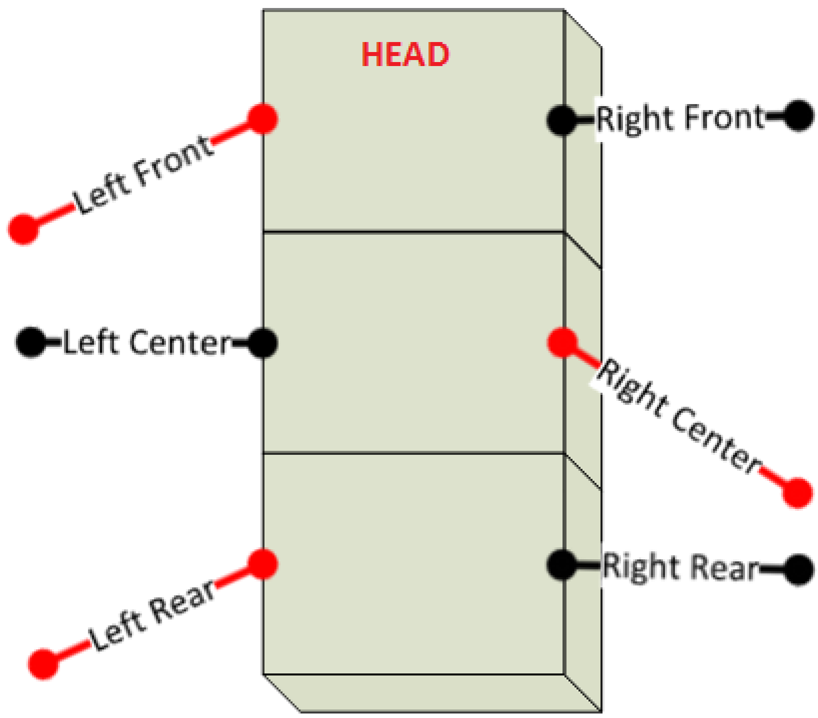


Figure D-25 Tripod A at Rear position

Since all Tripod A's Legs making the forward step at once (moving from front to rear), there will be 2 pushing force from both side of the body, from the back to the head, that making the APOD moving forward. This is the fundamental of making the APOD moving forward. The backward move can be achieve by making the reverse step of forward step (the backward step).

Walking

Now before we can figure out the walking method of APOD, let's examine the human walking method first. As we can see, the position of one leg in walking (or running) is always the opposite position of the other one. Or in short, their cycles is exactly 180^o out of phase with one another. The similar mechanism is applied for The Tripod Gait walking method since there are 2 tripods, which act like left and right legs in human case. Let Tripod A and B stand for 2 legs of human, then their cycles will always be 180^o out of phase with the other one. For conclusion, the full sequence for 2 tripods of The Tripod Gait walking method is described as below

Forward Sequence

State	Tripod A		Tripod B	
	Vertical	**Horizontal**	**Vertical**	**Horizontal**
0	Low	Front to Center	Mid to High	Rear to Center
1	Low	Center to Rear	High to Mid	Center to Front
2	Low	Rear	Mid to Low	Front
3	Low to Mid	Rear	Low	Front
4	Mid to High	Rear to Center	Low	Front to Center
5	High to Mid	Center to Front	Low	Center to Rear
6	Mid to Low	Front	Low	Rear
7	Low	Front	Low to Mid	Rear

Table D-4 Forward Sequence

Backward Sequence

State	Tripod A		Tripod B	
	Vertical	**Horizontal**	**Vertical**	**Horizontal**
0	Low	Rear to Center	Mid to High	Front to Center
1	Low	Center to Front	High to Mid	Center to Rear
2	Low	Front	Mid to Low	Rear
3	Low to Mid	Front	Low	Rear
4	Mid to High	Front to Center	Low	Rear to Center
5	High to Mid	Center to Rear	Low	Center to Front
6	Mid to Low	Rear	Low	Front
7	Low	Rear	Low to Mid	Front

Table D-5 Backward Sequence

b) <u>Turning Left – Right</u>

As mention above, we already know 3 basic position of tripod: Front, Center, Rear. In this section, there are 2 more positions: Left and Right.

The definition of 2 new position can be described by tripod A as follow:

- Position: Left

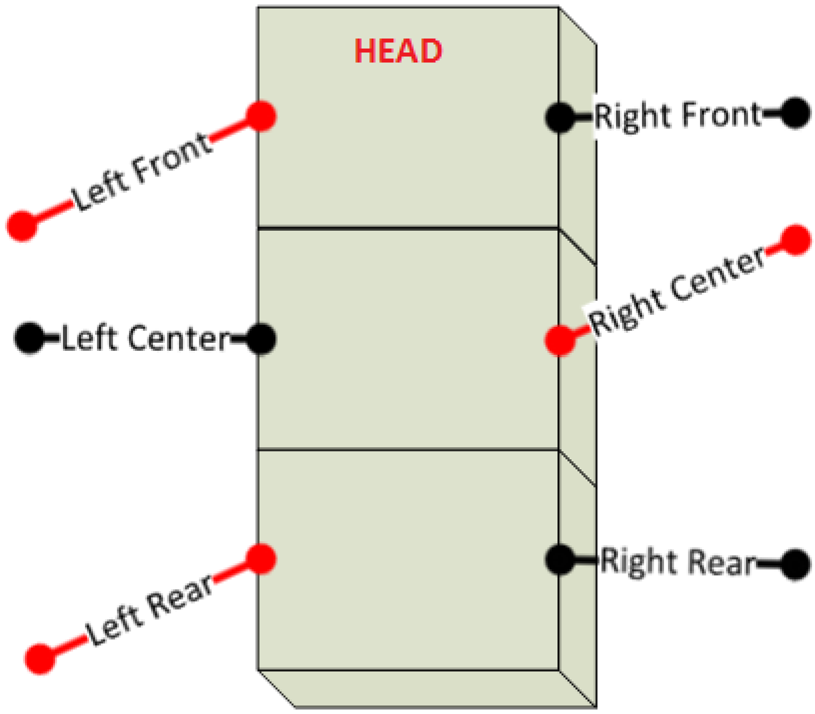


Figure D-26 Tripod A at Left position

- Position: Right

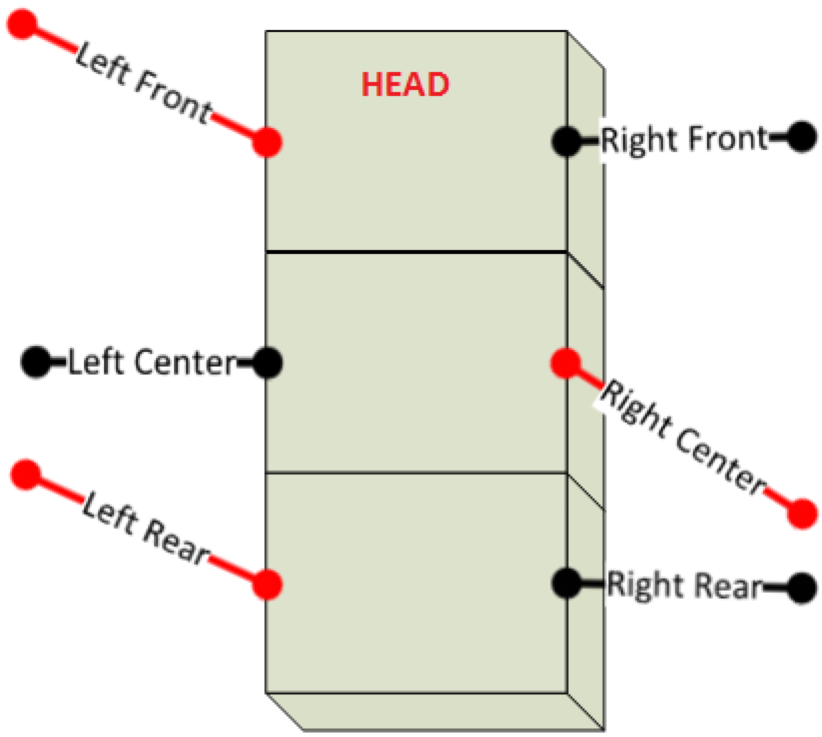


Figure D-27 Tripod A at Right position

Let's take a look when Tripod A making set :

- Left Front and Left Rear take the forward step
- Right Center take the backward step

There will be forces with 2 opposite direction on the APOD body. The left side force push the body forward and the right side force push the body backward. As a result of this combination. The APOD will turn from left to right. In other words, when the tripod making transition from left position to right position (through center position), the APOD will turn from left to right. And when the tripod making transition from right position to left position (through center position), the APOD will turn from right to left.

The combination sequence for turning the APOD:

Turn Left:

State	Tripod A		Tripod B	
	Vertical	Horizontal	Vertical	Horizontal
0	Low	Left to Center	Mid to High	Right to Center
1	Low	Center to Right	High to Mid	Center to Left
2	Low	Right	Mid to Low	Left
3	Low to Mid	Right	Low	Left
4	Mid to High	Right to Center	Low	Left to Center
5	High to Mid	Center to Left	Low	Center to Right
6	Mid to Low	Left	Low	Right
7	Low	Left	Low to Mid	Right

Table D-6 Turn Left Sequence

Turn Right:

State	Tripod A		Tripod B	
	Vertical	Horizontal	Vertical	Horizontal
0	Low	Right to Center	Mid to High	Left to Center
1	Low	Center to Left	High to Mid	Center to Right
2	Low	Left	Mid to Low	Right
3	Low to Mid	Left	Low	Right
4	Mid to High	Left to Center	Low	Right to Center
5	High to Mid	Center to Right	Low	Center to Left
6	Mid to Low	Right	Low	Left
7	Low	Right	Low to Mid	Left

Table D-7 Turn Right Sequence

2. SSC32 Board

2.1. Schematic

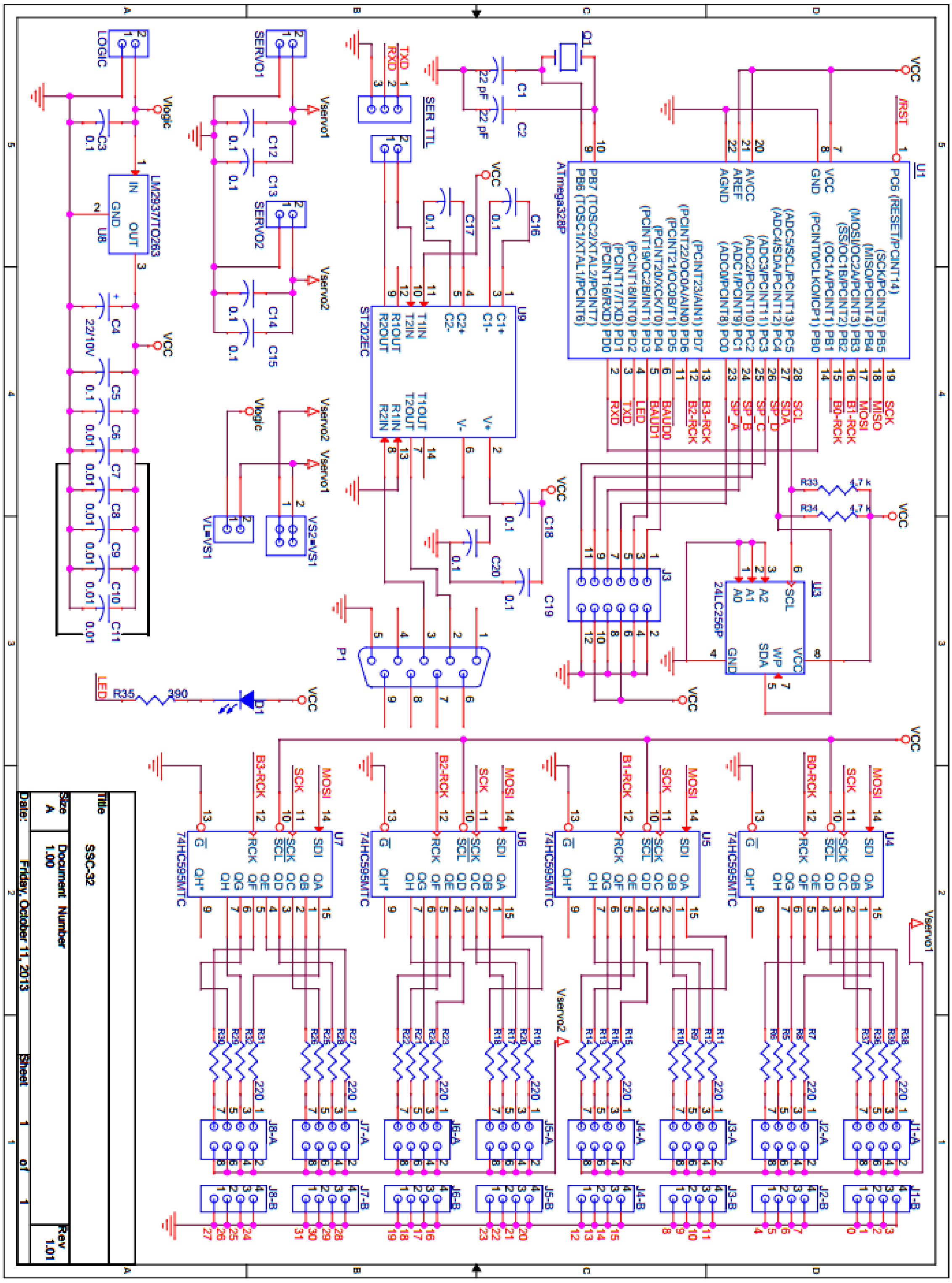


Figure D-28 SSC32 Schematic

2.2. Technical Point

2.2.1 Atmega328P Microcontroller (MCU)

a) Overview

ATmega328p is a high-performance Atmel picoPower 8-bit AVR RISC-based (Reduced Instruction Set Computing) microcontroller provided with these features:

- 32KB ISP (In-System Programming) flash memory with read-while-write capabilities
- 1024B EEPROM (Electrically Erasable Programmable Read-Only Memory)
- 2KB SRAM (Static Random-Access Memory)
- 23 general purpose I/O lines
- 32 general purpose working registers
- Three flexible timer/counters with compare modes
- Internal and external interrupts
- Serial programmable USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter)
- A byte-oriented 2-wire serial interface
- SPI (Serial Peripheral Interface) serial port
- A 6-channel 10-bit A/D
- Programmable watchdog timer with internal oscillator
- Five software selectable power saving modes

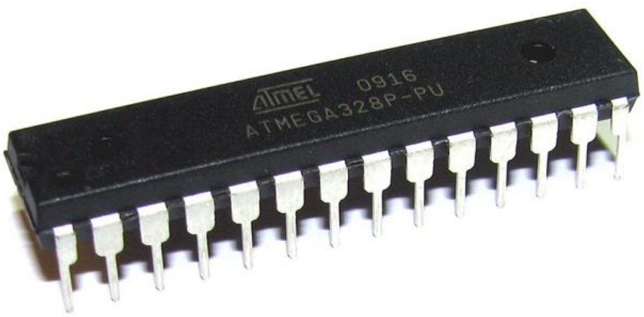


Figure D-29 ATmega328p Microcontroller

For more information about ATmega328 Features and Specifications, please refer to the ATmega328p Datasheet from Atmel web site:

http://www.atmel.com/devices/atmega328p.aspx?tab=documents

b) Pin Configuration

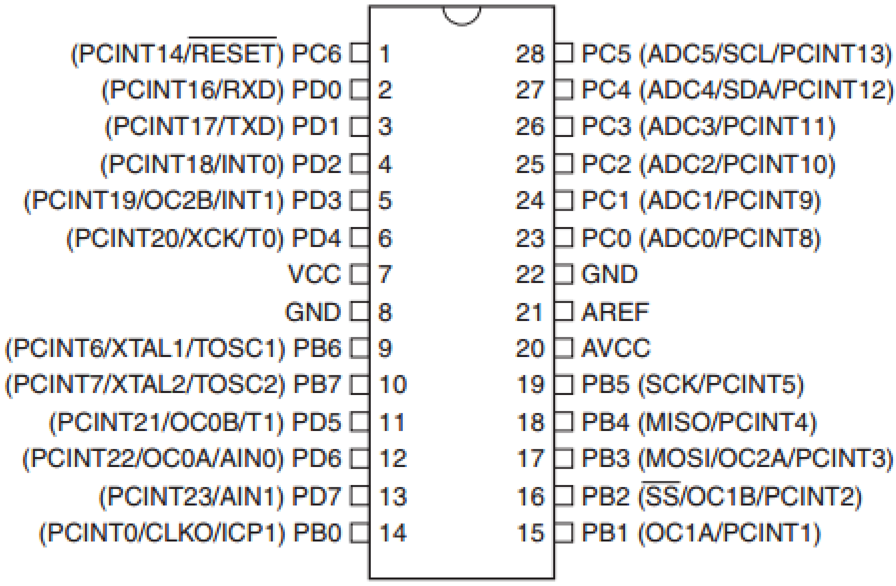


Figure D-30 ATmega328p 28 Pins DIP

c) Block Diagram

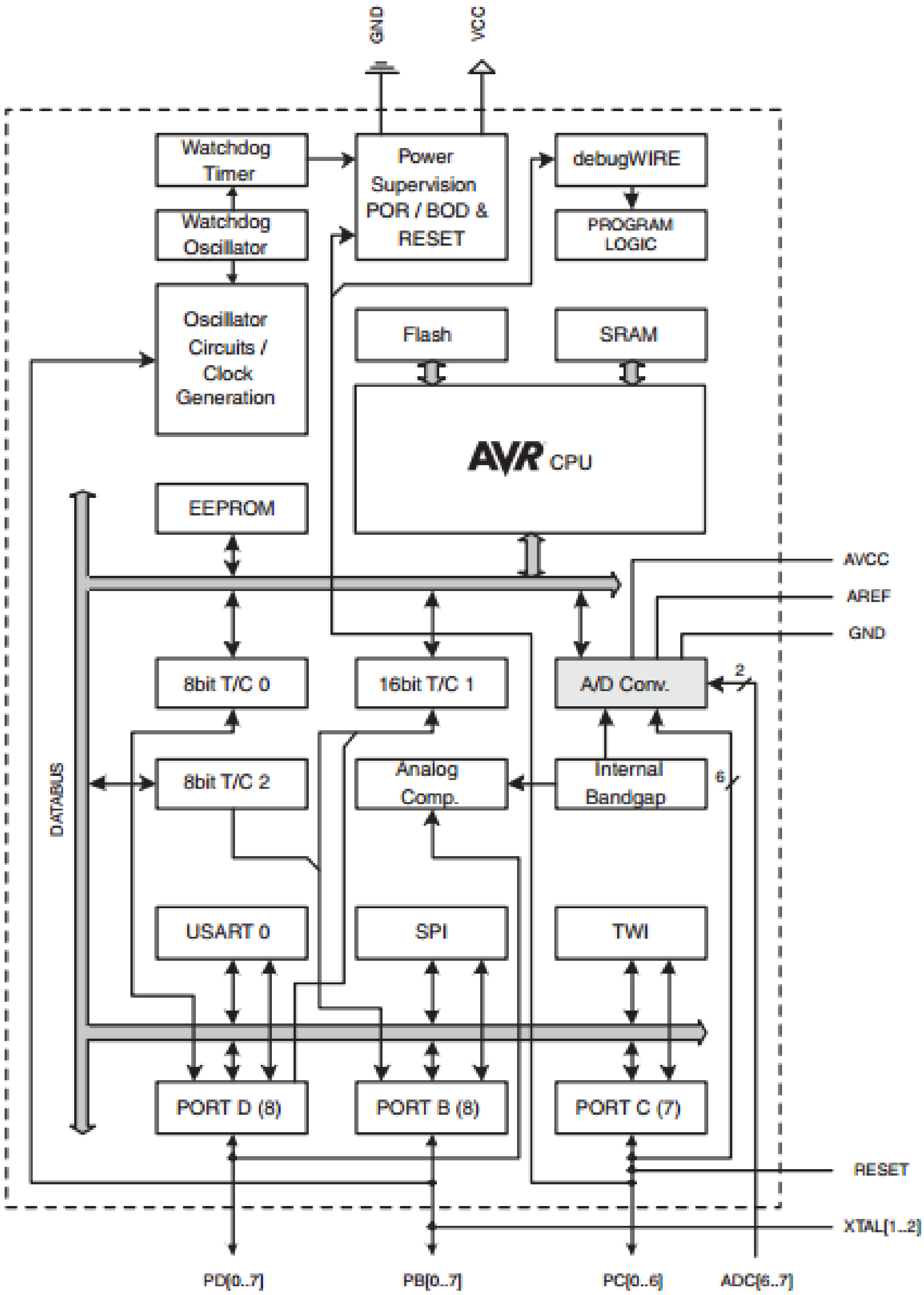


Figure D-31 ATmega328p Block Diagram

2.2.2 General Purposes Input / Output (GPIO)

a) Overview

The ATmega328p has 3 Ports: Port B, Port C, Port D and each of them consists of 8 pin PB[0..7], PC[0..7], PD[0..7]

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. The DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address. The details will be described later.

Notes:

- "x" represent the port name, can be "B","C" or "D".
- "n" represent the pin number of the x port, vary from 0 to 7 (each port has 8 pins, as mentioned above)

b) Configurations

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn.

2.2.3 Interrupts

The ATmega328p provides 26 interrupt vectors for user. In the scope of this project, we only use some of the interrupts services listed in the table below:

Number	**Source**	**Definition**
1	PCINT1	Pin Change Interrupt Request 1
2	TIMER1 COMPA	Timer/Counter1 Compare Match A
3	TIMER1 COMPB	Timer/Coutner1 Compare Match B
4	TIMER0 COMPA	Timer/Counter0 Compare Match A
5	USART, RX	USART Rx Complete
6	USART, TX	USART, Tx Complete

Table D-8 Interrupts Usage

2.2.4 Timer

The main feature of the SSC32 board is to control servos, more specific: 32 HC645MG servos. As mention in previous section, a RC servo is controlled by a signal, which basically is a PWM signal or simply an output pin toggled at specific time (high <-> low). It is the responsibility of Timer making the toggle to generate the PWM signal controlling servos. To support that purpose of this project, we will use 16-bit Timer of the ATmega328p (TIMER/COUNTER 1).

2.2.5 *Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART)*

a) Usage and Purposes

In this project, the use of USART will be the main communication channel between SSC32 module and other modules. The command will be receive through this channel as well as the response from SSC32 (when required).

b) Baud Rate

The SSC32 support method for user to change baud rate of USART via 2 jumper on board: baud0 and baud1

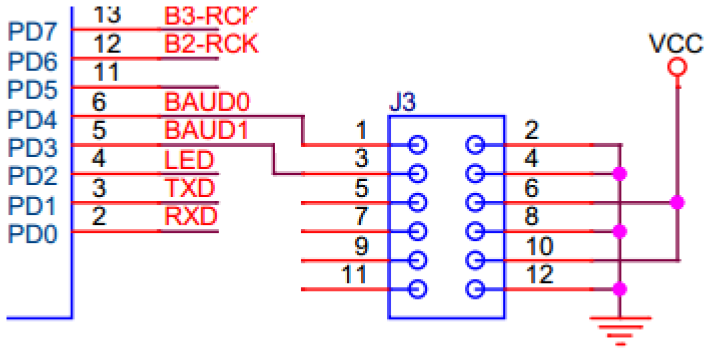


Figure D-32 Baud Rate setup

BAUD0	BAUD1	Baud Rate	UBRR0H	UBRR0L
0	0	2400	1	7F
0	1	9600	0	5F
1	0	38400	0	17
1	1	115200	0	7
Note: **- 1: indicate jumper is close** **- 0: indicate jumper open** **- The details about UBRR0H and UBRR1H can be found in Atmega328p Datasheet**				

Table D-9 Baud Rate setup details

After apply the jumper(s) on baud rate selection pin(s), the ATmega328p must be reset for the new baud rate to be applied.

2.2.6 *Serial Peripheral Interface (SPI)*

In this project, SPI communication will only be used for communication between ATmega328P and 74HC595, not for communication between microcontrollers. The detail of SPI- Shift Register interface will be discussed later in this document.

2.2.7 Shift Register (74HC595)

a) Overview

74HC595 is a shift register device with 8-bit serial input 8-bit serial or parallel output. This project will use the "serial to parallel conversion" feature of this Shift register.

b) How does it work?

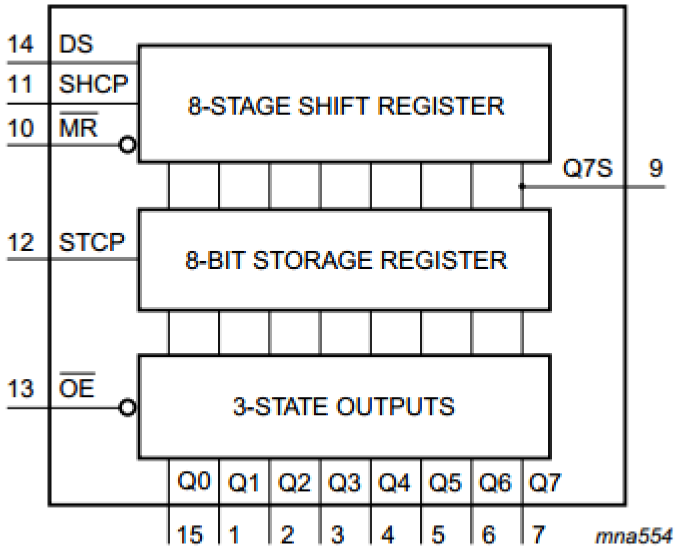


Figure D-33 74HC595 Block diagram

- **Serial Data Input (DS):** This pin is connected to any port of the microcontroller. This pin is responsible for getting the data serially from the microcontroller.
- **Shift Clock (SH_CP):** On this pin the clock signal is applied. On the rising edge, that is when the clock rises from negative to positive, the data on the DS line is sampled and it gets stored in the shift register. The bit on DS line is stored on the LSB (least significant place i.e. BIT0). On next pulse this bit moves to BIT1 location. After 8 clock pulses this bit is moved to BIT7 (MSB place). After 8 clock pulses the Shift Register has all 8 bits of a byte and is ready to convert them to parallel.
- **Store Clock (ST_CP):** You can hold this pin LOW while you get everything setup and nothing on the display pins will change. Then when you are done, and everything is how you want, you pull the pin HIGH and the 74HC595 will display the new settings. So even though we are changing values in the register in 8 steps, it looks like it was just one step
- **MR :** – Will empty the whole shift register, if pulled low, must be pulled high to enable.
- **OE:** – This pin enables the output when tied to ground, & disabled when HIGH.

2.2.8 *SPI - Shift Register interface*

a) Overview

Let's take a look of how we generate the signal controlling servos using timer. We have 32 servos to be control by the SSC32 Board.

First approach, we will use Output Compare feature of timer to generate the PWM signal. With this built-in feature, we can generate highly accurate PWM signal. But the problem is that ATmega328p have only 2 output compare pin for each timer. Even if we use all 3 timers, we can only control 6 servos by this approach.

Second approach, we will use the GPIO feature to output the PWM signal. The PWM will be toggle manually for each pins in ports. But on total, ATmega328 has only 3 GPIO ports which mean we can only control 8x3 = 24 servos while having 32 servos to control (or at least 25 servos for this particular project) and also wasting all the other features of the MCU.

The third approach is using some external device. In this case, we will use 4 more Shift Registers (74HC595) to output 32 PWM signal (8 pin on each Shift Register).

b) How does it work?

Take a look on SPI diagram

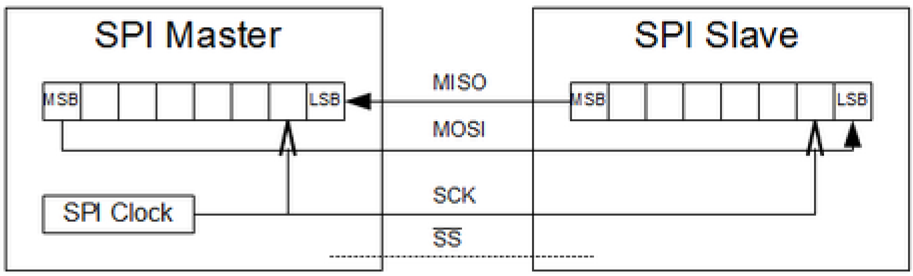


Figure D-34 SPI Diagram

We know that after the SPI Data register is written the SPI transmitting protocol will be trigger to send out data.

Data in the SPI Data register will be shifted out one bit at a time on the SCK clock, start with the MSB (Most Significant Bit or Most Left Bit). At one clock of SCK, one bit of data will be shift out on MOSI (Master Out Slaver In) pin.

Now we wired the 74HC595 with SPI output pins as follow:

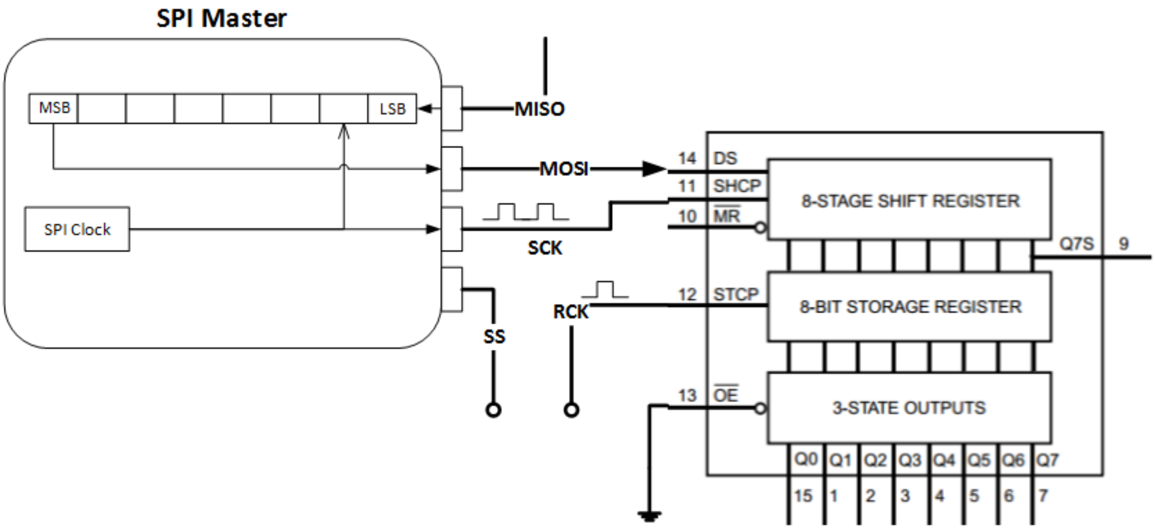


Figure D-35 SPI- Shift Register

Ignore the MISO (Master In Slaver Out) pin, which will be read as "0", the MOSI pin is now connect to the DS pin of the Shift register. SCK will be connected to the SH_CP Pin of the Shift register (or also SCK).

When the SPI transmitting is trigger, the SPI clock will also be the Shifting clock on the Shift Register. Each bit of the SPI data will follow the clock transfer into the 74HC595 Register through the MOSI – SDI wire. As a result, the SPI become a tool for writing data to shift register automatically. The output of 74HC595 can only change when a high pulse is apply to the ST_CP Pin (or the RCK as the Figure above).

Using the same MISO and SCK pin for 4 Shifter, 4 more pin for the output trigger. We have extended 6 pins of ATmega328p into 32 pins, which is enough for controlling 32 servos as mentioned above, while not wasting other features of the ATmega328p.

3. STM32F4 Discovery Board

3.1. I/O Schematic

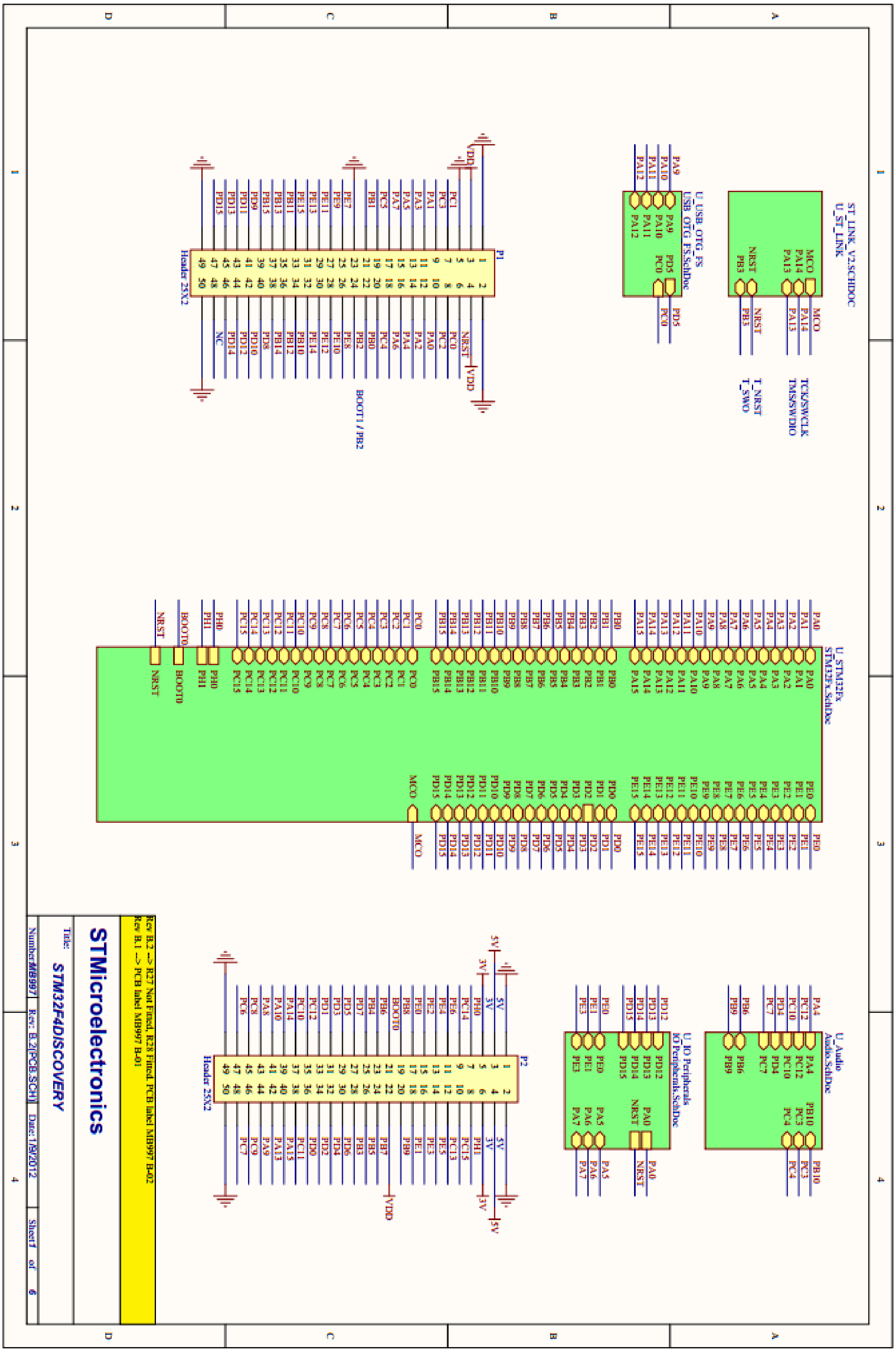


Figure D-36 STM32F4 I/O Schematic

In the scope of this project, we only use 4 pin:

- PA2, PA3 as Transmit and Receive using USART2 to serve the communicating purpose with SSC-32 Board and other.
- PB6, PB7 as Transmit and Receive using USART1 to communicate with Bluetooth HC05.

Table of Pin function (*) shown below:

STM32F40x pin and ball definitions (continued)

Pin number						Pin name (function after reset)(1)	Pin type	I / O structure	Notes	Alternate functions	Additional functions
LQFP64	WLCSP90	LQFP100	LQFP144	UFBGA176	LQFP176						
16	J10	25	36	P2	42	PA2	I/O	FT	(4)	USART2_TX/TIM5_CH3 / TIM9_CH1 / TIM2_CH3 / ETH_MDIO/ EVENTOUT	ADC123_IN2
17	H9	26	37	R2	47	PA3	I/O	FT	(4)	USART2_RX/TIM5_CH4 / TIM9_CH2 / TIM2_CH4 / OTG_HS_ULPI_D0 / ETH_MII_COL/ EVENTOUT	ADC123_IN3
58	C7	92	136	B6	164	PB6	I/O	FT		I2C1_SCL/ TIM4_CH1 / CAN2_TX / DCMI_D5/USART1_TX/ EVENTOUT	
59	B7	93	137	B5	165	PB7	I/O	FT		I2C1_SDA / FSMC_NL / DCMI_VSYNC / USART1_RX/ TIM4_CH2/ EVENTOUT	

Figure D-37 Pin Function

Note: (*) For more detail, refer to STM32F4 Datasheet.

3.2. STM32F407VGT6 Microcontroller (MCU)

3.2.1 Overview

This ARM Cortex-M4 32-bit MCU with FPU (Float-Point Unit) has 210 DMIPS (Dhrystone Microprocessor without Interlocked Pipeline Stages), up to 1 MB Flash/192+4 KB RAM (Random Access Memory), USB OTG HS/FS (Universal serial bus on-the-go full-speed/half speed), Ethernet, 17 timers, 3 ADCs, 15 comm. interfaces and a camera.

3.2.2 Configuration

Figure 11. STM32F40x LQFP100 pinout

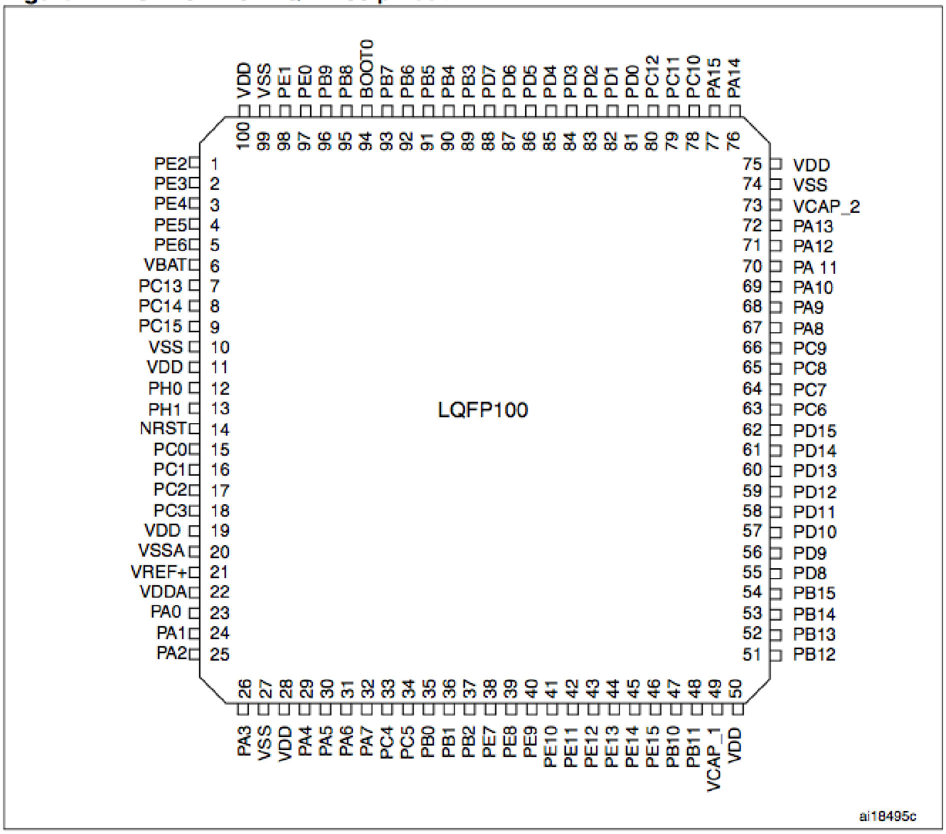


Figure D-38 STM32F407VGT6 Pin

3.2.3 Block Diagram

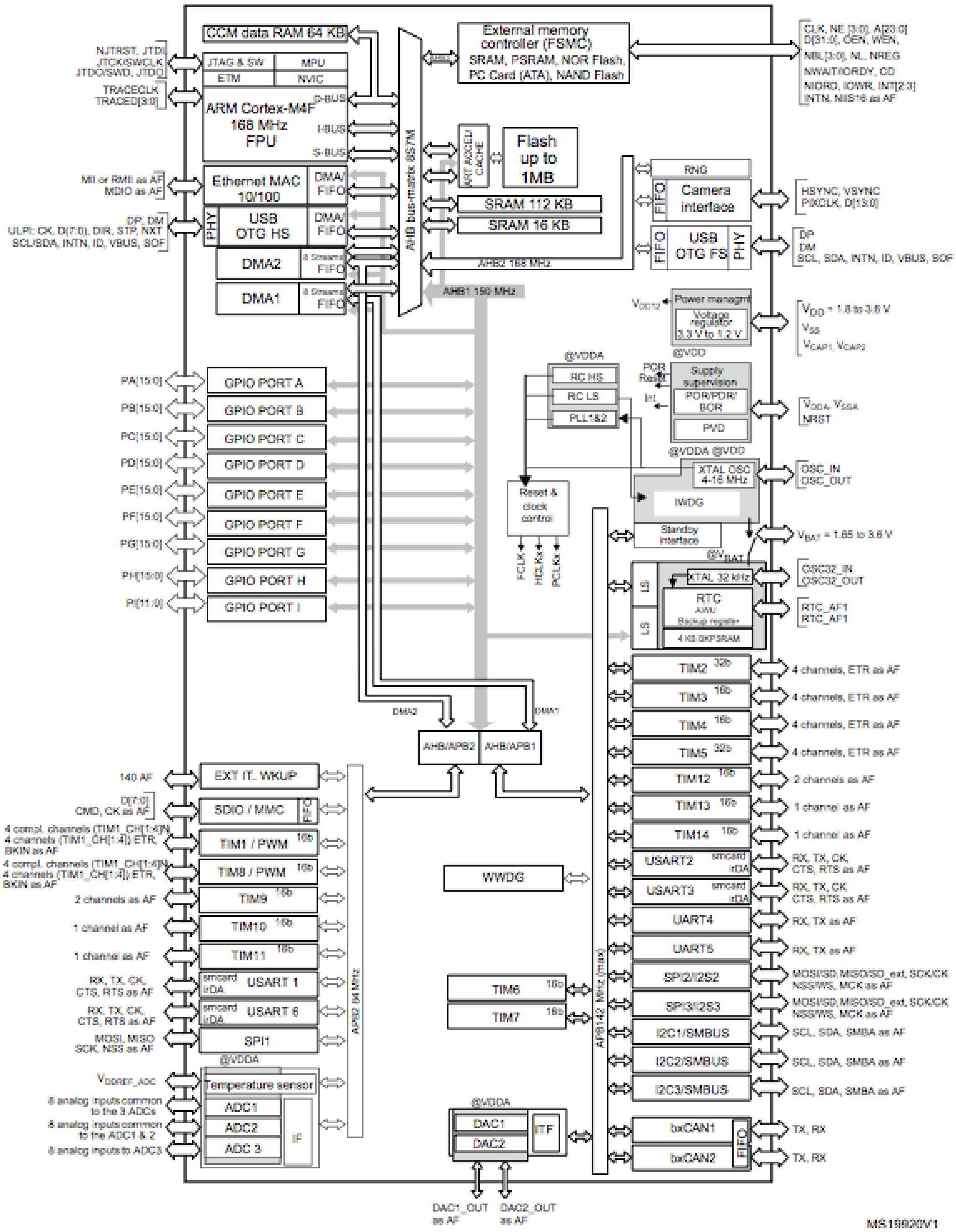


Figure D-39 STM32F407VGT6 Block Diagram

3.3. General Purposes Input / Output (GPIO)

3.3.1 Features

- Up to 16 I/O s under control
- Output states: push-pull or open drain + pull-up/down
- Output data from output data register (GPIOx_ODR) or peripheral (alternate function output)
- Speed selection for each I/O
- Input states: floating, pull-up/down, analog
- Input data to input data register (GPIOx_IDR) or peripheral (alternate function input)
- Bit set and reset register (GPIOx_BSRR) for bitwise write access to GPIOx_ODR
- Locking mechanism (GPIOx_LCKR) provided to freeze the I/O configuration
- Analog function
- Alternate function input/output selection registers (at most 16 AFs per I/O)
- Fast toggle capable of changing every two clock cycles
- Highly flexible pin multiplexing allows the use of I/O pins as GPIOs or as one of several peripheral functions

3.3.2 Configuration

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		I/O configuration	
	0			0	0	GP output	PP
	0			0	1	GP output	PP + PU
	0			1	0	GP output	PP + PD
01	0	SPEED [B:A]		1	1	Reserved	
	1			0	0	GP output	OD
	1			0	1	GP output	OD + PU
	1			1	0	GP output	OD + PD
	1			1	1	Reserved (GP output OD)	
	0			0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
10	0	SPEED [B:A]		1	1	Reserved	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	Reserved	
	x	x	x	0	0	Input	Floating
00	x	x	x	0	1	Input	PU
	x	x	x	1	0	Input	PD
	x	x	x	1	1	Reserved (input floating)	
	x	x	x	0	0	Input/output	Analog
11	x	x	x	0	1	Reserved	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = general-purpose, PP = push-pull, PU = pull-up, PD = pull-down, OD = open-drain, AF = alternate function.

Figure D-40 STM32F407VGT6 GPIO Configuration

3.3.3 Register Description

Each of the GPIOs has four 32-bit memory-mapped control registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR) to configure up to 16 I/Os. The GPIOx_MODER register is used to select the I/O direction (input, output, AF, analog). The GPIOx_OTYPER and GPIOx_OSPEEDR registers are used to select the output type (push- pull or open-drain) and speed (the I/O speed pins are directly connected to the corresponding GPIOx_OSPEEDR register bits whatever the I/O direction). The GPIOx_PUPDR register is used to select the pull-up/pull-down whatever the I/O direction.

3.4. Interrupts

All ports have external interrupt capability. To use external interrupt lines, the port must be configured in input mode.

The external interrupt/event controller consists of up to 23 edge detectors for generating event/interrupt requests. Each input line can be independently configured to select the type (pulse or pending) and the corresponding trigger event (rising or falling or both). Each line can also masked independently. A pending register maintains the status line of the interrupt requests.

Features

The main features of the external interrupt (EXTI) controller are the following:

- Independent trigger and mask on each interrupt/event line G dedicated Status bit for each interrupt line
- Generation of up to 23 software event/interrupt requests
- Detection of external signals with a pulse width lower than the APB2 clock period. Refer to the electrical characteristics section of the STM32F40x and STM32F41x datasheets for details on this parameter.

In this case, we only use Interrupt for USART, include Input (Rx) and Output (Tx).

3.5. Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART)

3.5.1 Usage and Purpose

In this project, the use of USART will be the main communication channel between STM32F4 Discovery module and other modules. The command will be receive through this channel as well as the response from STM32F4 (when required).

3.5.2 Baud Rate

Baud rate for standard USART (SPI mode included)

$$\text{Tx/Rx baud} = \frac{f_{CK}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

(*)

Figure D-41 Baud Rate Calculation

Notes: (*) Refer to STM32F407VGT6 Datasheet for more details.

3.6. APOD’s Leg control

This is the overview of APOD Leg’s structure:

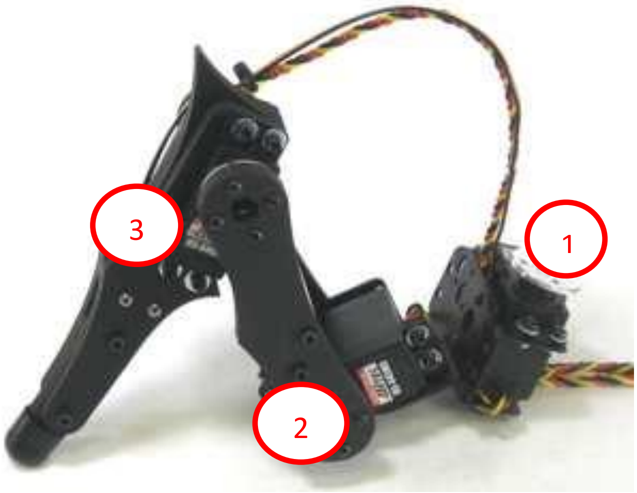


Figure D-42 APOD Leg

The APOD leg contain 3 servo:

- 1 servo responsible for Horizontal orientation (servo 1)
- 2 servos responsible for Vertical orientation (servo 2 and 3)

The position of the leg will be represent by a set of 3 number (α, β, γ) where α, β, γ is respectively the position of servo 1, 2 and 3 (in µS).

Setting the position of servo 1 will be the main cause for the leg to move in the horizontal orientation, in this case will be the transition between 3 positions: Forward, Center, Backward.

Setting the position of servo 2 and 3 will be the main cause for the leg to move in vertical orientation, in this case will be the transition between 3 positions: High, Middle, Low.

IV. ACTIVITY AND CLASS STRUCTURE

1. APOD Controller (HMI Application)

1.1. Activity

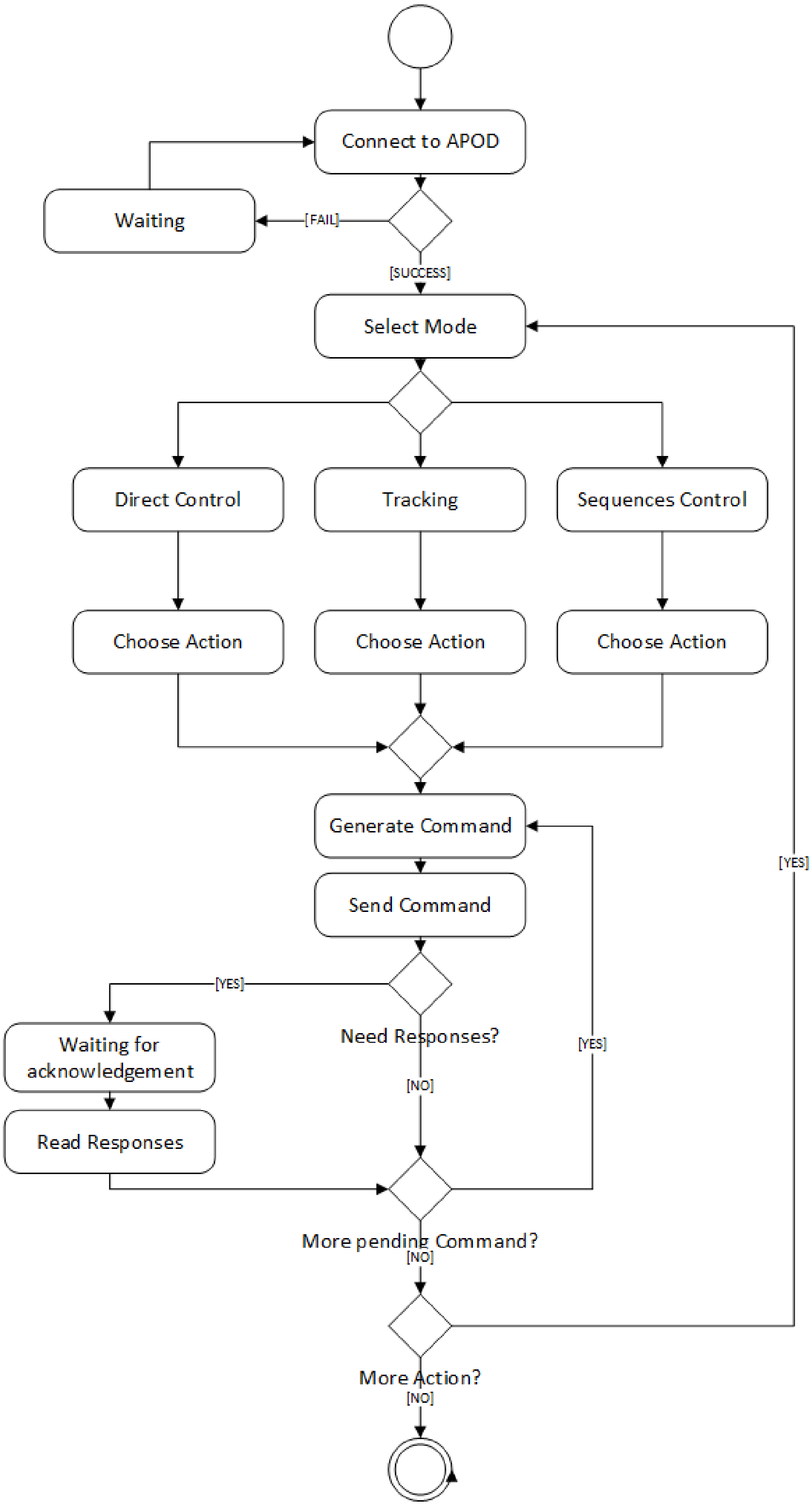


Figure D-43 APOD Controller Application Activity

1.2. Class Diagram

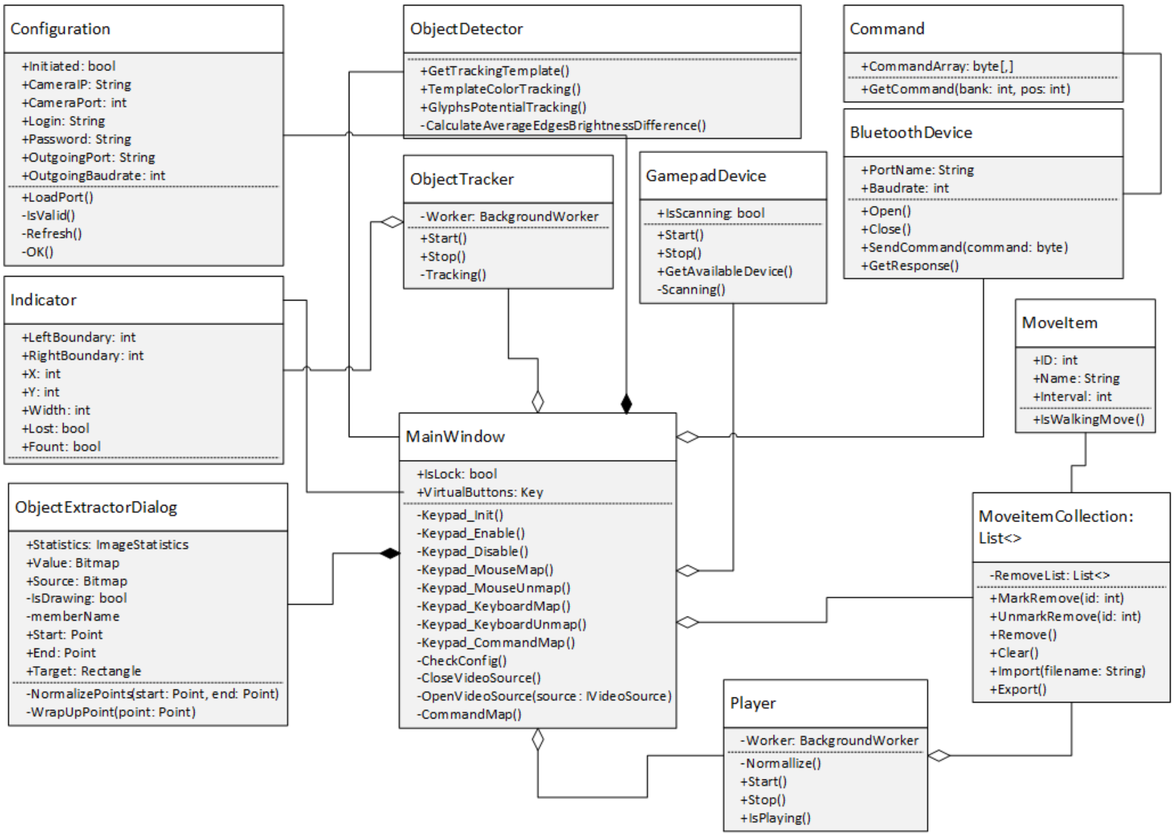


Figure D-44 Class diagram

1.3. External Library

AForge.NET (Kirillov, n.d.) framework including namespaces to support image processing:

- Aforge
- Aforge.Controls
- Aforge.Imaging
- Aforge.Math
- Aforge.Video
- Aforge.Video.DirectShow
- Aforge.Vision

MahApps.Metro UI toolkit (Jenkins, n.d.):

- MahApps.Metro

2. STM32F4 Discovery (Central processing board)

2.1. Activity

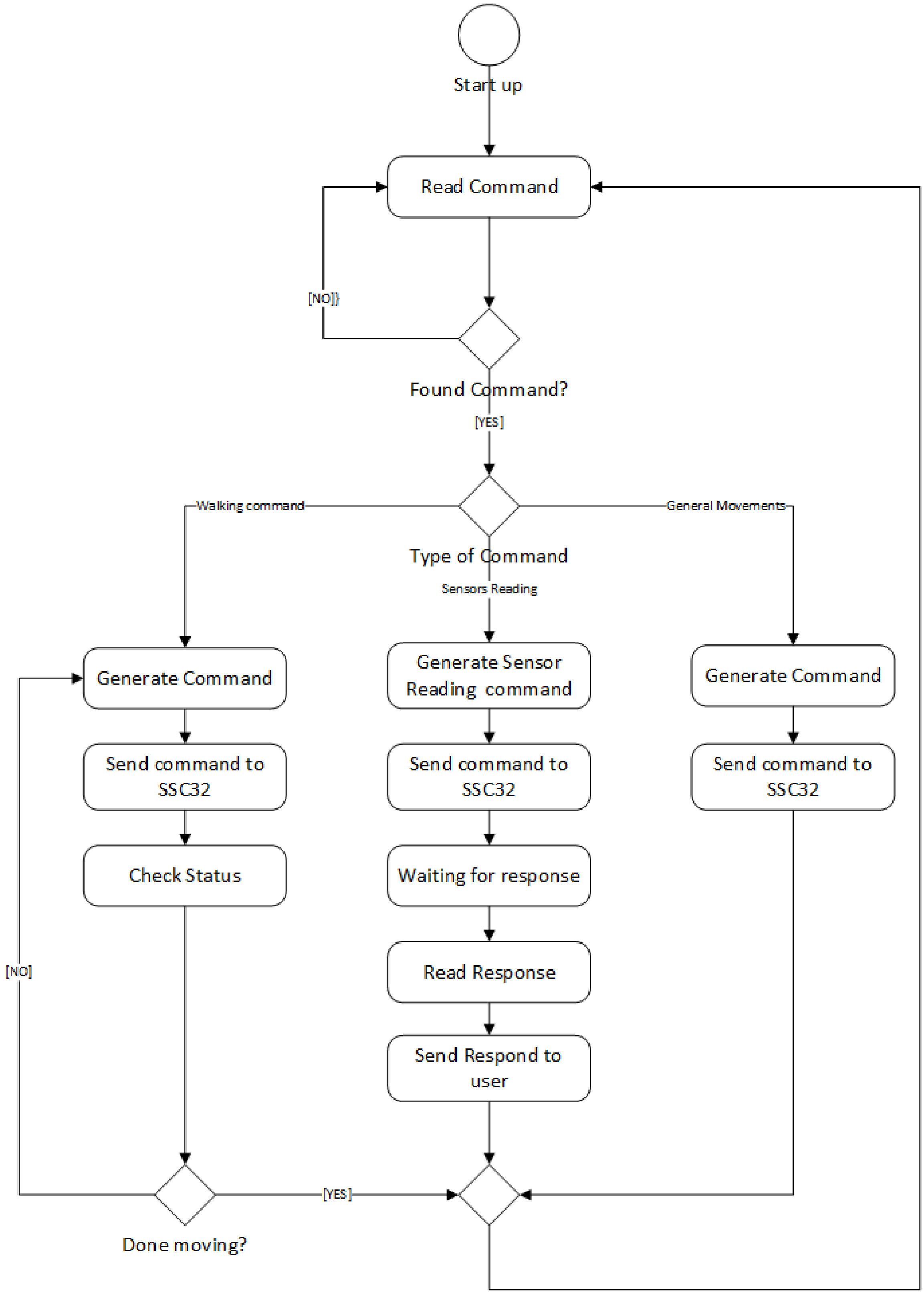


Figure D-45 STM32F4 Activity

2.2. Command Sheet

Command	Description
0x11	Reset APOD, all servos are set to balance stage
0x31	Start APOD, all servos are set to 1500 pulse width at initial stage
0x32	Stop APOD, all servos are discharged
0x33	Move APOD forward with limited loop
0x34	Move APOD Backward with limited loop
0x35	Move APOD Left with limited loop
0x36	Move APOD Right with limited loop
0x37	APOD's body is toward the front
0x38	APOD's body is toward the back
0x39	APOD's body is squeezed to Left
0x40	APOD's body is squeezed to Right
0x41	Stand Lift, APOD's body also stand and is higher
0x42	Stand Drop, APOD's body also stand and is lower
0x43	Stand Balance, APOD's body stand in balance
0x44	Wave APOD's Tail
0x45	Rotate APOD's head to Left
0x46	Rotate APOD's head to Right
0x47	Turn APOD's head to Left
0x48	Turn APOD's head to Right
0x49	Turn APOD's head Up
0x50	Turn APOD's head Down
0x51	Grip APOD's Mandible
0x52	Release APOD's Mandible
0x53	Greeting Audience
0x3A	Advance move Forward, APOD move until order (command from user by release controller) is end
0x3B	Advance move Backward, APOD move until order (command from user by release controller) is end
0x3C	Advance turn Left, APOD move until order (command from user by release controller) is end
0x3D	Advance turn Right, APOD move until order (command from user by release controller) is end
0x13	Read Distance sensor and send to PC
0xA1	Auto Grip, APOD grip mandible until reach the condition

Table D-10 STM32F4 Command Sheet

3. SSC32 Board (Servos control and Sensor reader)

3.1. Activity

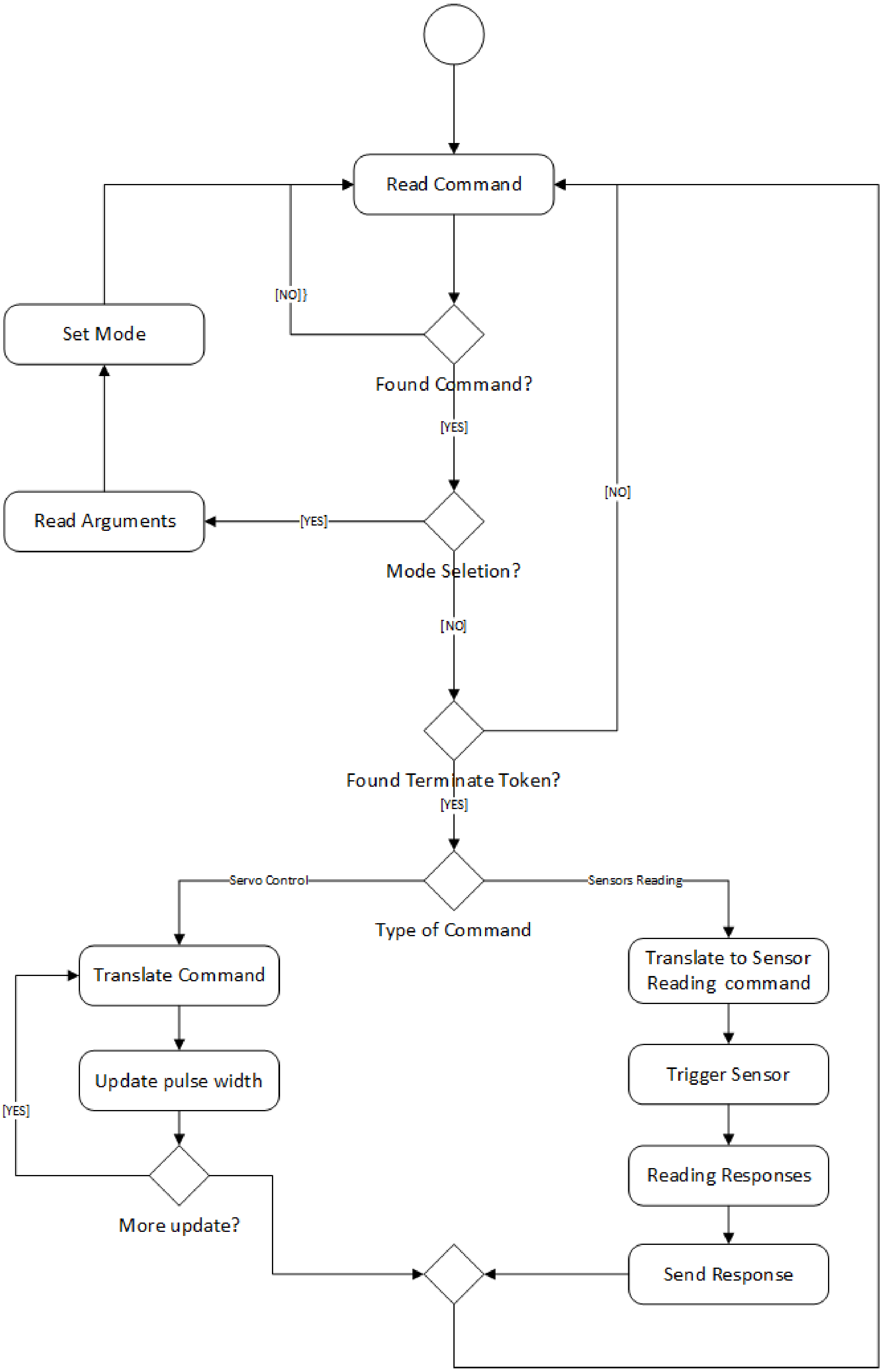


Figure D-46 SSC32 Activity

3.2. Command Sheet

Command	Argument 1	Argument 2	Description
[1 byte]	*[1 byte]*	*[1 byte]*	
'I'	[interval length]	N/A	Selecting speed for servo move. The [*interval length*] argument take value from 0 to 31 describe the waiting time (in ms) of each 20 µS pulse of servo. (*)
'D'	N/A	N/A	Get the distance sensor value, return 2-bytes response contain the sonar travel time.
'V'	[channel]	N/A	Get the ADC value from selected channel. 2 available channel: 'C' and 'D'
'S'	[mode]	N/A	Start-up command for all servos: - [*mode*] = 1: Start - [*mode*] = 2: Stop
Note: (*) Refer to the Implementation Approaches section for more information.			

Table D-11 Mode & Reader command (SSC32)

Command	Argument 1	Argument 2	Description
[1 byte]	*[1 byte]*	*[2 bytes]*	
'#'	[channel]	[pulse width]	Main servo command. Rotate servo at [*channel*] to the [*pulse width*] position. - [*channel*]: select servo channel - [*pulse width*]: pulse width value corresponding for the servo position
'T'	[mode]	N/A	Terminate command that allow servos to start moving to their new position set up by '#' command. - [mode] = 0: move with maximum speed - [mode] = 1: move with limited speed. (*)
Note: (*) Refer to the Implementation Approaches section for more information.			

Table D-12 Servo Control command (SSC32)

V. SEQUENCE DIAGRAM

1. Embedded Controller

1.1. Forward

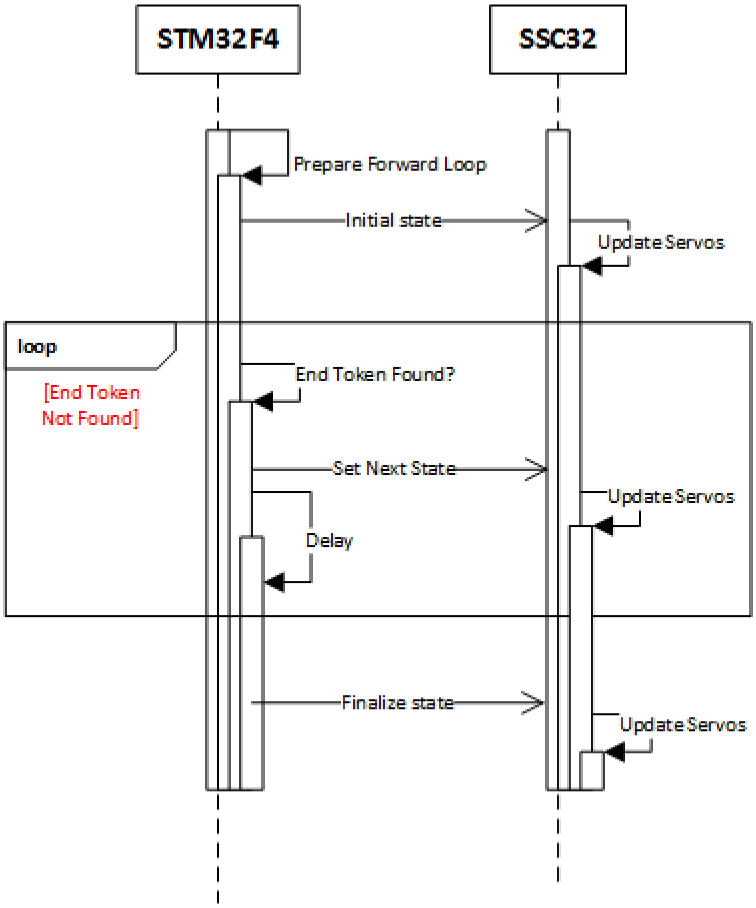


Figure D-47 Embbed Forward sequence

1.2. Backward

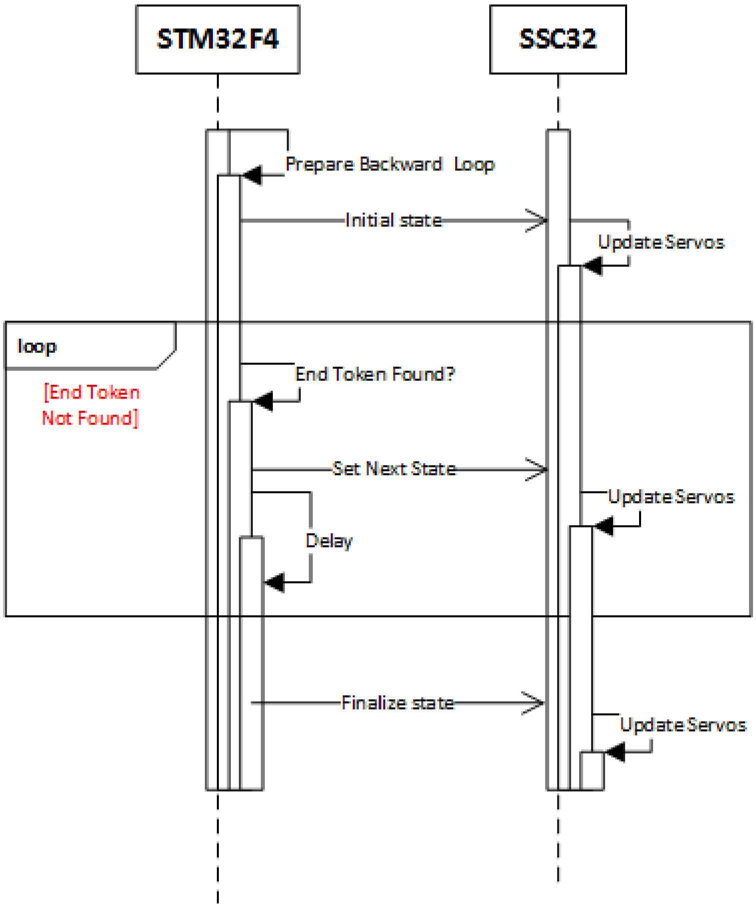


Figure D-48 Embbed Backward sequence

1.3. Turn Left

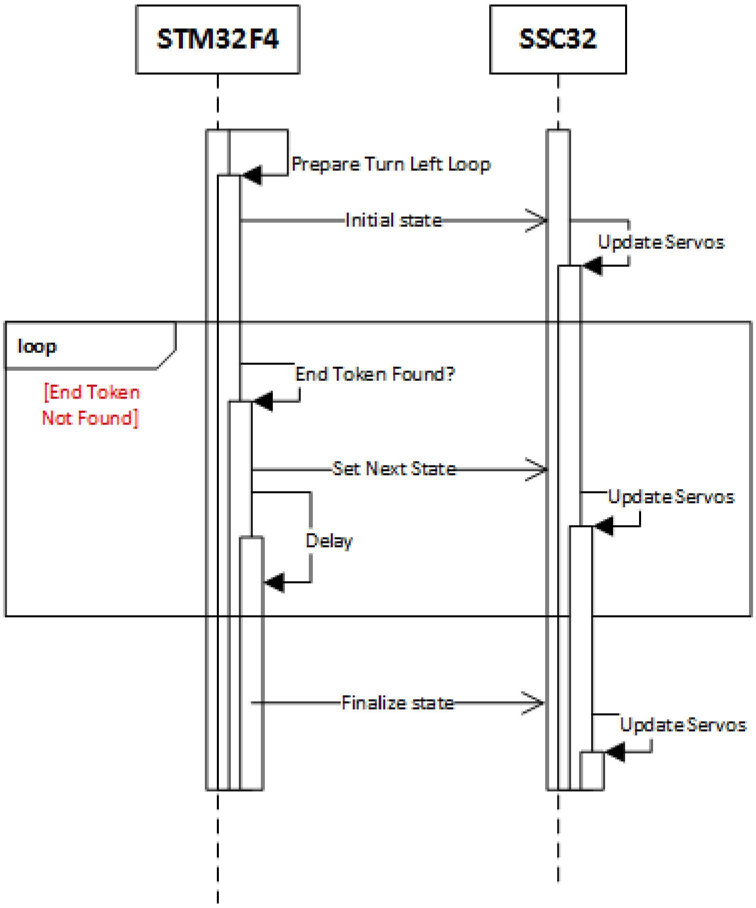


Figure D-49 Embbed Turn left sequence

1.4. Turn Right

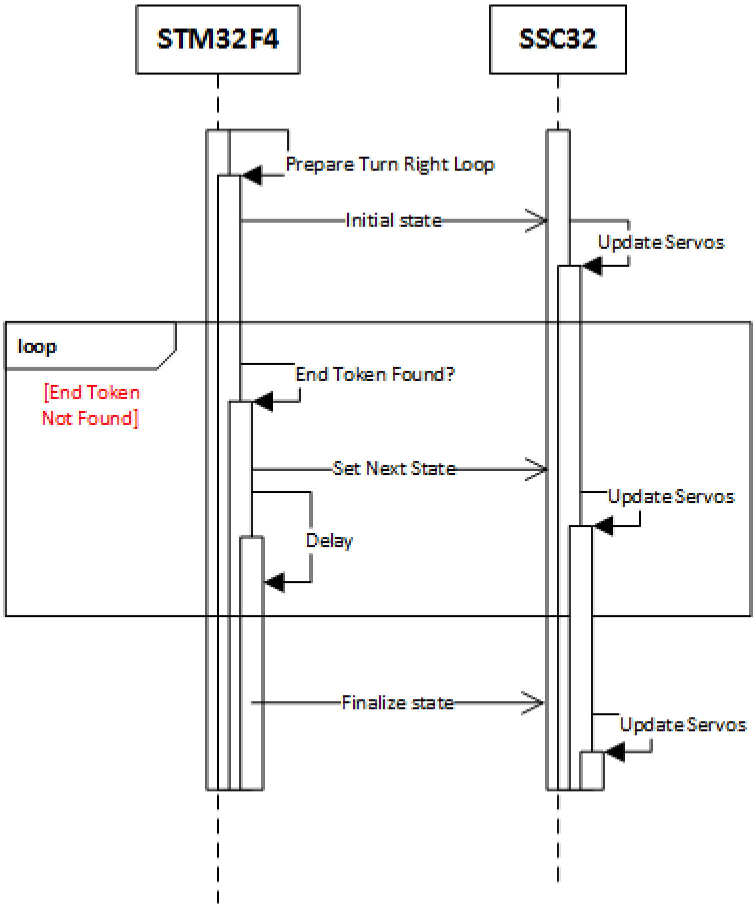


Figure D-50 Embbed Turn Right sequence

2. Use-cases Sequences

2.1. Sequences Control

2.1.1 Add State

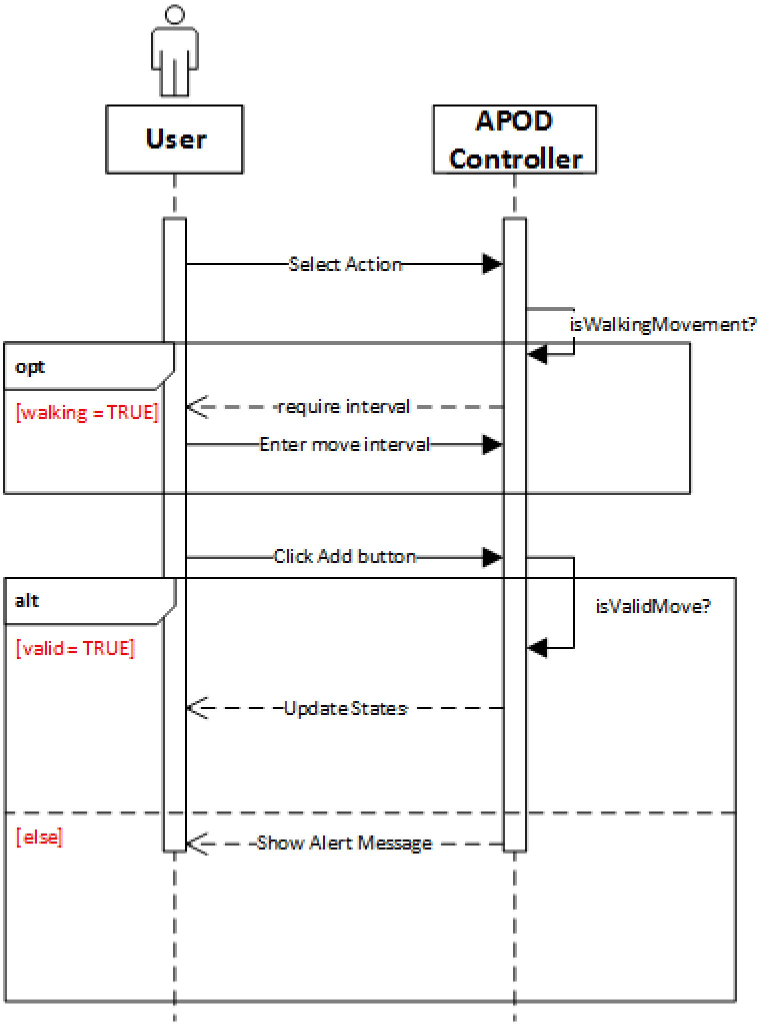


Figure D-51 Add State Use case sequence

2.1.2 Remove States

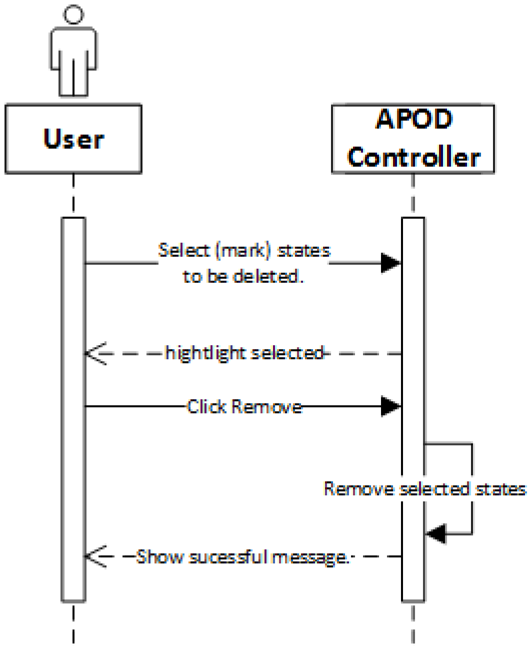


Figure D-52 Remove States Use case sequence

2.1.3 Import States

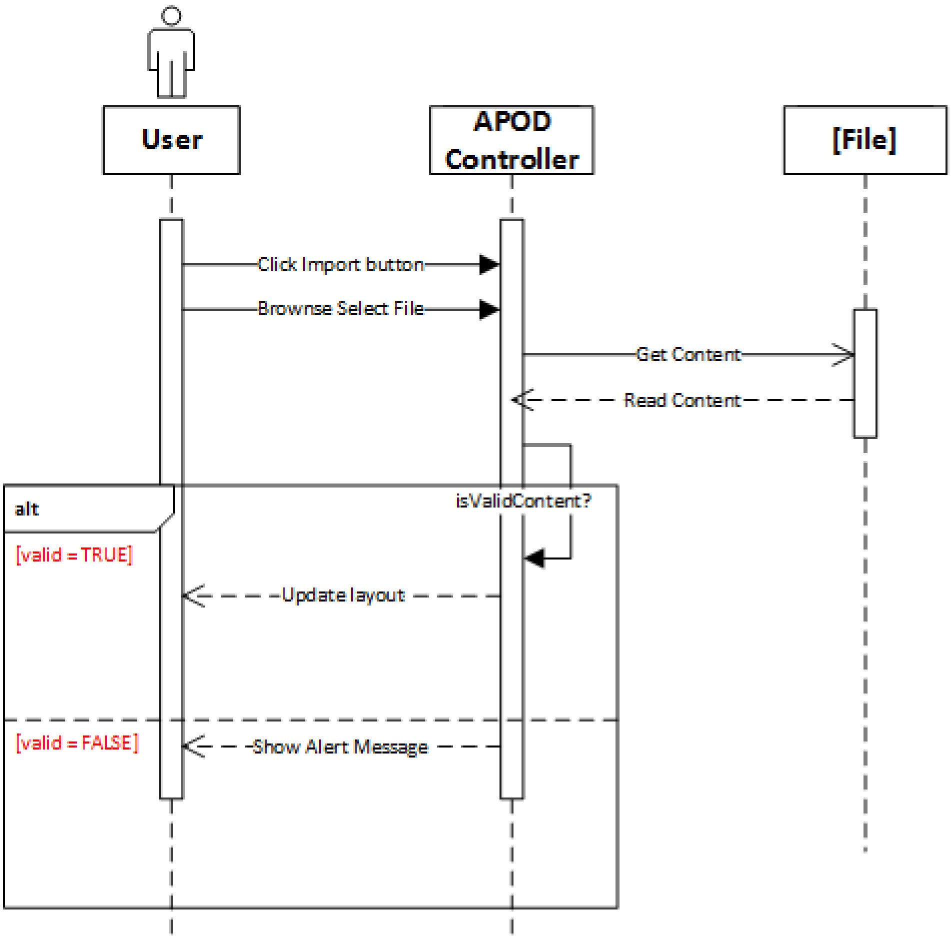


Figure D-53 Import States Use case sequence

2.1.4 Export States

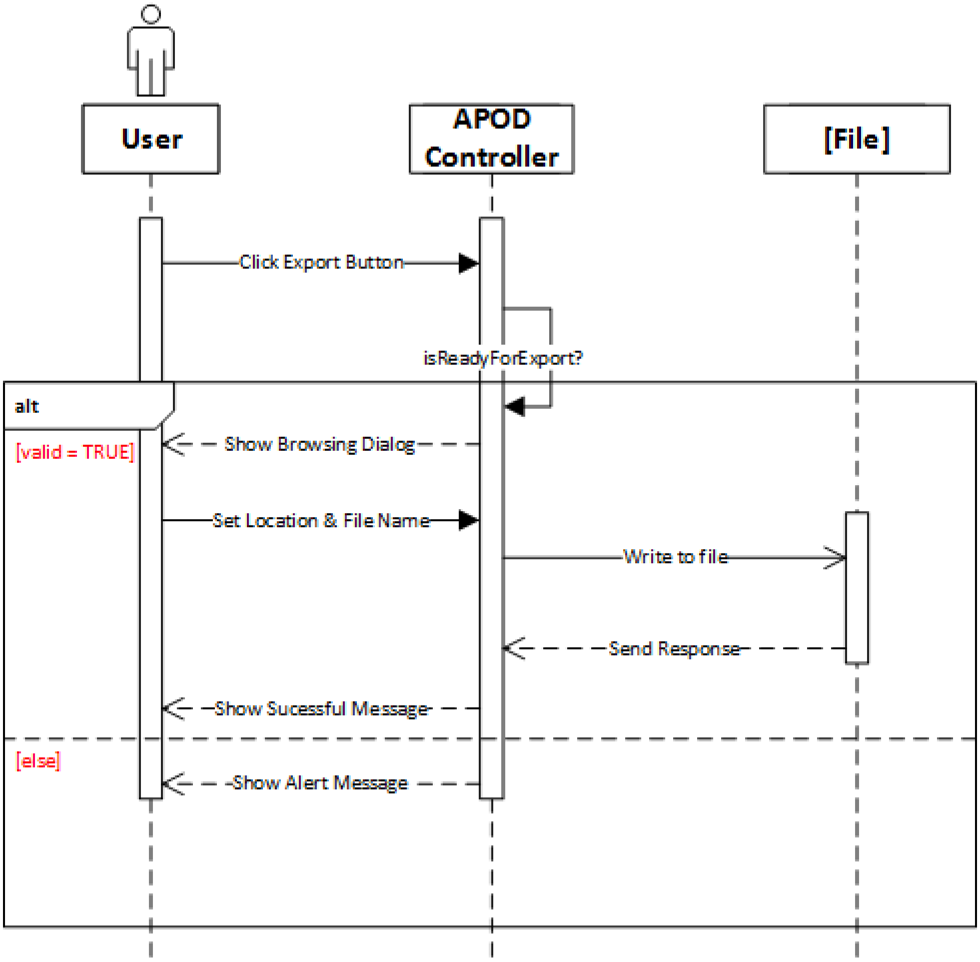


Figure D-54 Export States Use case sequence

2.2. Direct Control

2.2.1 Mode Selection

a) Normal Input

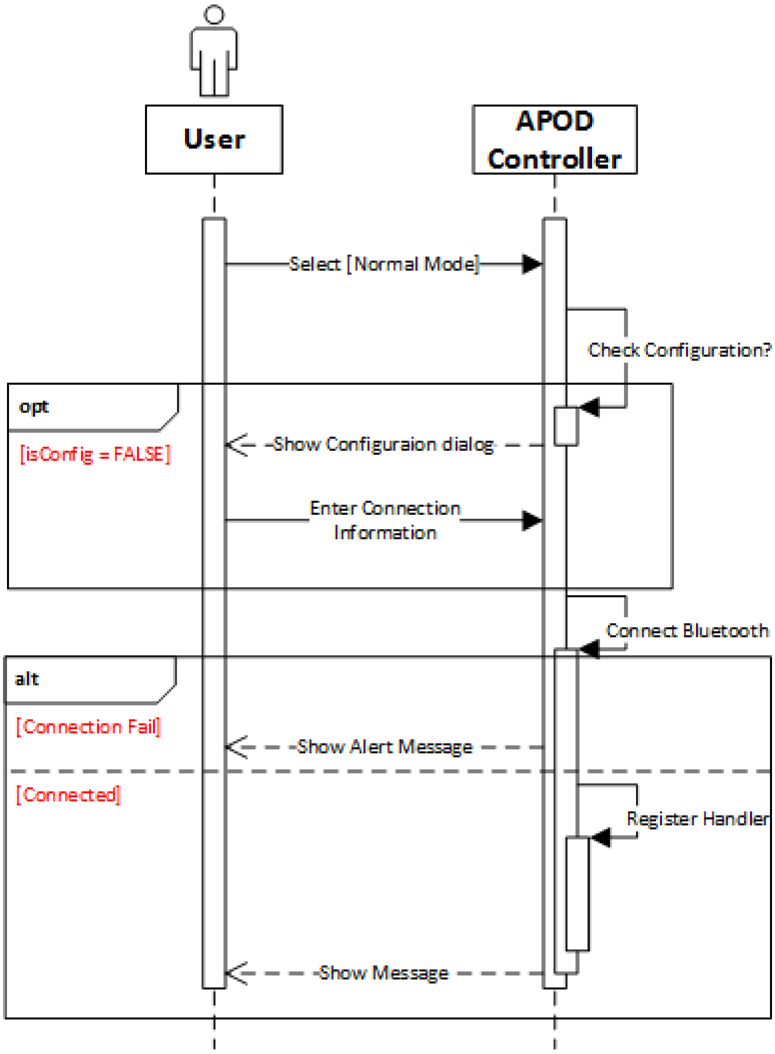


Figure D-55 Normal Input Use case sequence

b) <u>Gamepad Input</u>

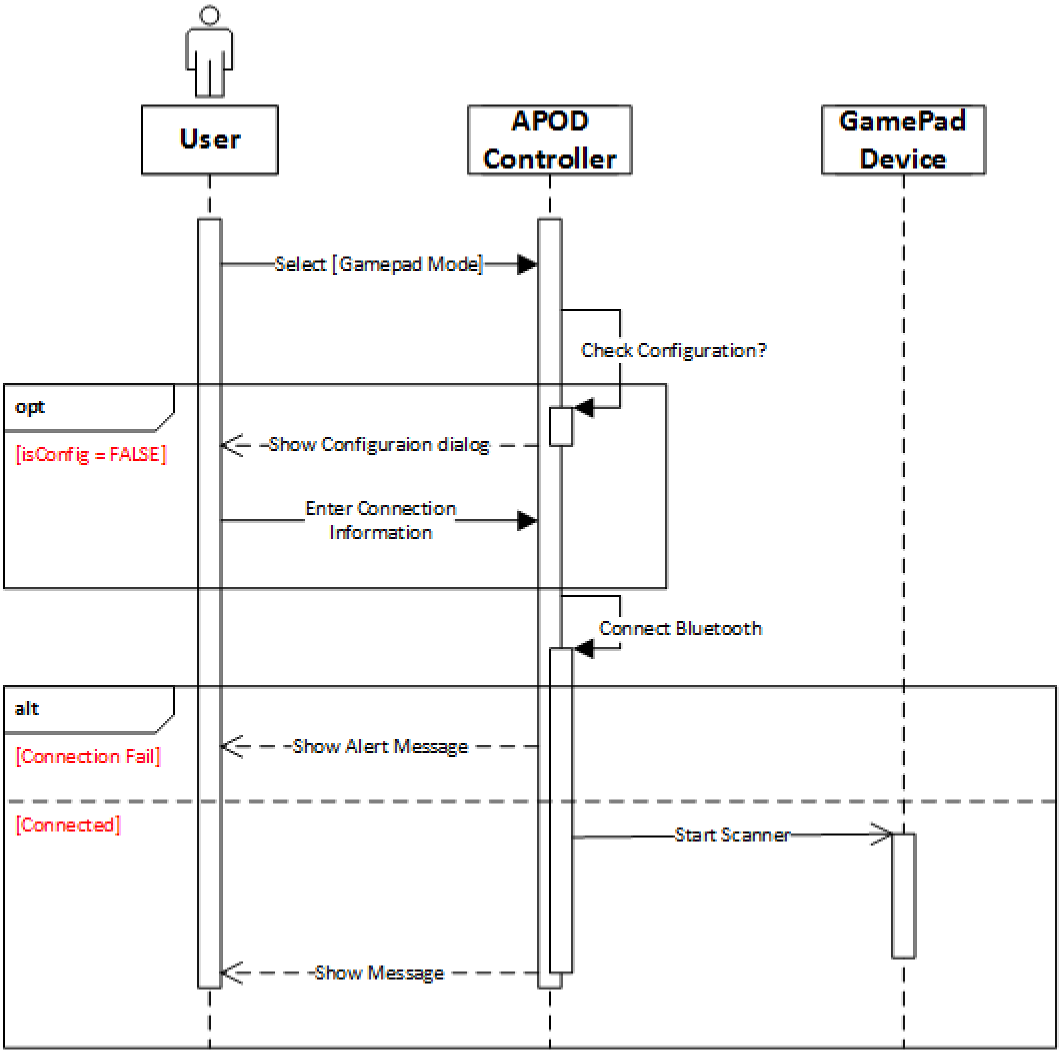


Figure D-56 Gamepad Input Use case sequence

2.2.2 Movement Control

a) Forward

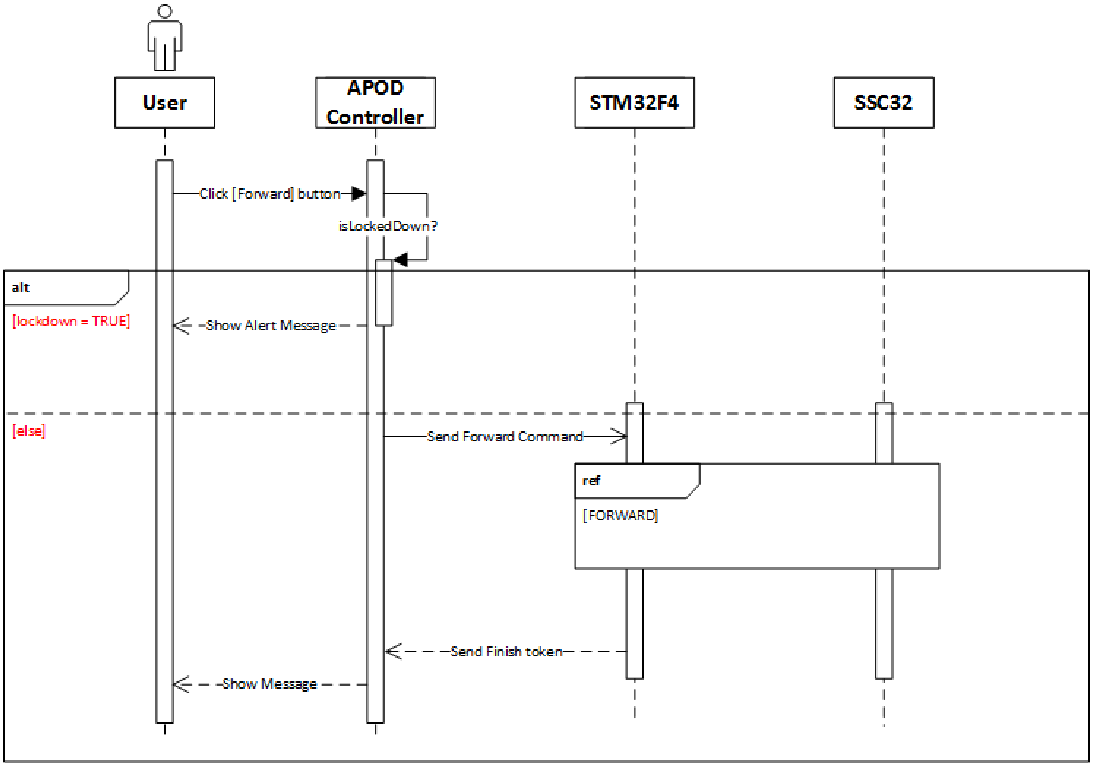


Figure D-57 Move Forward Use case sequence

b) Backward

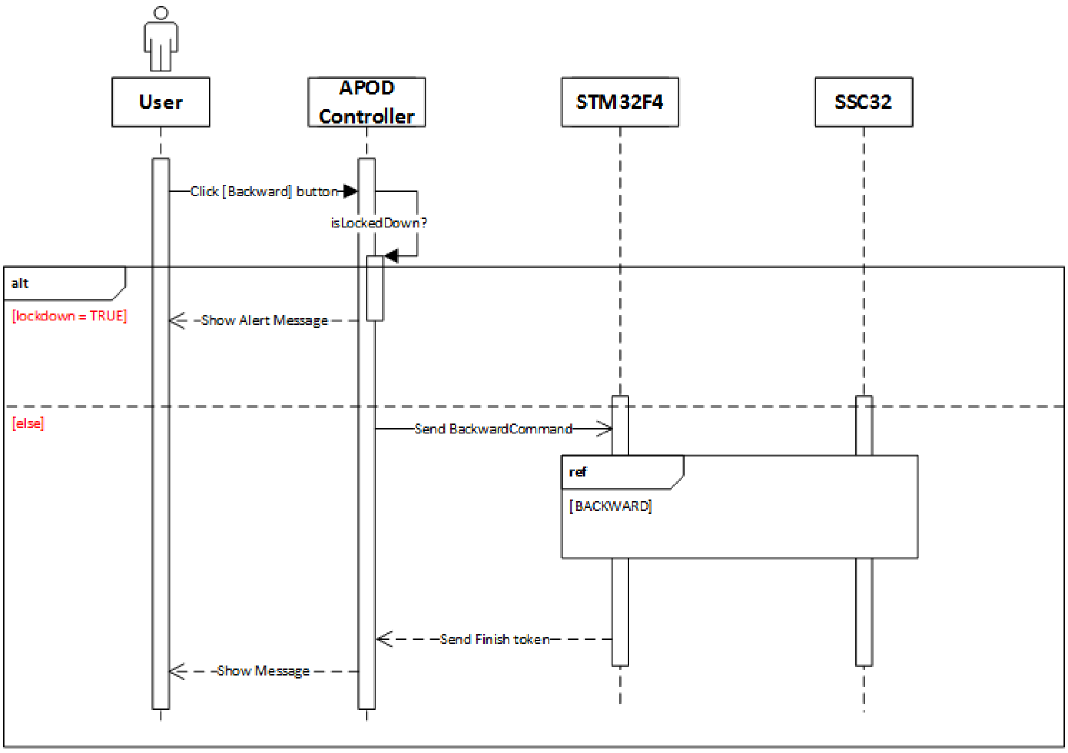


Figure D-58 Move Backward Use case sequence

c) Turn Left

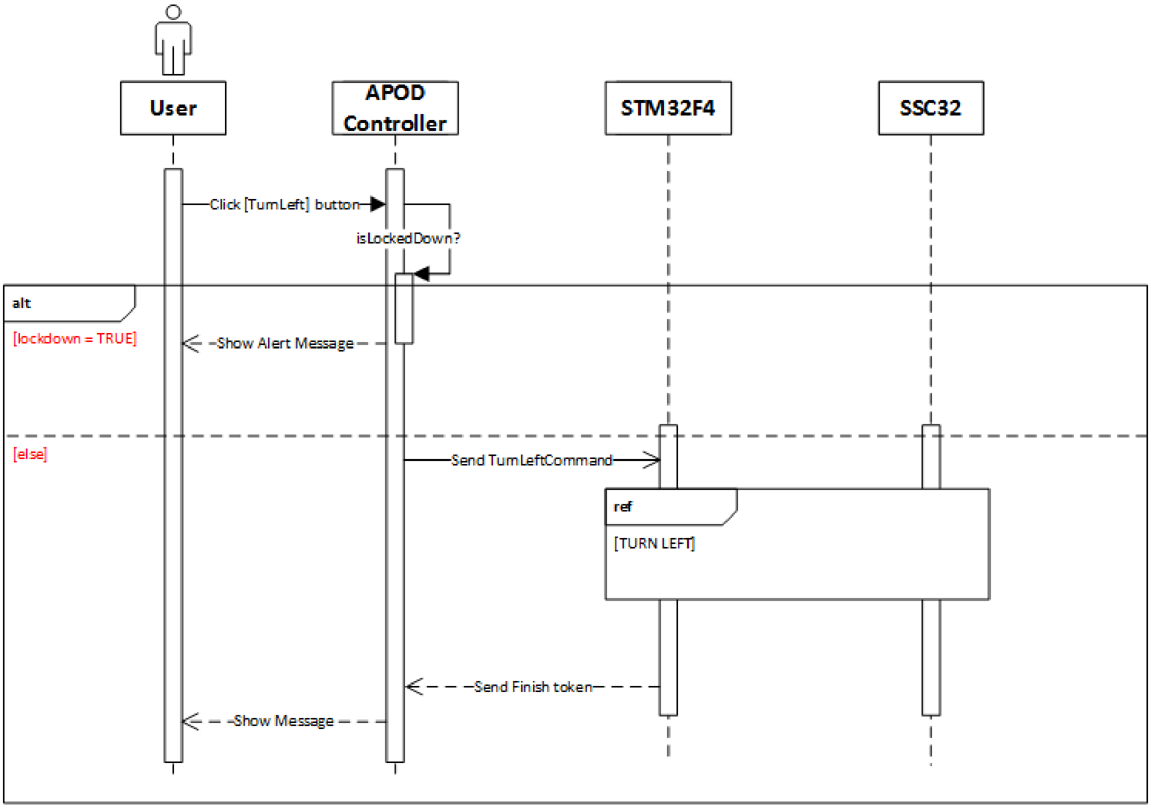


Figure D-59 Turn Left Use case sequence

d) Turn Right

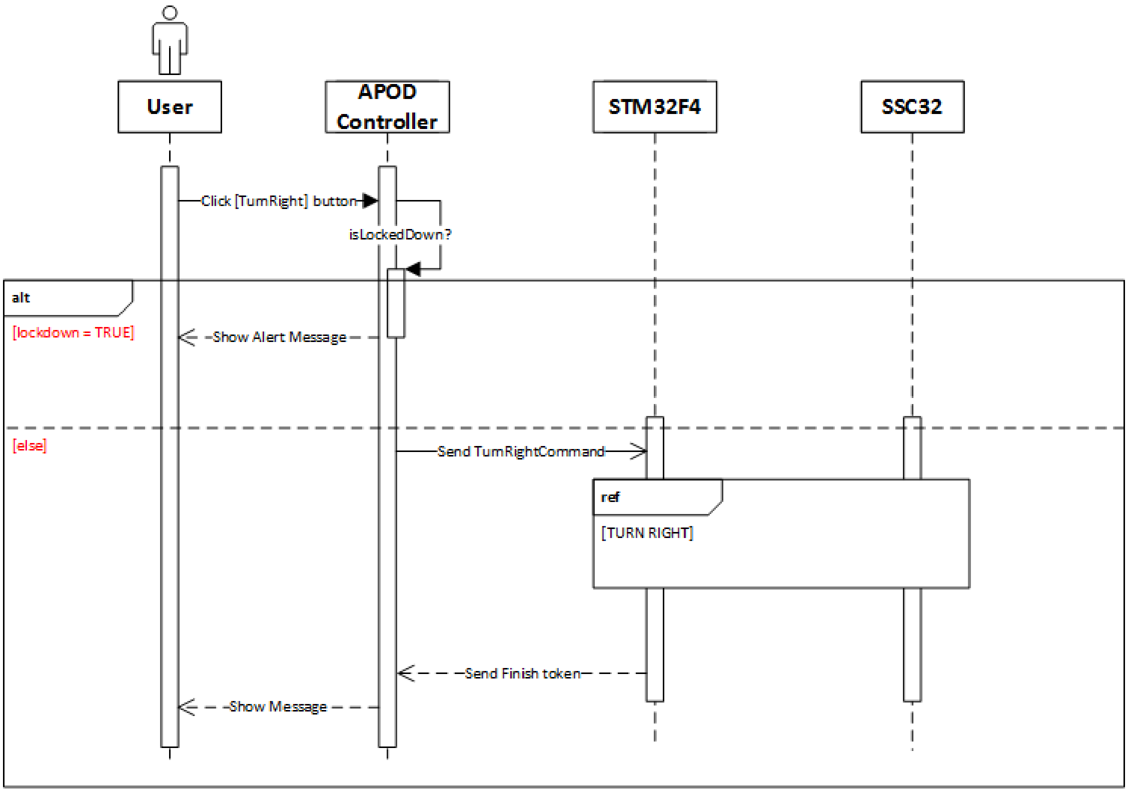


Figure D-60 Turn Right Use case sequence

e) Body Lift

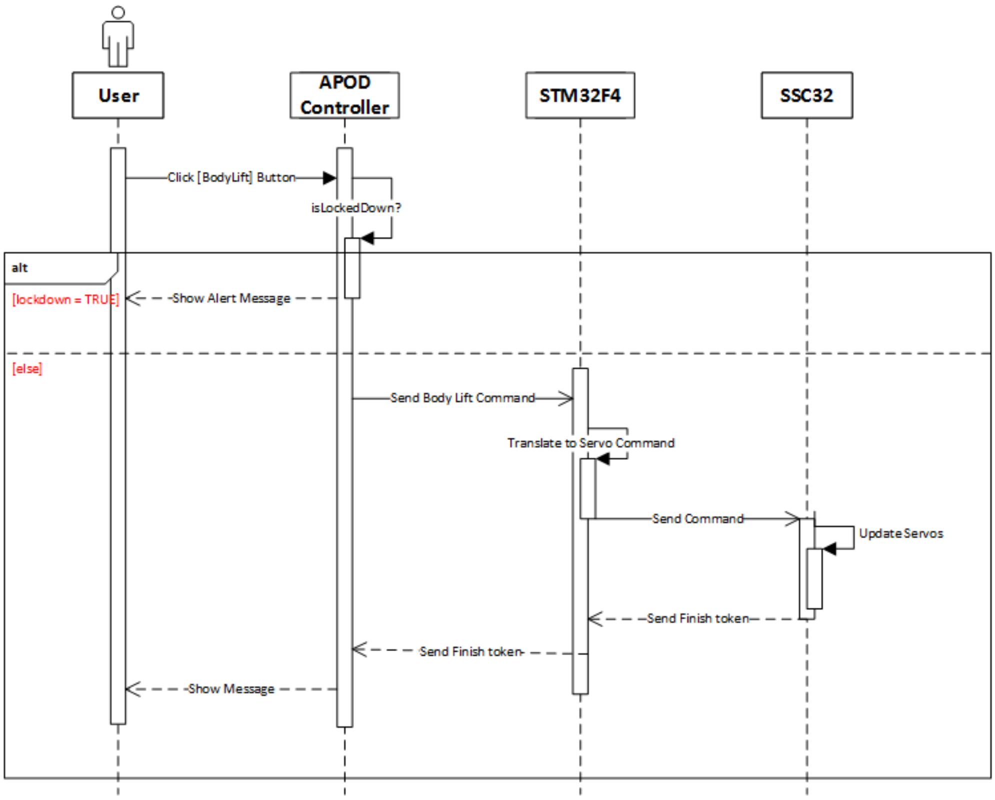


Figure D-61 Body Lift Use case sequence

f) Body Drop

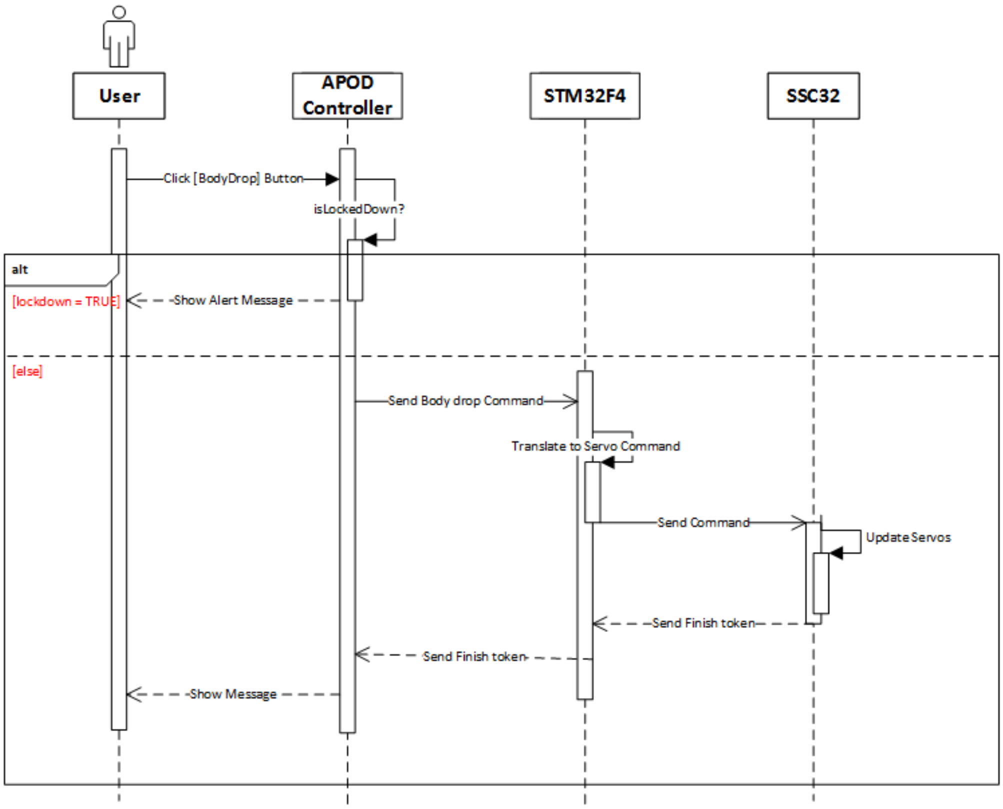


Figure D-62 Body Drop Use case sequence

g) Squeeze Left

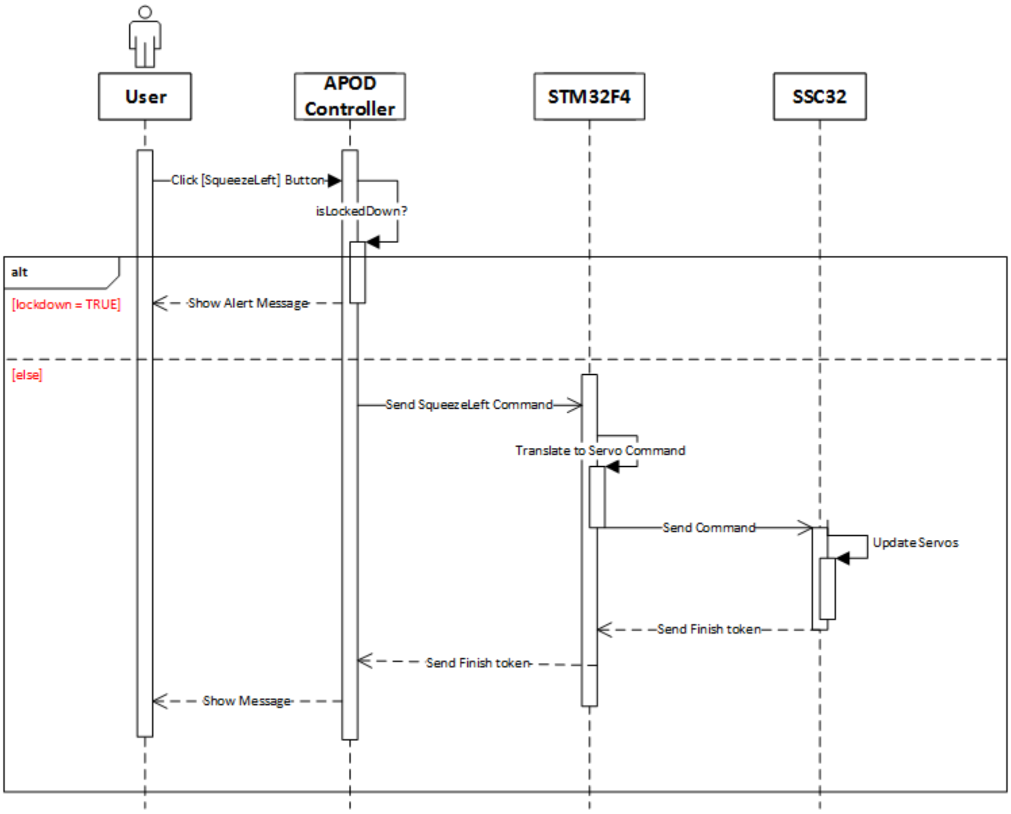


Figure D-63 Squeeze Left Use case sequence

h) Squeeze Right

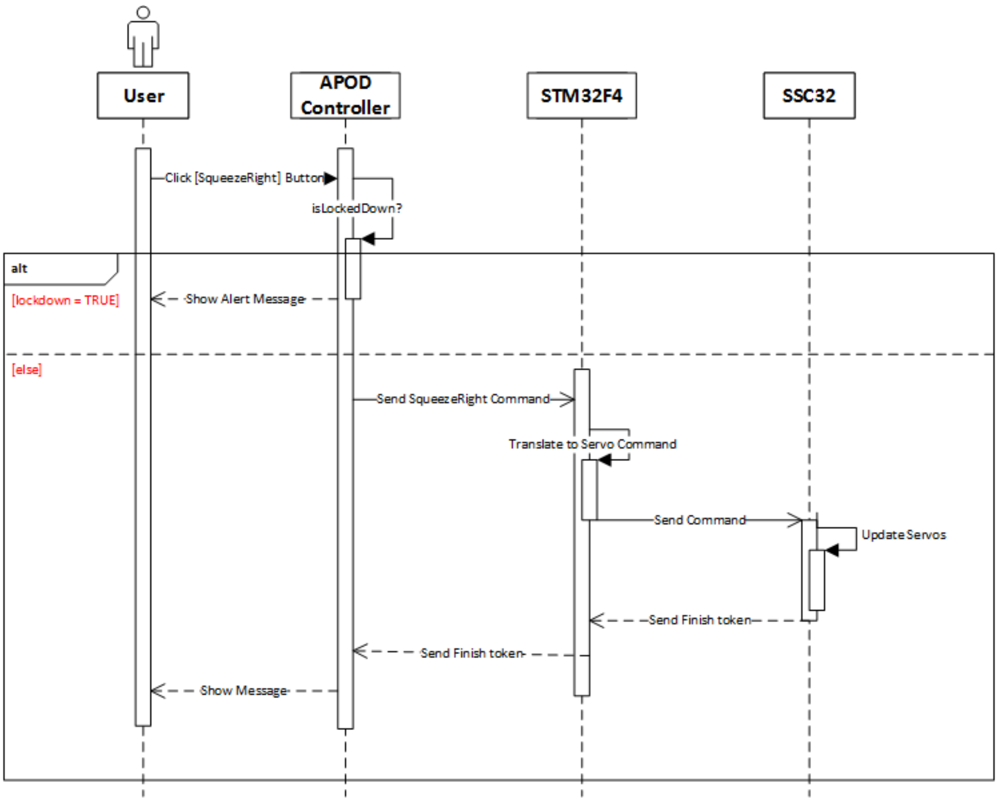


Figure D-64 Squeeze Right Use case sequence

i) Manual Grip/Release

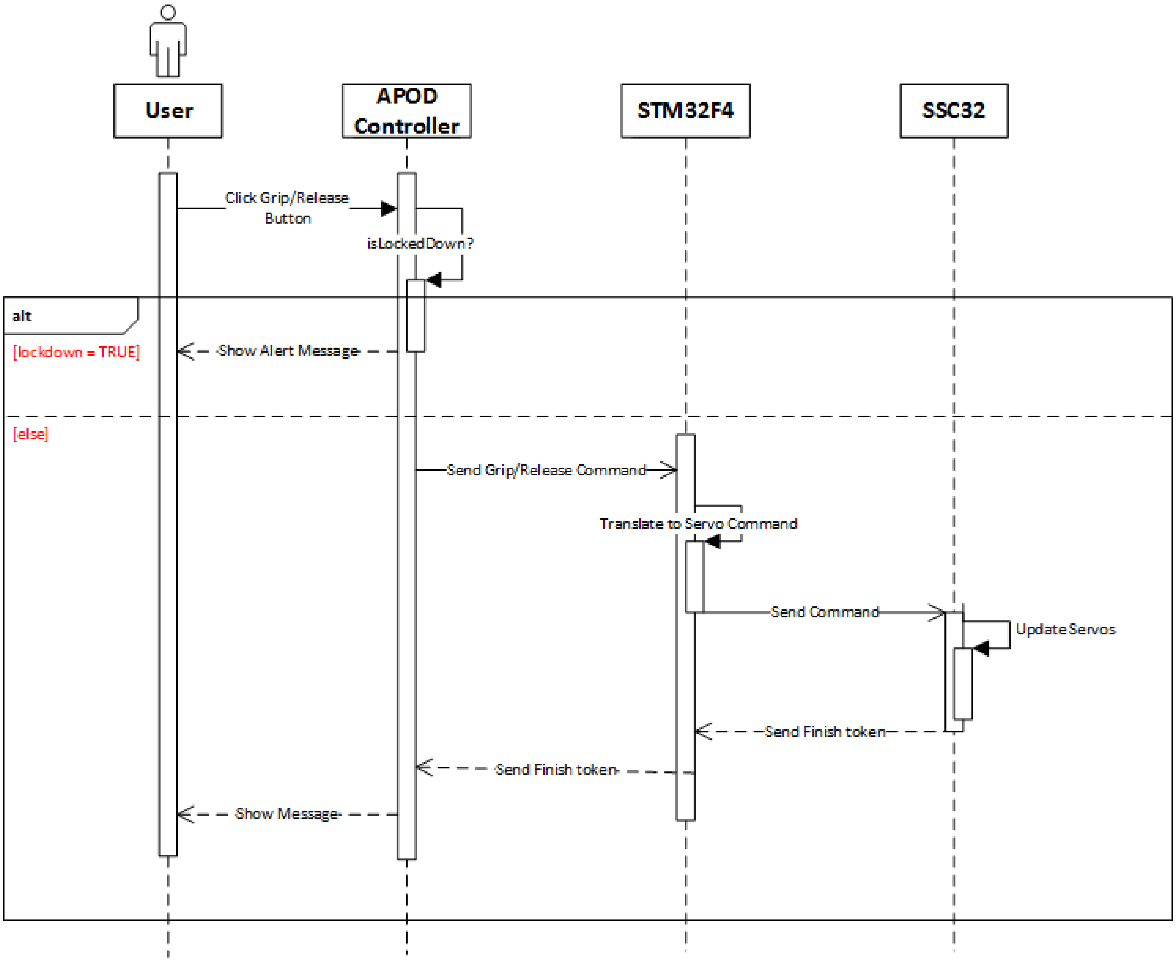


Figure D-65 Manual Grip/Release Use case sequence

2.3. Object Tracking

2.3.1 Auto Grip

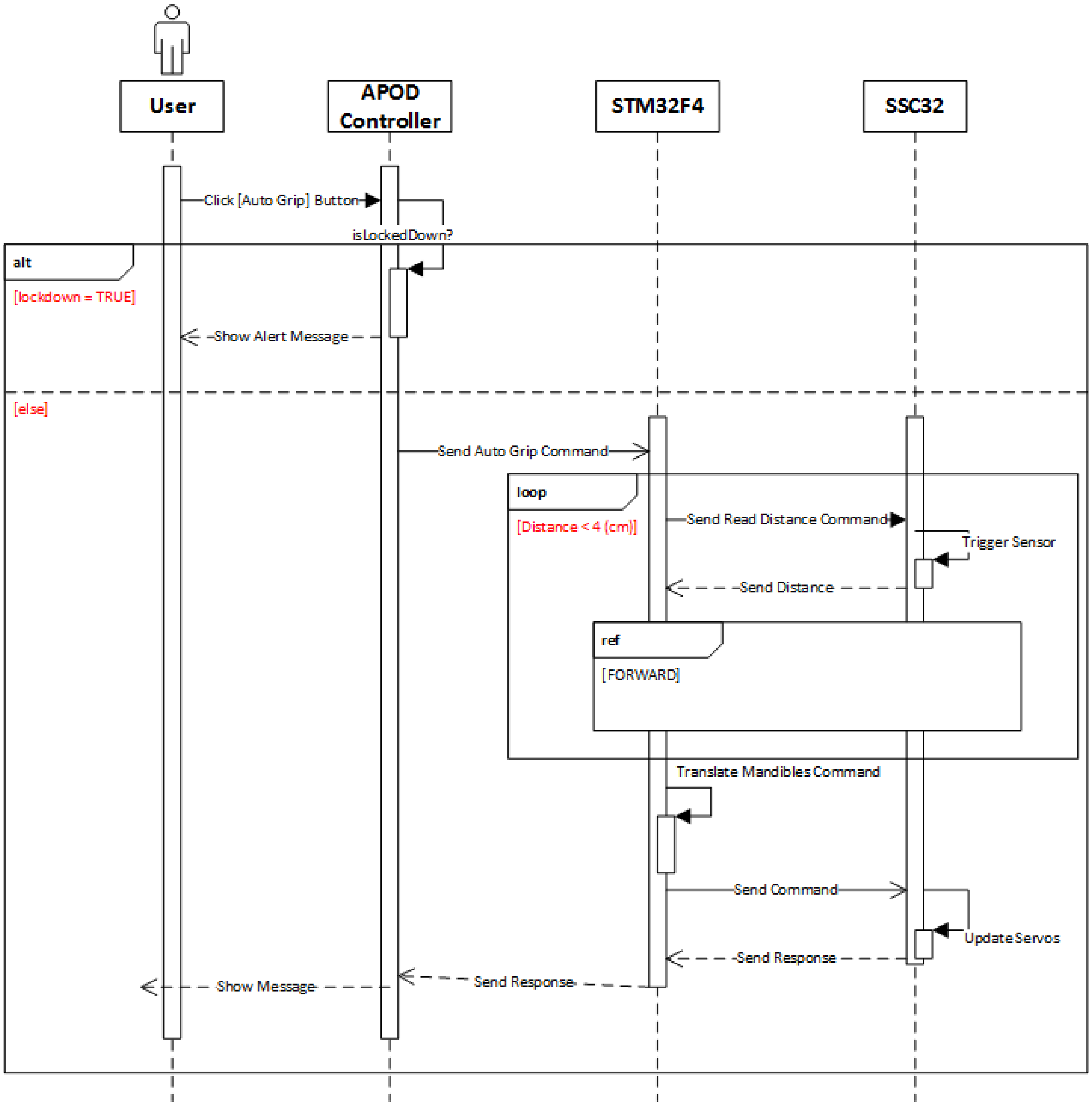


Figure D-66 Auto Grip Use case sequence

2.3.2 Choose target

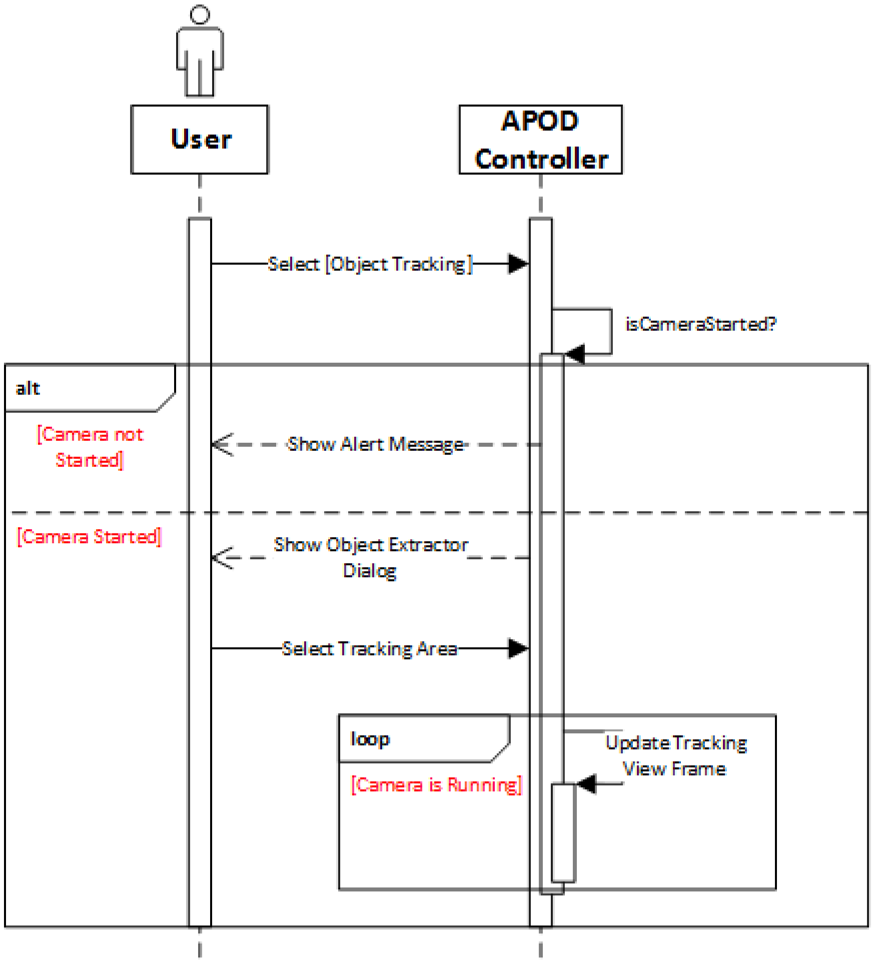


Figure D-67 Choose Target Use case sequence

2.3.3 Change Target

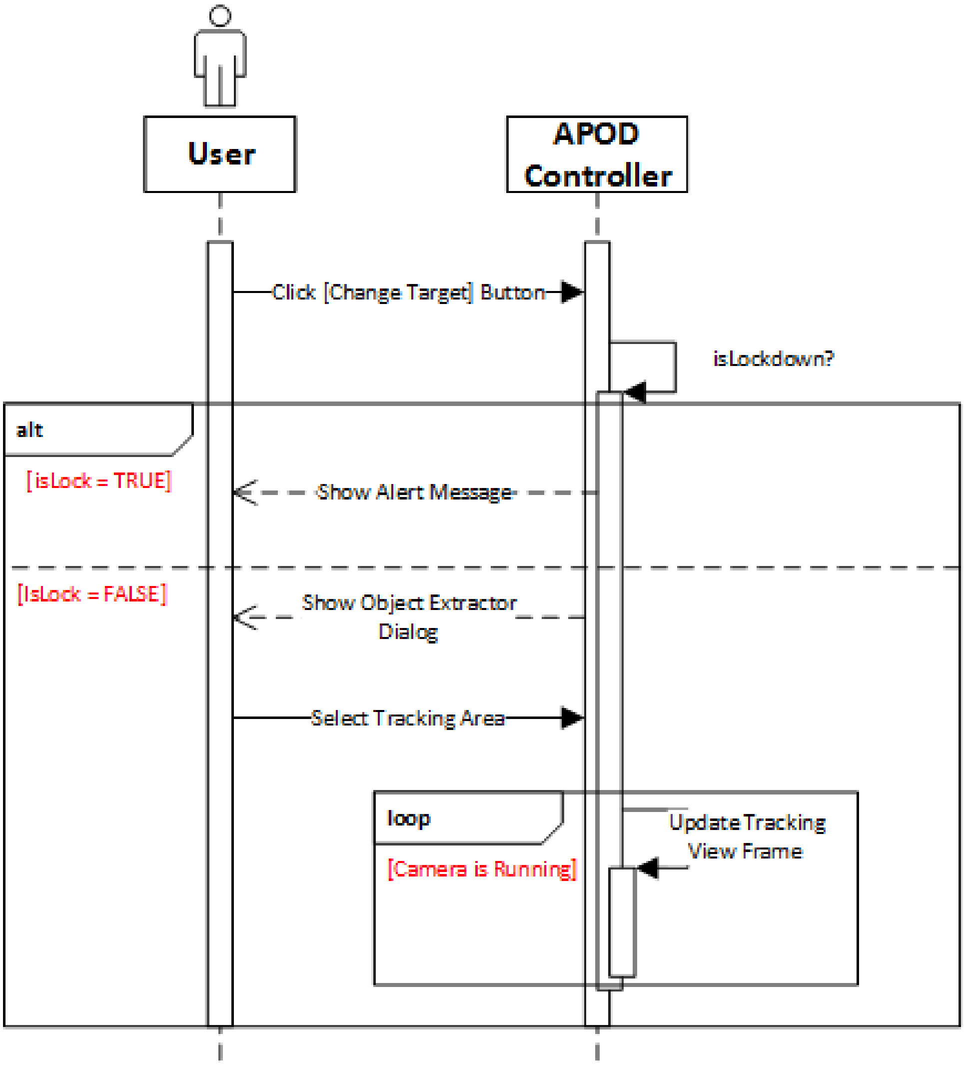


Figure D-68 Change Target Use case sequence

2.3.4 Lock Target

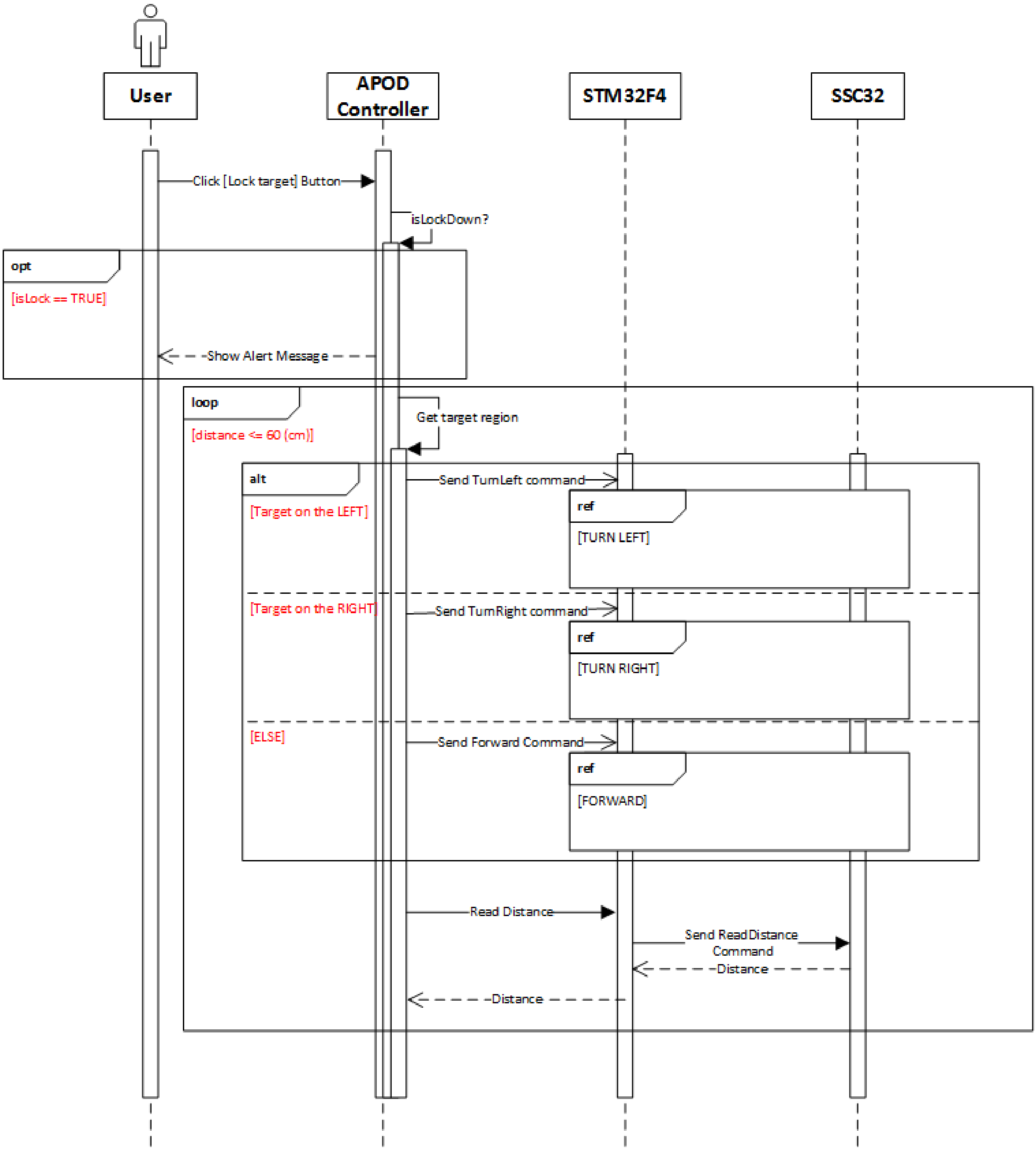


Figure D-69 Lock Target Use case sequence

VI. USER INTERFACE DESIGN

1. *Live Control*

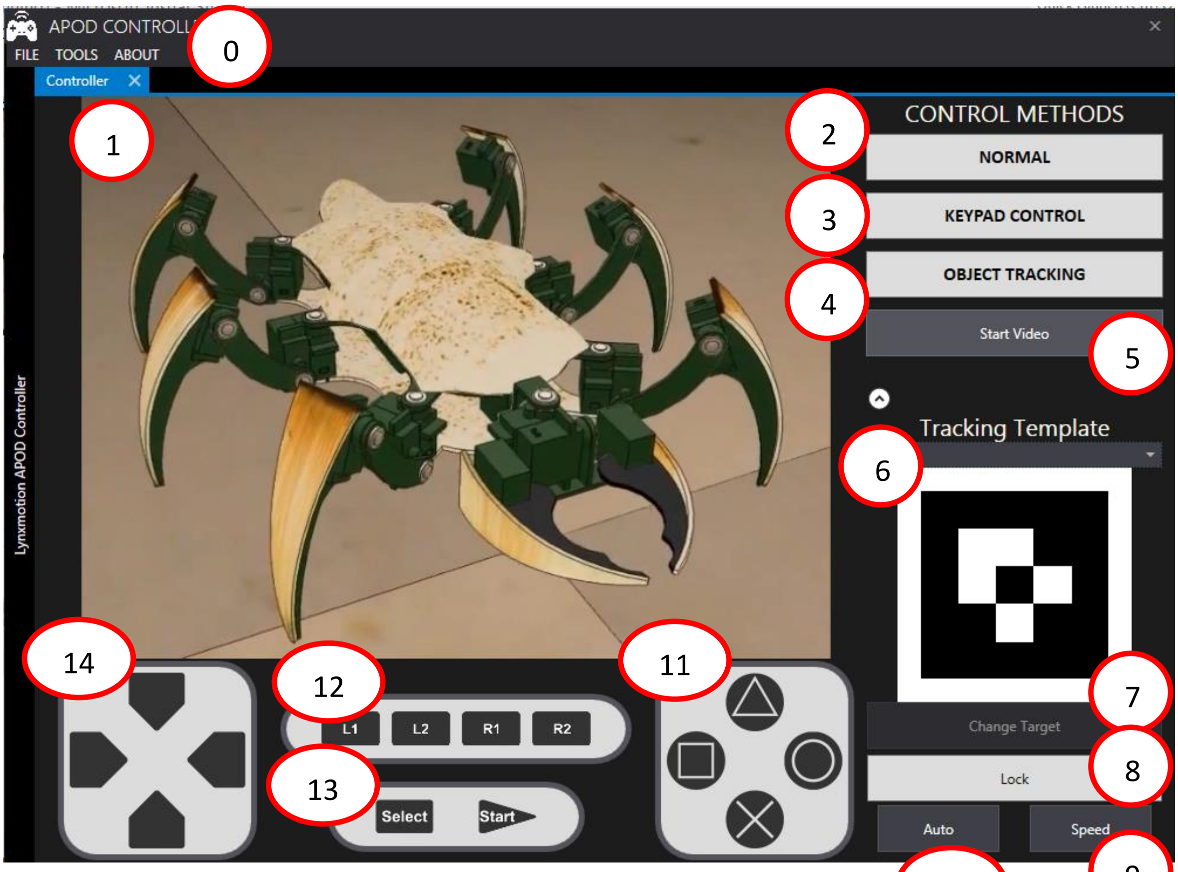


Figure D-70 Live Control Interface Design

Item NO.	Component Name	Description
0	Menu	Select Features
1	VideoSourcePlayer	Display Camera IP 's View
2	Checkbox (Button)	Select Normal input mode
3	Checkbox (Button)	Select Gamepad input mode
4	Checkbox (Button)	Select object tracking mode
5	Button	Start streaming video from Camera
6	Image	Display tracking template
7	Button	Change Tracking template
8	Checkbox (Button)	Lock target – Start tracking
9	Button	Switch speed
10	Button	Auto grip
11	Keypad	Action selection
12	Keypad	Action bank selection
13	Keypad	Set mode/function
14	Keypad	Navigation Selection

Table D-13 Live Control Interface Design Description

2. Sequence Player

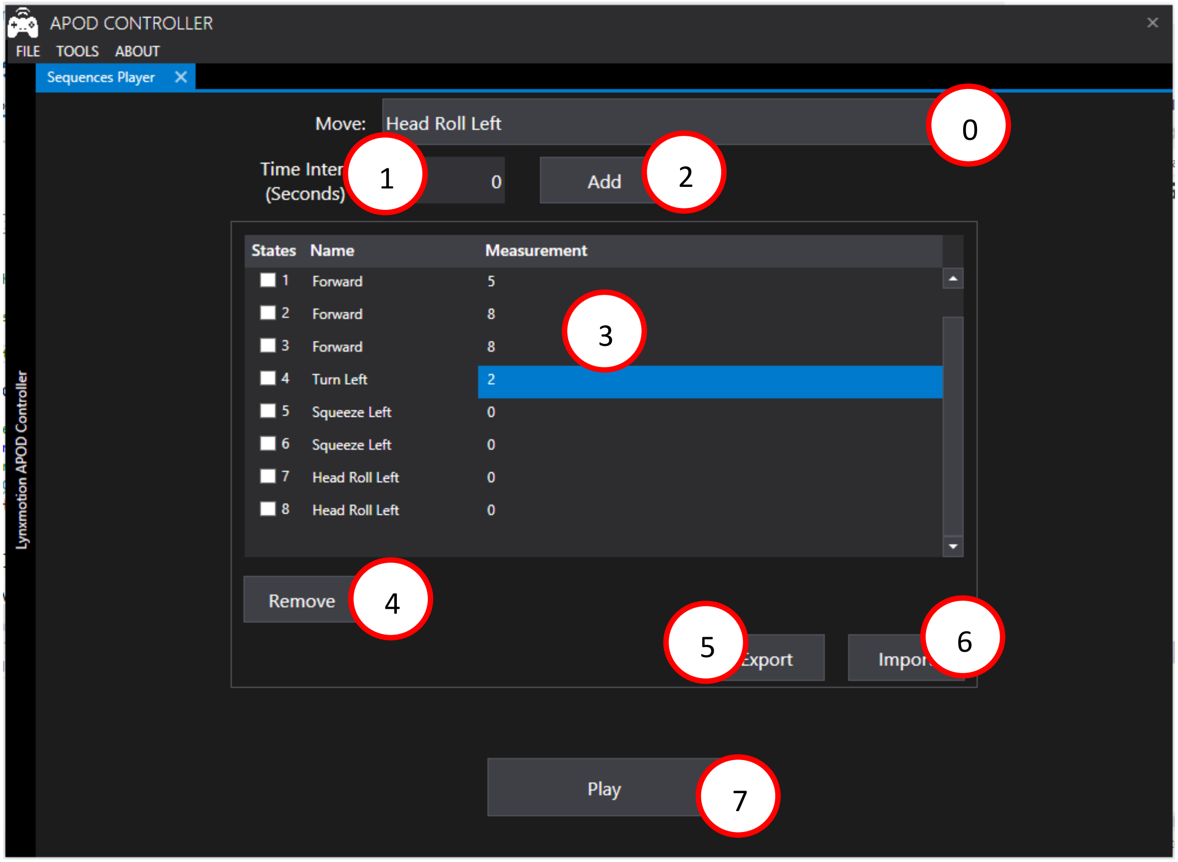


Figure D-71 Sequence Player Interface Design

Item NO.	Component Name	Description
0	Combo Box	Select action to add
1	Textbox	Movement interval
2	Button	Add action to sequence
3	Data Grid	Display current sequence content
4	Button	Remove selected states from sequence
5	Button	Export current sequence to file
6	Button	Import sequence from file
7	Button	Start playing the sequence

Table D-14 Sequence Player Interface Design Description

3. Configuration

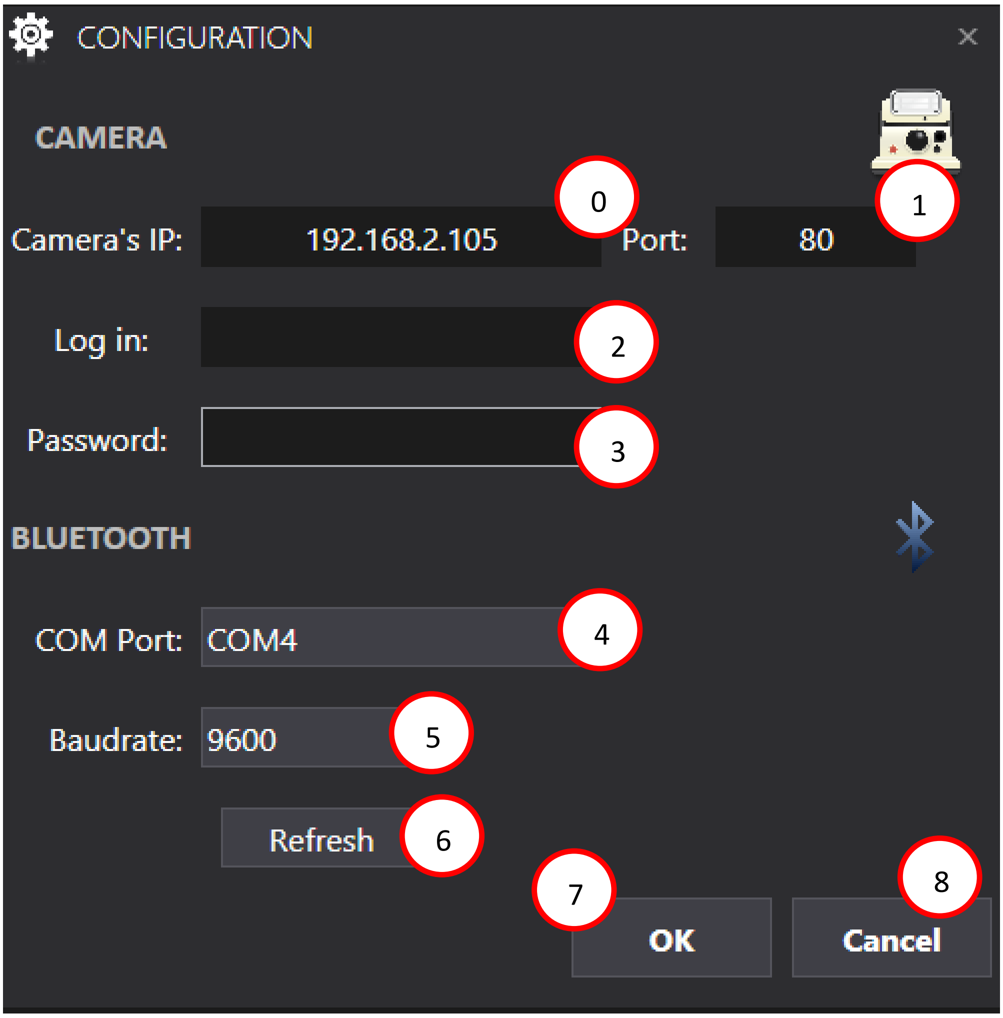


Figure D-72 Configuration Interface Design

Item NO.	Component Name	Description
0	Textbox	Enter IP of camera
1	Textbox	Camera port to get stream
2	Textbox	Username of camera IP
3	Password Box	Password of camera IP
4	Combo Box	All available COM ports
5	Combo Box	Bluetooth baud rate selection
6	Button	Refresh list of Ports
7	Button	Accept information
8	Button	Cancel action

Table D-15 Configuration Interface Design Description

4. Object Extractor

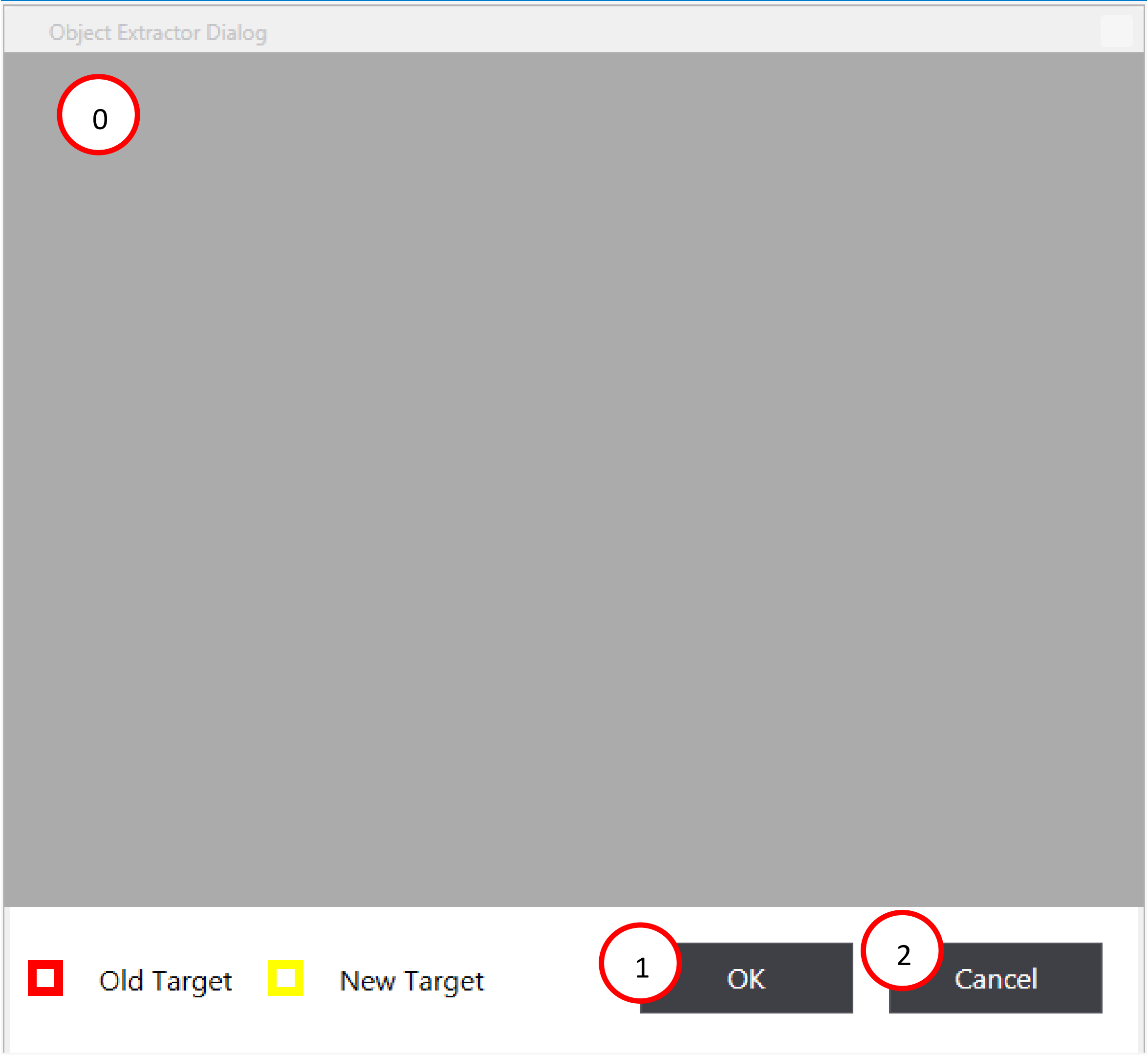


Figure D-73 Object Extractor Interface Design

Item NO.	Component Name	Description
0	Image	Capture frame for template selection
1	Button	Accept new template
2	Button	Cancel action

Table D-16 Object Extractor Interface Design Description

E. SOFTWARE TEST DOCUMENTATION (STD)

I. Introduction

1. Overview

The purpose of this project is to build a physical hexapod robotic with named APOD. He has the ability to get balance on his legs, move like a real insect, can record by the camera and auto follow and grip item which is chosen. In the other hand, the robot communicate with Pc through Bluetooth technology, and be Controlled by PS2 controller

2. Test Approach

White box testing : Developers self-test on code in which function they developed (unit test).

Black box testing : Test on each function of each module in system to ensure each module execute the right function. Then do integration testing to test the function. Finally, system test to test the whole system.

Testing Goal : Discover bug , fix code, regression test and finally completed system.

II. Test Plan

We have two main kind of test phases: Function test (Integration test) and System test.

For the Integration test, we get the functional test to show up the working of each function is right or wrong, so we have to focus on the result of the function in difference cases.

For the System test, all activities of system are needed to check after setup, for example what does he(APOD) do when we start up and observe him if something happen . We create test case to check each activity of him from turn on to turn off and log what he do when we act on. Finally, we record the result

1. Features to be tested

Functional Testing:

- Integration Testing: Test all function of robot, C# application
- System Testing: Test all function for APOD – C# application

UI Testing:

- Test UI of C# application

2. Features not to be tested

Embedded function on STM32F4 and SSC32

3. Test Environment

In addition to the application and any other user specified software, the following list of software should be considered a minimum:

- NET Framework 4.0
- Visual Studio 2012

Hardware tools:

- A PC/laptop that has Bluetooth device and can connect Wireless.
- Wireless Router

Operating System: Window 8

III. Test Cases

1. Functional Test

No	Function		Pre-condition	Step of Testing	Expected result	Create by	Result
	Medium Function	Small Function					
1	Start robot	Turn robot on	Robot is turned off	Switch the trigger on robot	Notice led on two board is on	QuanNM	P
2		Turn robot off	Robot is turned on	Switch the trigger on robot	Notice led on two board is off	QuanNM	P
3	Connect Bluetooth device	Connect PC/Laptop that has Bluetooth	Robot is turned on	1.Turn ON Bluetooth board on two device 2.Pair device and enter pin '1234' on PC	1. If two devices can connect to each other, the light of Bluetooth board on Robot is continuous light. If two devices can't connect to each other, the light of Bluetooth	QuanNM	

					board on Robot is flashing light.		
4	Operate in “Gamepad Controller” mode	Start robot	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8	Click “Start” button on PS2 controller	1.Robot start 2. All servos are set with 1500 pulse width	QuanNM	P
5		Stop robot	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Select] button on PS2 controller	1. Robot is stop. 2. All servos are released.	QuanNM	P
6		Robot move forward	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Up Dpad] button on PS2 controller	Robot move forward 1 time	QuanNM	P

7		Robot move backward	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Down Dpad] button on PS2 controller	Robot move backward 1 time	QuanNM	P
8		Turn robot left	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Left Dpad] button on PS2 controller	Robot turn left 1 time	QuanNM	P
9		Turn robot right	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Right Dpad] button on PS2 controller	Robot turn right 1 time	QuanNM	P
10		Drop robot’s body	1. Robot and PC/ Laptop connected with each other 2. Environment:	Click Combo button [Trigger L1] + [Trigger R1] on PS2 controller	Robot drop the height 1 time	QuanNM	P

			- Using Serial Port communication in win 7/8 3. Robot is on				
11		Lift Robot’s body	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Trigger R2] on PS2 controller	Robot lift the height 1 time	QuanNM	P
12		Stretch robot forward	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Up Dpad] on PS2 controller	Robot stretch forward 1 time	QuanNM	P
13		Stretch robot backward	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Down Dpad] on PS2 controller	Robot stretch backward 1 time	QuanNM	P

14		Stretch robot left	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Left Dpad] on PS2 controller	Robot stretch to left 1 time	QuanNM	P
15		Stretch robot right	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Right Dpad] on PS2 controller	Robot stretch to right 1 time	QuanNM	P
16		Rotate Robot's neck to left	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Square] on PS2 controller	Robot's neck rotate to left 1 time	QuanNM	P
17		Rotate Robot's neck to right	1. Robot and PC/ Laptop connected with each other 2. Environment:	Click [Circle] on PS2 controller	Robot's neck rotate to right 1 time	QuanNM	P

			- Using Serial Port communication in win 7/8 3. Robot is on				
18		Turn robot's neck up	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L2] + [Up Dpad] on PS2 controller	Robot's neck turn up 1 time	QuanNM	P
19		Turn robot's neck down	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click combo button [Trigger L2] + [Down Dpad] on PS2 controller	Robot's neck turn down 1 time	QuanNM	P
20		Turn robot's neck left	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click combo button [Trigger L2] + [Left Dpad] on PS2 controller	Robot's neck turn left 1 time	QuanNM	P

21		Grip robot's ripper	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Triangle] button on PS2 controller	Robot's Gripper grip 1 time	QuanNM	P
22		Release robot's ripper	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click combo button [Trigger L2] + [Left Dpad] on PS2 controller	Robot's Gripper release 1 time	QuanNM	p
23	Operate in "Normal" mode (using keyboard)	Start robot	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8	Click "Start" button on Main PC Interface	1.Robot start 2. All servos are set with 1500 pulse width	QuanNM	P
24		Stop robot	1. Robot and PC/ Laptop connected with each other 2. Environment:	Click [Select] button on Main PC Interface	1. Robot is stop. 2. All servos are released.	QuanNM	P

			- Using Serial Port communication in win 7/8 3. Robot is on				
25		Robot move forward	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Up Dpad] button on Main PC Interface	Robot move forward 1 time	QuanNM	P
26		Robot move backward	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Down Dpad] button Main PC Interface	Robot move backward 1 time	QuanNM	P
27		Turn robot left	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Left Dpad] button Main PC Interface	Robot turn left 1 time	QuanNM	P

28		Turn robot right	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is turned on	Click [Right Dpad] button Main PC Interface	Robot turn right 1 time	QuanNM	p
29		Drop robot's body	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Trigger R1] Main PC Interface	Robot drop the height 1 time	QuanNM	P
30		Lift Robot's body	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Trigger R2] Main PC Interface	Robot lift the height 1 time	QuanNM	P

31		Stretch robot forward	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Up Dpad] Main PC Interface	Robot stretch forward 1 time	QuanNM	P
32		Stretch robot backward	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Down Dpad] Main PC Interface	Robot stretch backward 1 time	QuanNM	P
33		Stretch robot left	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L1] + [Left Dpad] on Main PC Interface	Robot stretch to left 1 time	QuanNM	P
34		Stretch robot right	1. Robot and PC/ Laptop connected with each other 2. Environment:	Click Combo button [Trigger L1] + [Right Dpad] on Main PC Interface	Robot stretch to right 1 time	QuanNM	P

			- Using Serial Port communication in win 7/8 3. Robot is on				
35		Rotate Robot's neck to left	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Square] on Main PC Interface	Robot's neck rotate to left 1 time	QuanNM	P
36		Rotate Robot's neck to right	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Circle] on Main PC Interface	Robot's neck rotate to right 1 time	QuanNM	P
37		Turn robot's neck up	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click Combo button [Trigger L2] + [Up Dpad] on Main PC Interface	Robot's neck turn up 1 time	QuanNM	P

38		Turn robot's neck down	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click combo button [Trigger L2] + [Down Dpad] on Main PC Interface	Robot's neck turn down 1 time	QuanNM	P
39		Turn robot's neck left	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click combo button [Trigger L2] + [Left Dpad] on Main PC Interface	Robot's neck turn left 1 time	QuanNM	P
40		Grip robot's ripper	1. Robot and PC/ Laptop connected with each other 2. Environment: - Using Serial Port communication in win 7/8 3. Robot is on	Click [Triangle] button on Main PC Interface	Robot's Gripper grip 1 time	QuanNM	P
41		Release robot's ripper	1. Robot and PC/ Laptop connected with each other 2. Environment:	Click combo button [Trigger L2] + [Left Dpad] on Main PC Interface	Robot's Gripper release 1 time	QuanNM	P

			- Using Serial Port communication in win 7/8 3. Robot is on				

Table E-1 Functional Test Cases

2. UI Test

No	Function		Pre-condition	Step of Testing	Expected result	Create by	Result
	Medium Function	Small Function					
1	Normal mode	Connect Bluetooth with normal controller Connect Bluetooth with normal controller	1.Application is open. 2.Bluetooth is already to pair 3.Wireless connected	1. Click “Normal” button. 2.In Configuration, Fill in: - IP camera:”192.168.2.100” - Port: “80” - Admin: “admin” - Password: NONE - Com port: COM3 (depend on each computer) -Baud rate: “9600” 3. Click OK	- Configuration windows is disappear. -IP camera connected - Bluetooth pair with robot	QuanNM	P
2			1.Application is open. 2.Bluetooth is already to pair 3.Wireless connected	1. Click “Normal” button. 2.In Configuration, Fill in: - IP camera: NONE - Port: NONE - Admin: “admin” - Password: NONE - Com port: NONE(depend on each computer) -Baud rate: “9600” 3. Click OK	1.Each blank detail(IP camera, Port, Com port) will noticed by change the color (red) 2. System still not connect	QuanNM	P
3			1.Application is open. 2.Bluetooth is already to pair	1. Click “Normal” button. 2.In Configuration, Fill in: - IP camera: NONE - Port: 80 - Admin: “admin”	1. IP camera is noticed to red color. 2.System still not connect	QuanNM	p

			3.Wireless connected	- Password: NONE - Com port: COM No.(depend on each computer) -Baud rate: "9600" 3. Click OK			
4			1.Application is open. 2.Bluetooth is already to pair 3.Wireless connected	1. Click "Normal" button. 2.In Configuration, Fill in: - IP camera: "192.168.2.100" - Port: NONE - Admin: "admin" - Password: NONE - Com port: COM No.(depend on each computer) -Baud rate: "9600" 3. Click OK	1. Port is noticed to red color. 2.System still not connect	QuanNM	P
5			1.Application is open. 2.Bluetooth is already to pair 3.Wireless connected	1. Click "Normal" button. 2.In Configuration, Fill in: - IP camera: "192.168.2.100" - Port: 80 - Admin: "admin" - Password: NONE - Com port: NONE No.(depend on each computer) -Baud rate: "9600" 3. Click OK	1. COM Port is noticed to red color. 2.System still not connect	QuanNM	P
6			1.Application is open. 2.Bluetooth is already to pair	1. Click "Normal" button. 2.In Configuration, Fill in: - IP camera:"192.168.2.100" - Port: "80"	Configuration windows is disappear.	QuanNM	p

			3.Wireless connected	- Admin: “admin” - Password: NONE - Com port: COM3 (depend on each computer) -Baud rate: “9600” 3. Click OK			
7		Connect Bluetooth with normal controller Connect Bluetooth with Keypad controller	1.Application is open. 2.Bluetooth is already to pair 3.Wireless connected	1. Click “Normal” button. 2.In Configuration, Fill in: - IP camera:”192.168.2.100” - Port: “80” - Admin: “admin” - Password: NONE - Com port: COM3 (depend on each computer) -Baud rate: “9600” 3. Click OK	- Configuration windows is disappear. -IP camera connected - Bluetooth pair with robot	QuanNM	P
8			1.Application is open. 2.Bluetooth is already to pair 3.Wireless connected	1. Click “Normal” button. 2.In Configuration, Fill in: - IP camera: NONE - Port: NONE - Admin: “admin” - Password: NONE - Com port: NONE(depend on each computer) -Baud rate: “9600” 3. Click OK	1.Each blank detail(Ip camera, Port, Com port) will noticed by change the color (red) 2. System still not connect	QuanNM	P
9			1.Application is open.	1. Click “Normal” button. 2.In Configuration, Fill in: - IP camera: NONE	1. IP camera is noticed to red color.	QuanNM	p

			2.Bluetooth is already to pair 3.Wireless connected	- Port: 80 - Admin: “admin” - Password: NONE - Com port: COM No.(depend on each computer) -Baud rate: “9600” 3. Click OK	2.System still not connect		
10			1.Application is open. 2.Bluetooth is already to pair 3.Wireless connected	1. Click “Normal” button. 2.In Configuration, Fill in: - IP camera: “192.168.2.100” - Port: NONE - Admin: “admin” - Password: NONE - Com port: COM No.(depend on each computer) -Baud rate: “9600” 3. Click OK	1. Port is noticed to red color. 2.System still not connect	QuanNM	P
11			1.Application is open. 2.Bluetooth is already to pair 3.Wireless connected	1. Click “Normal” button. 2.In Configuration, Fill in: - IP camera: “192.168.2.100” - Port: 80 - Admin: “admin” - Password: NONE - Com port: NONE No.(depend on each computer) -Baud rate: “9600” 3. Click OK	1. COM Port is noticed to red color. 2.System still not connect	QuanNM	P

12		Start camera	1.Bluetooth is already connected. 2.Wireless is already connected. 3.Camera is already connected.	Click “Start Video” button	Camera screen appear in camera view frame	QuanNM	P
13	Object Tracking	Getting item	1.Bluetooth is already connected. 2.Wireless is already connected. 3.Camera is already connected. 4.Camera screen started	1. Click “Object tracking” button. 2. Click on item’s color 3. Click Ok	1. Yellow square focus on chosen item. 2. On the right screen there is a show of item’s color	QuanNM	P
14		Follow particular Item	1.Bluetooth is already connected. 2.Wireless is already connected. 3.Camera is already connected.	Click “Auto” button	1.On APOD: - Gripper is open widely. - APOD auto moves to item. - After he reach the item, then he nips. 2. On Pc:	QuanNM	P

			4.Camera screen started 5. Item is tracked		- Tracking zone is always focus depend on APOD movement		
15		Get distance	1.Bluetooth is already connected. 2.Wireless is already connected. 3.Camera is already connected. 4.Camera screen started 5. Item is tracked	Click “Distance” button	Receive current distance from APOD to item.	QuanNM	P
16	Sequence Player	Making sequence to perform(1)	1.Bluetooth is already connected. 2.Wireless is already connected.	1.Click menu “Tool”, choose “Sequence player” menu. 2. In move: chose [Forward]. 3. In Time: Enter [5]. 4. click Add. 5.click Play.	APOD move forward in 5 second	QuanNM	P
17		Making sequence to perform(2)	1.Bluetooth is already connected. 2.Wireless is already connected.	1.Click menu “Tool”, choose “Sequence player” menu. 2. In move: chose [Forward]. 3. In Time: Enter [5]. 4. click Add. 5. In move: chose [Turn Left].	APOD move forward in 5 seconds, turn left in 3 seconds	QuanNM	P

				6. In Time: Enter [3]. 7.click Play.			
18		Making sequence to perform(3)	1.Bluetooth is already connected. 2.Wireless is already connected.	1.Click menu “Tool”, choose “Sequence player” menu. 2. In move: chose [Forward]. 3. In Time: Enter [5]. 4. click Add. 5. In move: chose [Turn Left]. 6. In Time: Enter [3]. 7. In move: chose [Turn Right]. 8. In Time: Enter [10]. 9.click Play.	APOD move forward in 5 seconds, turn left in 3 seconds, turn right in 10 seconds	QuanNM	P
19		Making sequence to perform(4)	1.Bluetooth is already connected. 2.Wireless is already connected.	1.Click menu “Tool”, choose “Sequence player” menu. 2. In move: chose [Backward]. 3. In Time: Enter [5]. 4. click Add. 5. In move: chose [Head left]. 7. In move: chose [toward Front]. 8.click Play.	APOD move forward in 5 seconds, Head to left, APOD is toward front	QuanNM	P
20		Making sequence to perform(4)	1.Bluetooth is already connected. 2.Wireless is already connected.	1.Click menu “Tool”, choose “Sequence player” menu. 2. In move: chose [Forward]. 3. In Time: Enter [-5].	System prevent to enter “-”	QuanNM	P

21		Export sequence	1.Bluetooth is already connected. 2.Wireless is already connected.	1.Click menu “Tool”, choose “Sequence player” menu. 2. In move: chose [Backward]. 3. In Time: Enter [5]. 4. click Add. 5. In move: chose [Head left]. 7. In move: chose [toward Front]. 8.click “Export” button. 9.Choose direction. 10. Enter name of file 11. click OK	Sequence file is created with xml type	QuanNM	P
22		Import sequence	1.Bluetooth is already connected. 2.Wireless is already connected.	1.Click Import button 2.chose file xml to import. 3.Click OK.	List of sequence in file are listed to perform	QuanNM	P
23		Remove sequence	1.Bluetooth is already connected. 2.Wireless is already connected.	1.Click menu “Tool”, choose “Sequence player” menu. 2. In move: chose [Backward]. 3. In Time: Enter [5]. 4. click Add. 5. In move: chose [Head left]. 7. In move: chose [toward Front]. 8. stick on [Head Left] combo box 8.click Remove button.	[Head left] is removed in list	QuanNM	p

Table E-2 UI Test Cases

F. SOFTWARE USER MANUAL (SUM)

I. Installation

Following below figures step-by-step for the installation of APOD Controller Software

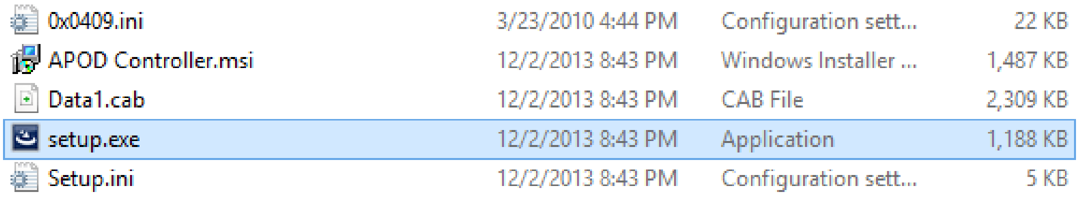


Figure F-1 Run Setup file in delivered package

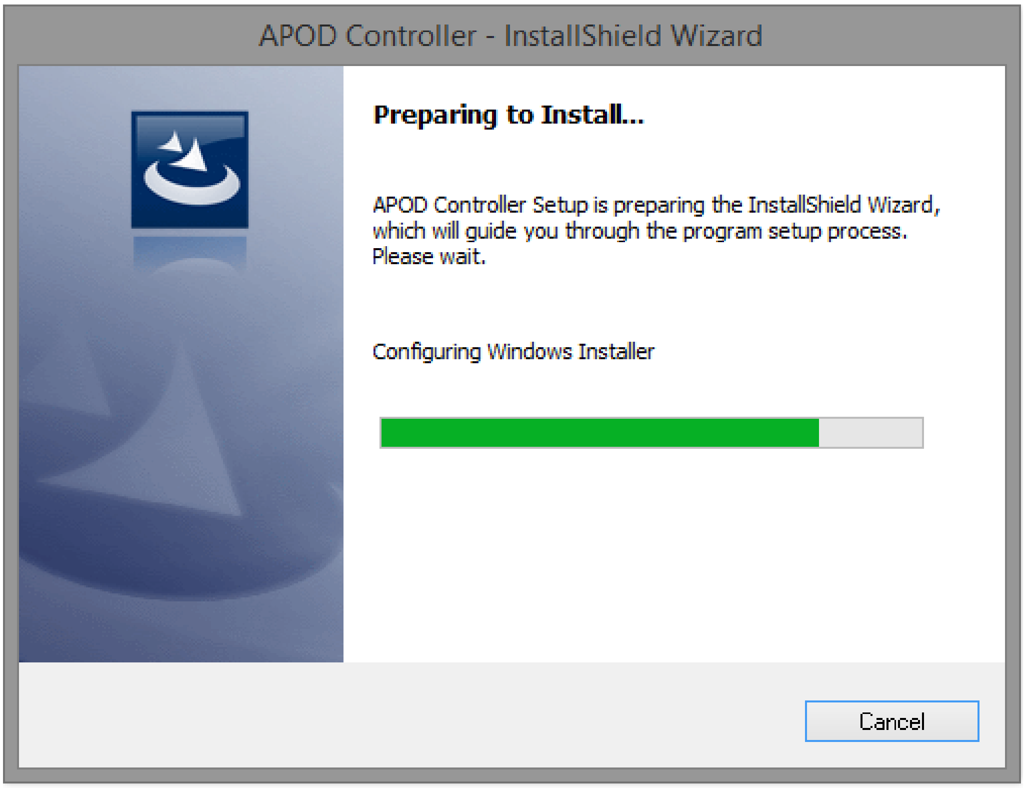


Figure F-2 Waiting for Install Shield to configure Window Installer

Figure F-3 Click Next on Welcome dialog

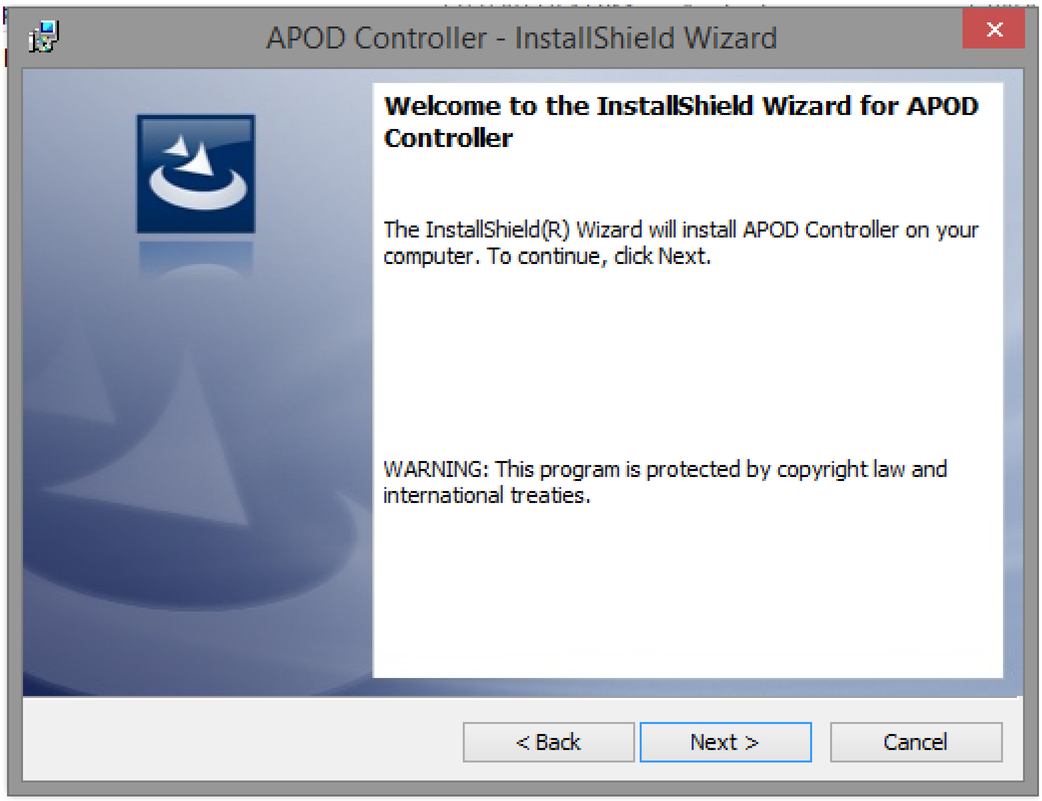


Figure F-4 Click Next to start.

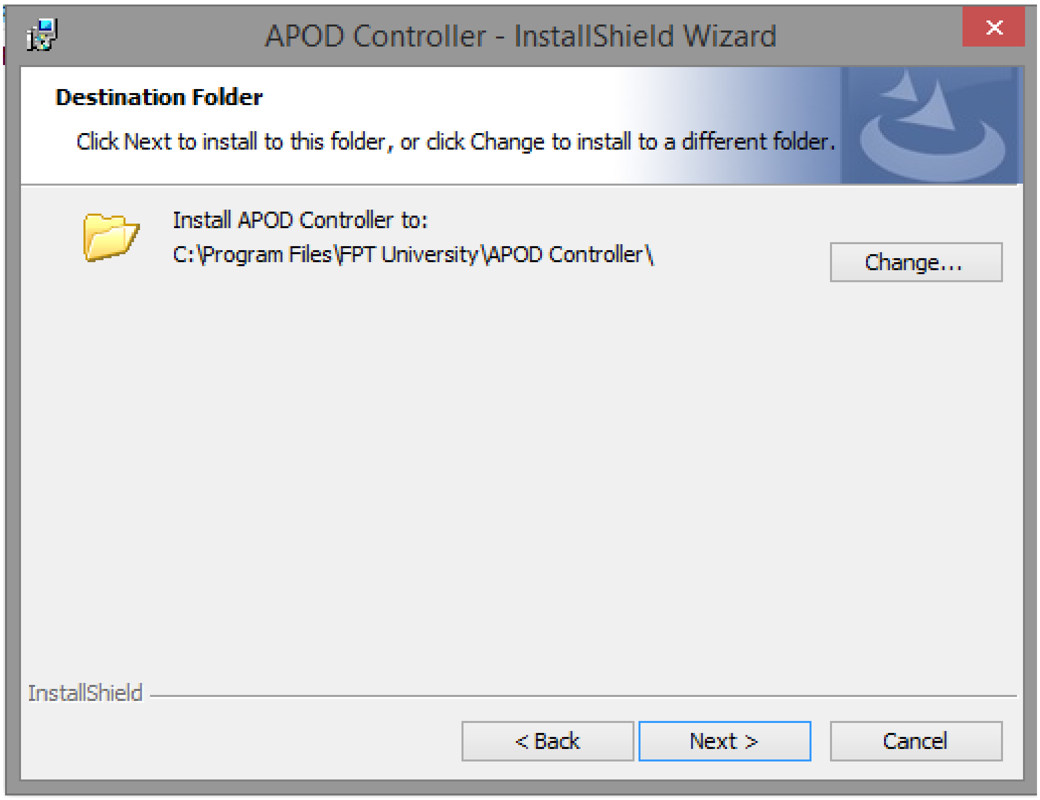


Figure F-5 Choose the location of new program

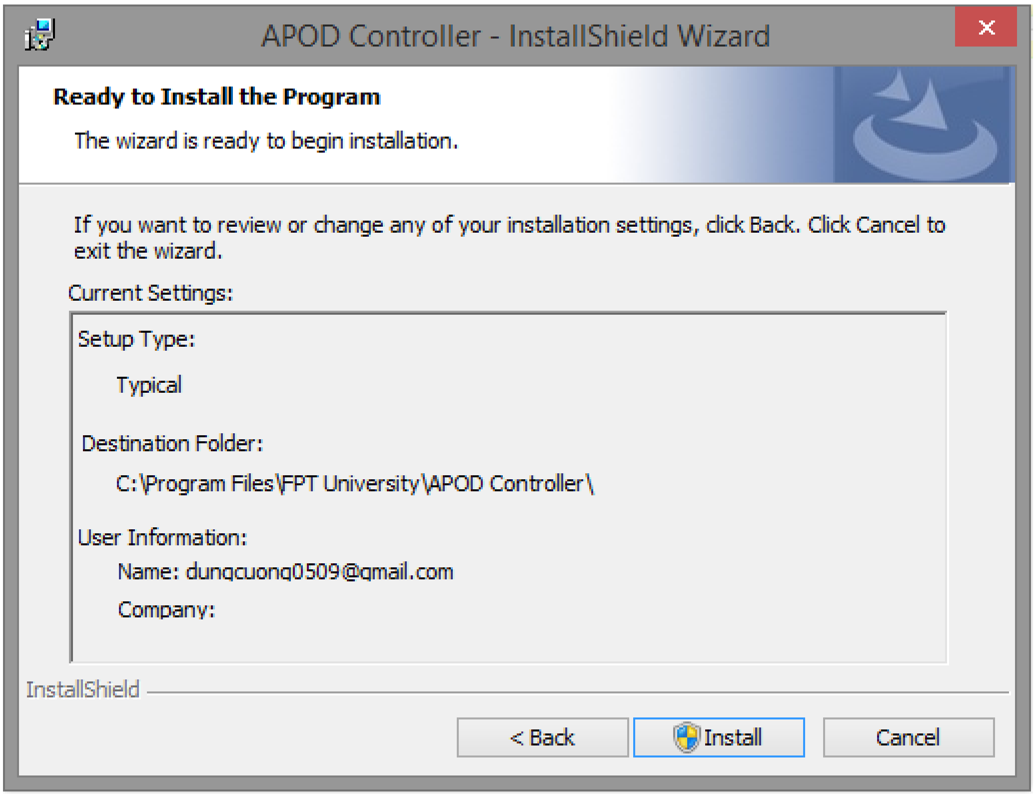


Figure F-6 Re-check setup information

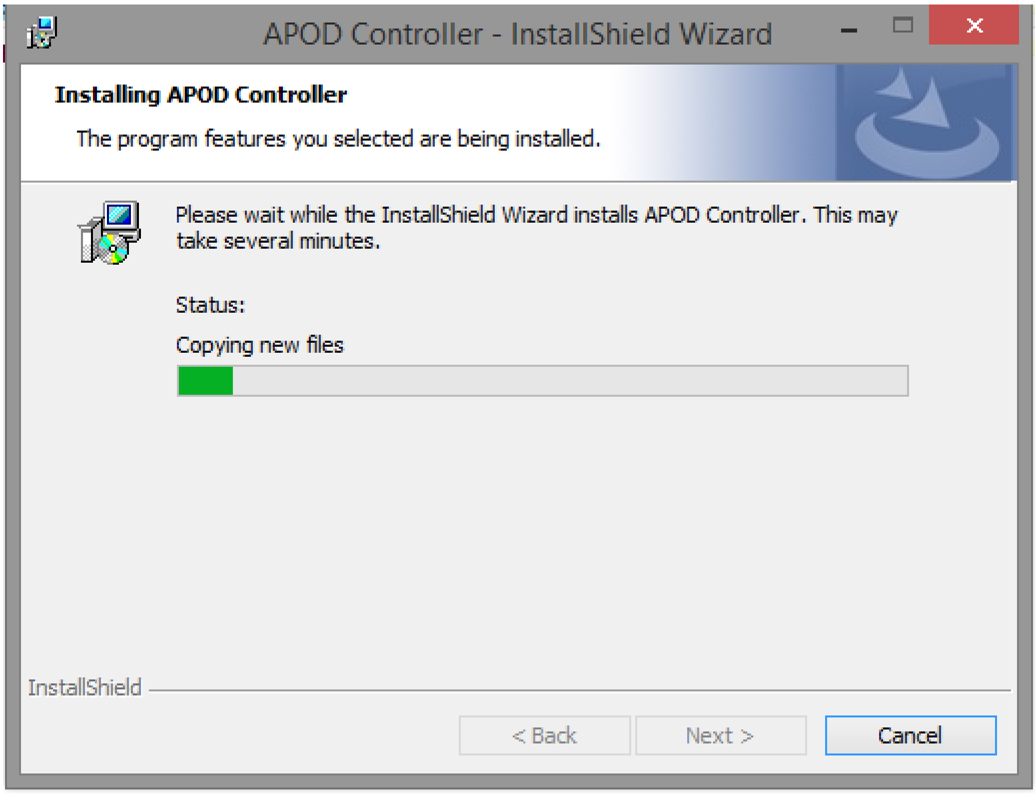


Figure F-7 Waiting for the installation to finish

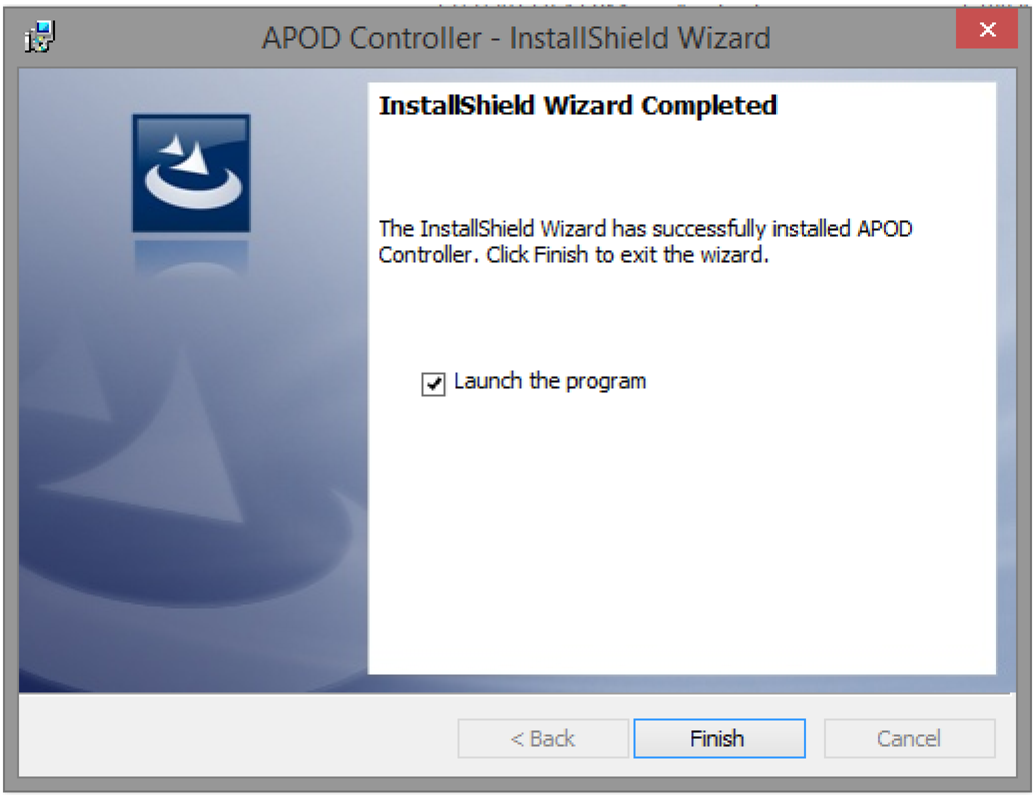


Figure F-8 Finish the installation

II. Removal

To remove the program:

- Go to Control Panel -> Programs -> Programs and Features
- In the dialog, Select the "APOD Controller"
- Click Uninstall.

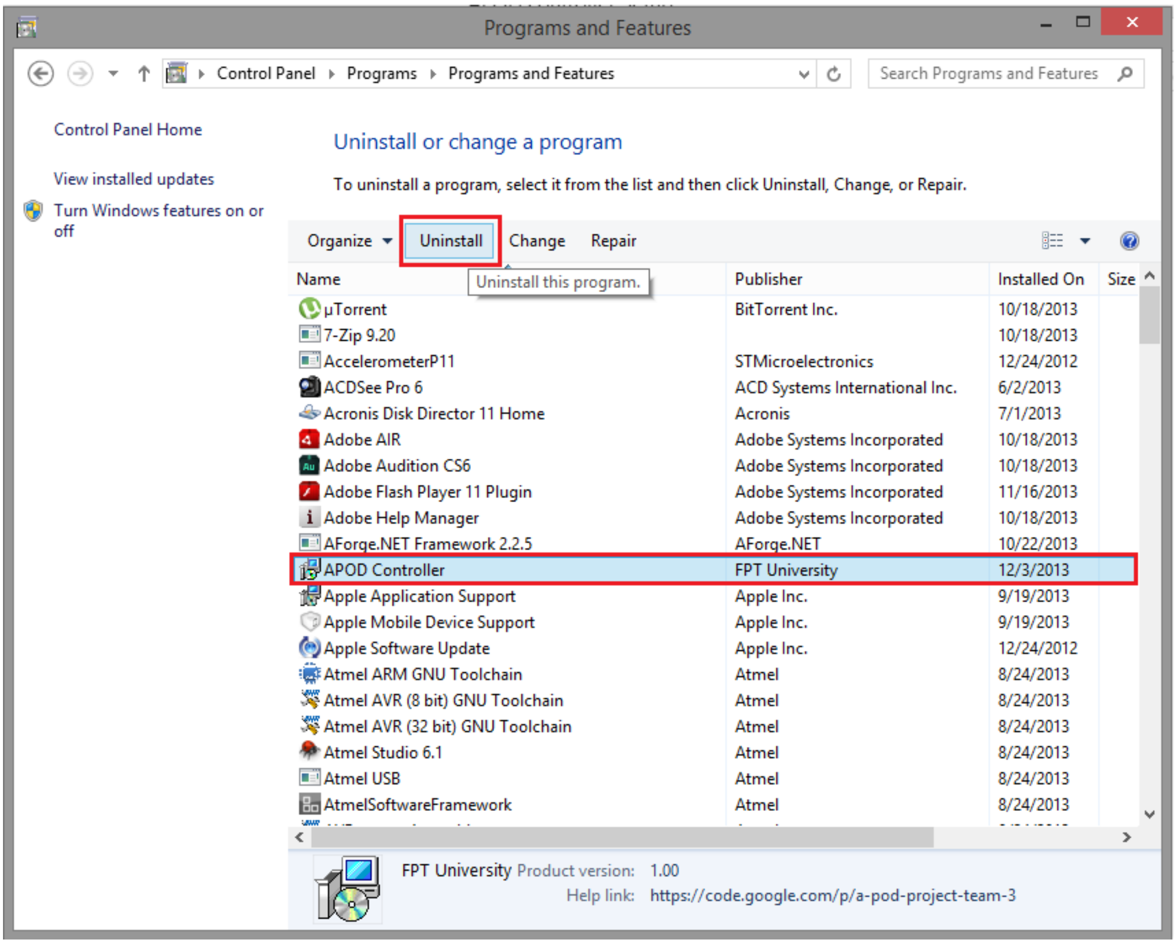


Figure F-9 Remove

III. User Guide

1. *Configuration*

1.1. *How to access?*

Before using any feature of the software the configuration is needed for setting up connection between user PC and APOD.

To access the configuration Dialog click on menu Tool on the top left corner and select Configuration.

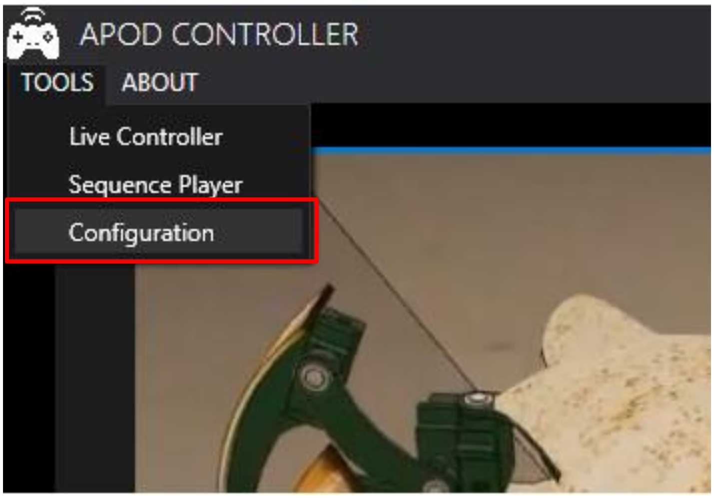


Figure F-10 Menu Configuration

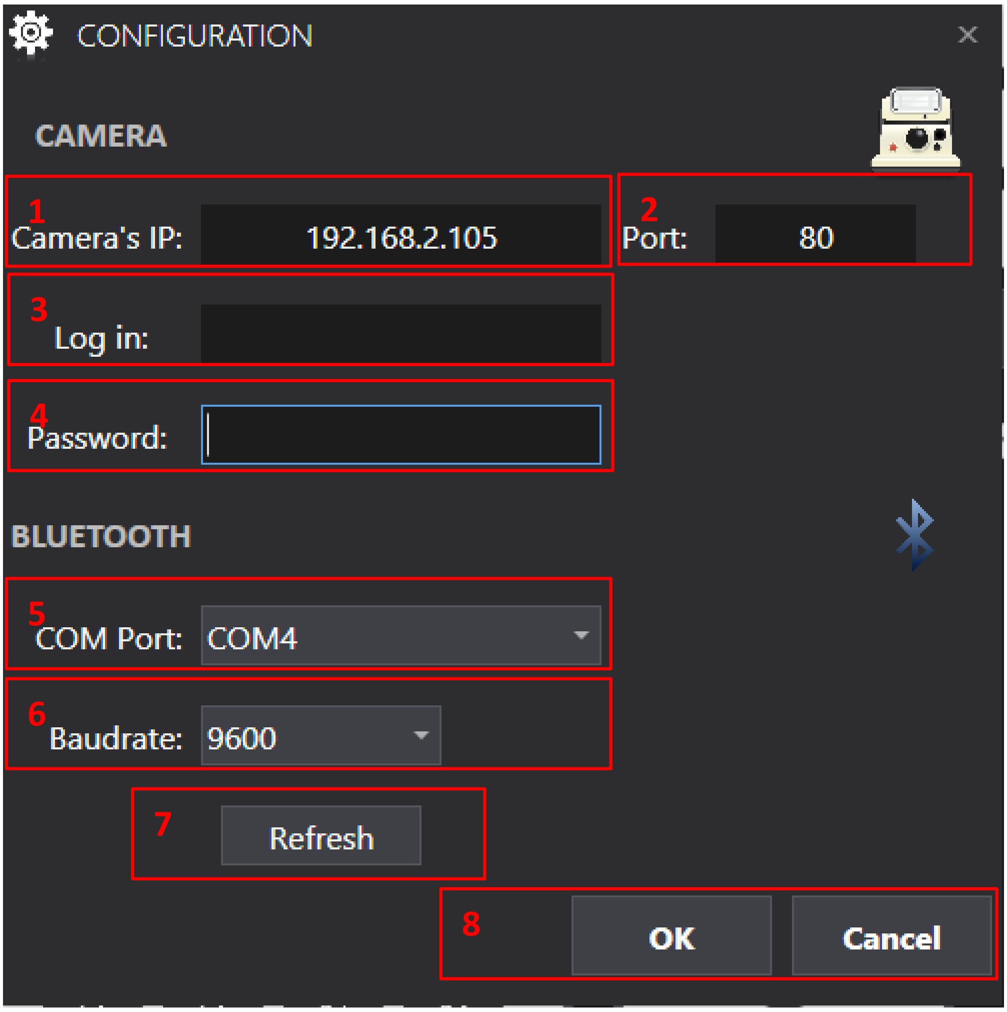


Figure F-11 Configuration

Section NO.	Description
1	This field contain the IP of Camera in your local LAN. Make sure the IP is corrected or the broadcast won't be able to perform. Access router for the exact IP of Camera.
2	This is the port number to access camera video stream. This port is fixed from the Camera's configuration itself (default is 80). In case camera's configuration is changed, you must enter the new port here.
3	Camera log in user name. Check the camera configuration for details
4	Camera log in password. Check the camera configuration for details
5	This is the communication COM port (*) for Bluetooth connection. Select the right port used by Bluetooth or the connection will fail.
6	COM port baud rate.
7	Get the new list of available COM port.
8	Accept or reject the configuration. If you click OK, the connection will be establish and there will be a message if connection fail (wrong configuration)

Table F-1 Configuration Description

1.2. How to know Bluetooth Port?

This method describe below is applied for those using Broadcom Bluetooth device driver. Different manufacturers may have different methods to recognize Bluetooth COM port.

On the bottom right corner of your screen. Find the Bluetooth Icon, right click on it and select “Open Settings”

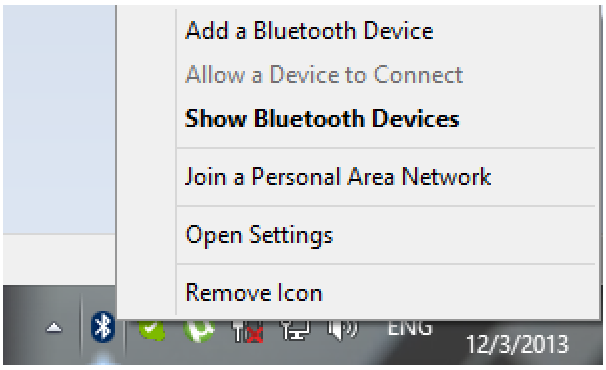


Figure F-12 Open Bluetooth setting

When the Bluetooth Settings dialog appear, switch to the “COM ports” tab.

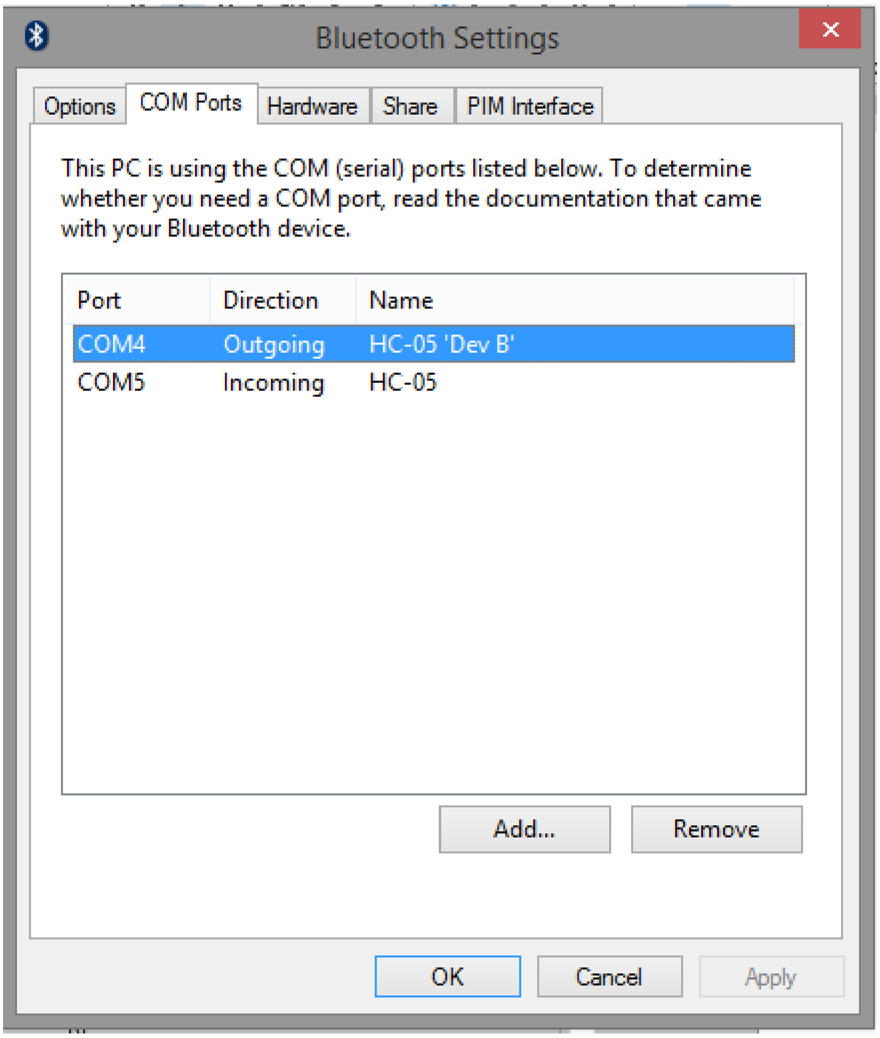


Figure F-13 Bluetooth Settings dialog

As you can see, there a list of COM port using for the Bluetooth, the right port for selecting in the Configuration is the one with “Outgoing” direction. In this case, the right COM port is COM4.

2. *Live Control*

2.1. *How to access?*

On the top left menu, click on menu “TOOL” and select “Live Controller”.

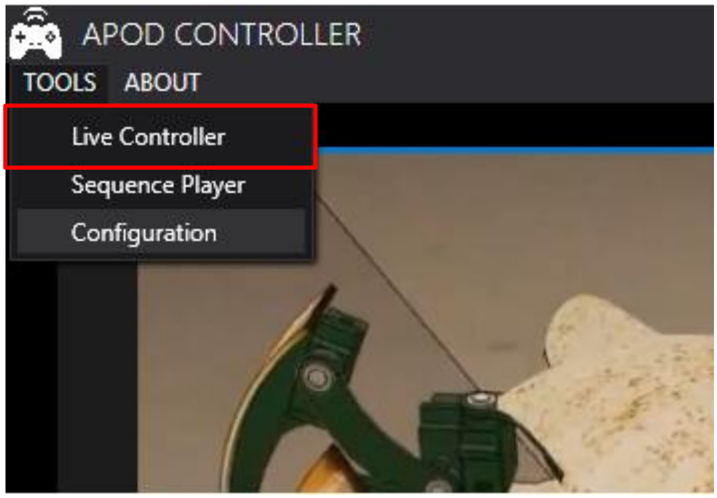


Figure F-14 Access Live Controller

2.2. How to Control?

This is the main view of live control tab.

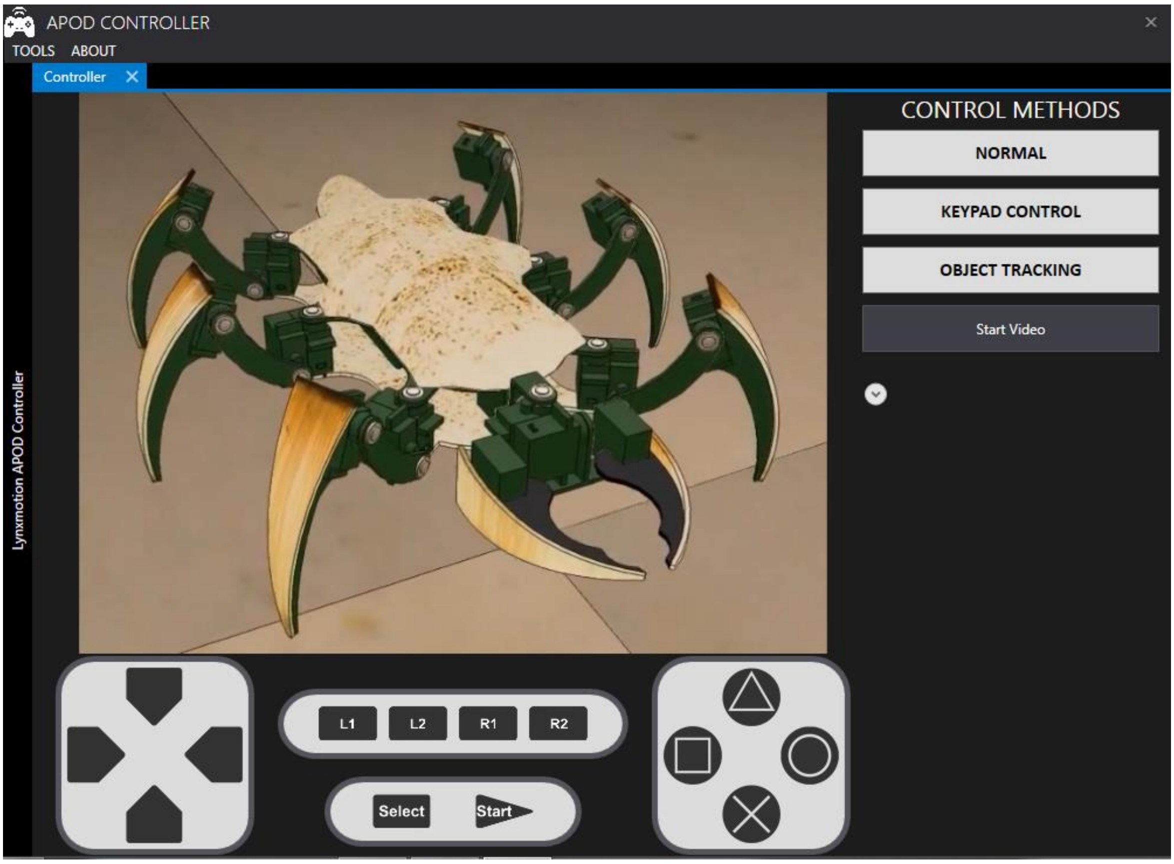


Figure F-15 Live Control Main View

The view of control button is divided into 4 group:

- Navigation: Up ( ), Down ( ), Left ( ), Right ( )
- Action: Triangle ( ), Cross ( ), Square ( ), Circle ( )

- Bank: L1 (), L2 (), R1 (), R2 ()
- Mode: Select (), Start ()

The mission of each button in group will be shown as below table.

L1 - L2 (*)	00	10	01
Up	Forward	Toward Front	Head up
Down	Backward	Toward Back	Head Down
Left	Turn Left	Squeeze Left	Head Left
Right	Turn Right	Squeeze Right	Head Right
Triangle	Mandibles Grip	--	--
Cross	Mandibles Release	--	--
Square	--	--	Neck Roll Left
Circle	--	--	Neck Roll Right
R1	--	--	--
R2	--	--	--
Start	Start Up	Body Lift	--
Select	Stop	Body Drop	--

Table F-2 Button Command

Notes: (*) “1” indicate “pressed” and “0” indicate release.

Depend on the selected mode, the input will be mapped to the control buttons.

2.3. Normal Input

2.3.1 Access

To select the Normal input mode, click on the Normal button on the right side. If no configuration was made, you will be prompted to make the configuration (refer to the Configuration section for details).

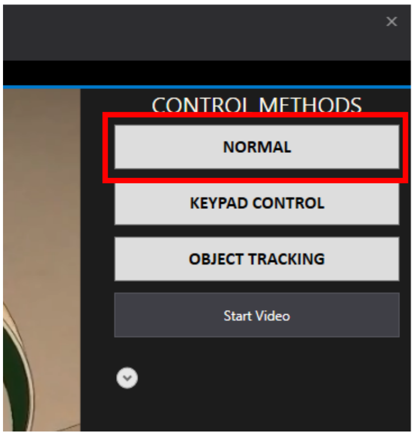


Figure F-16 Select Normal input

2.3.2 Control

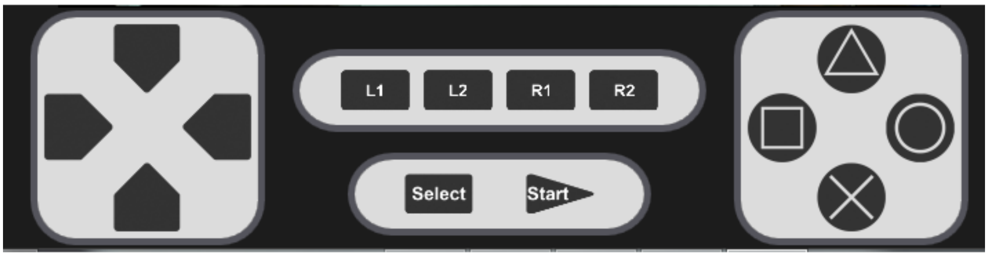


Figure F-17 Control Area using Normal Input

Button	Input (keyboard)	Alternate Input
Up	w	Click on
Down	S	Click on
Left	A	Click on
Right	D	Click on
Triangle	I	Click on
Cross	K	Click on
Square	J	Click on
Circle	L	Click on
L1	Q	Click on L1
L2	E	Click on L2
R1	U	Click on R1
R2	O	Click on R2
Start	H	Click on Start
Select	G	Click on Select

Table F-3 Normal Input Control Map

2.4. Gamepad Input

2.4.1 Access

To select the Normal input mode, click on the Normal button on the right side. If no configuration was made, you will be prompted to make the configuration (refer to the Configuration section for details).

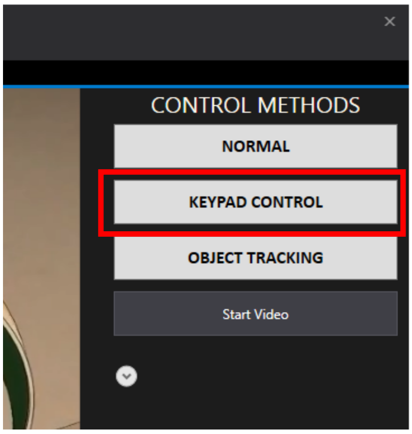


Figure F-18 Select Gamepad Input

2.4.2 Control

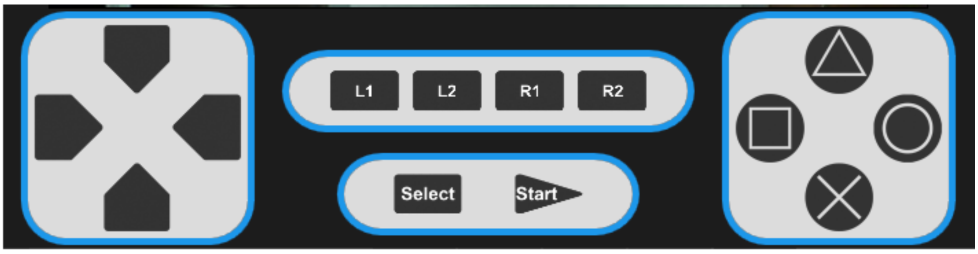


Figure F-19 Control Area using Gamepad Input

Button	Input on keypad	Alternate Input
Up		N/A
Down		N/A
Left		N/A
Right		N/A
Triangle	Button 1 or	N/A
Cross	Button 3 or	N/A
Square	Button 4 or	N/A
Circle	Button 2 or	N/A
L1	Button 7 or L1	N/A
L2	Button 5 or L2	N/A
R1	Button 8 or R1	N/A
R2	Button 6 or R2	N/A
Start	Button 10 or Start	N/A
Select	Button 9 or Select	N/A

Table F-4 Gamepad Input Control Map

2.5. Start Camera

To start video streaming, click on the "Start Video" button on the right side. If no configuration was made, you will be prompted to make the configuration (refer to the Configuration section for details).

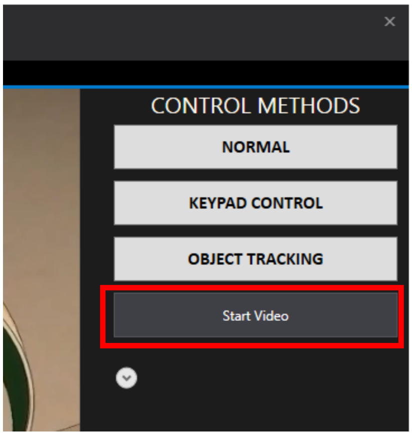


Figure F-20 Start Video

After starting camera view, the main window will be updated with the view.

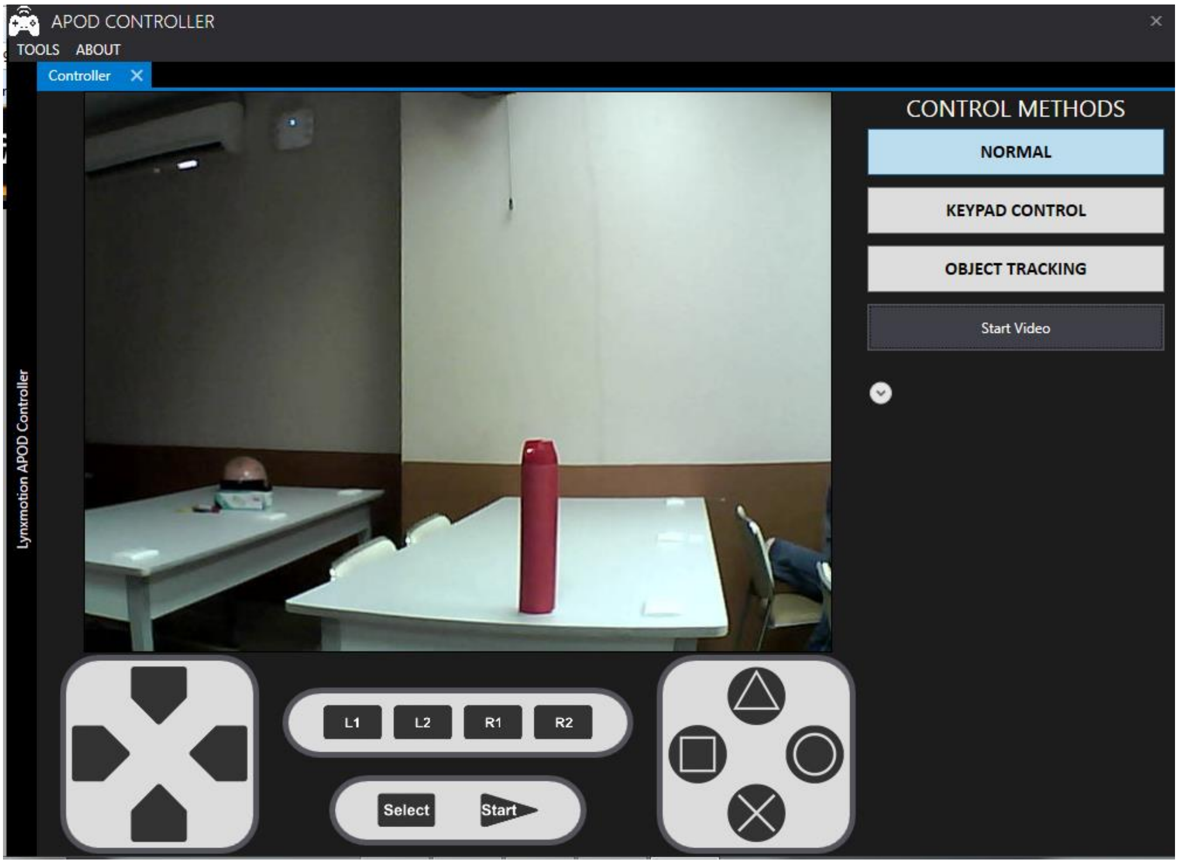


Figure F-21 Camera View

2.6. Object Tracking

2.6.1 Choose Target

The Object Tracking mode will ask you to choose the tracking object for the first time you select.

First step is selected the mode: click on the “Object Tracking” button on the right side.

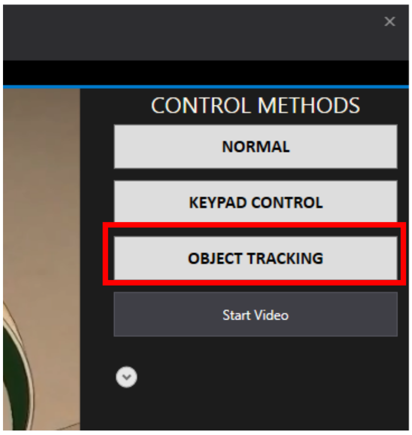


Figure F-22 Select Object Tracking

If the video stream wasn’t started, you will be prompted an alert. The video stream must be started before using Object Tracking features.

Once you select the mode, a dialog will appear for you to select object. Note that the feature only works for object with solid colors.

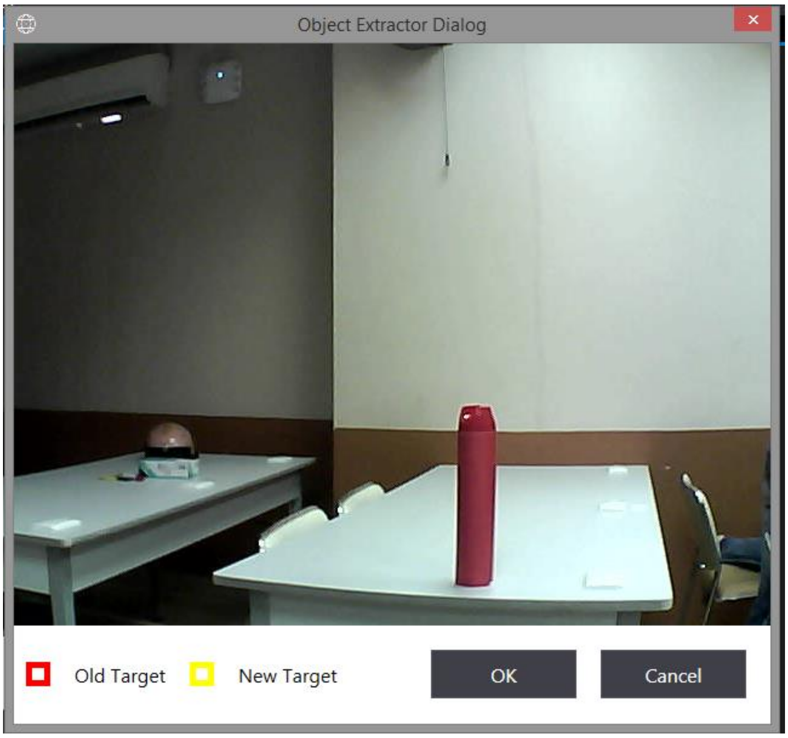


Figure F-23 Object Extractor dialog

Click and drag the mouse on the solid-color area of the target (Tracking template) you want to track.

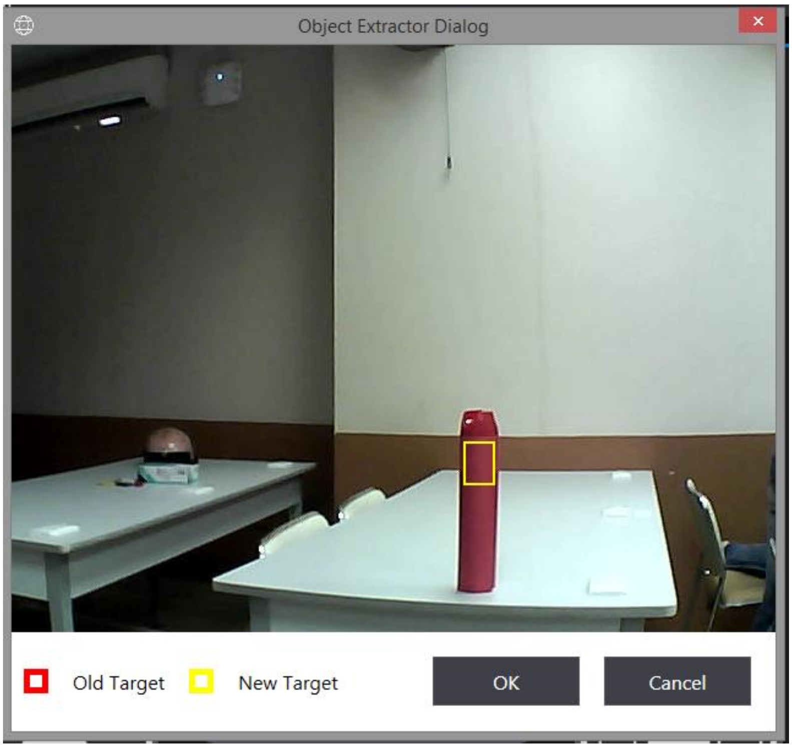


Figure F-24 Select Template

The selected template will be surround by a Yellow rectangle. If you wish to change template, repeat from the “click and drag” step to select new are.

Click “ok” to accept the template.

When return, the camera view will mark the object with a red rectangle as below figure:

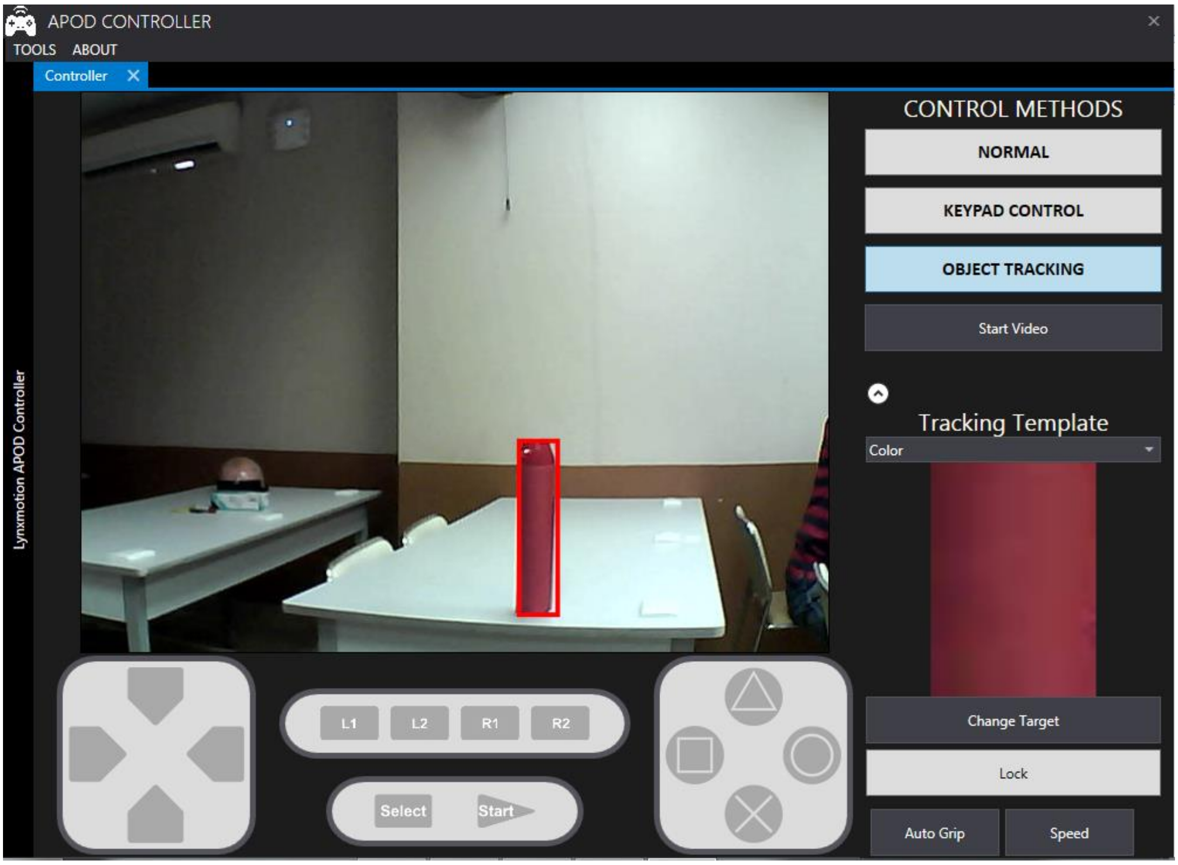


Figure F-25 Camera View with tracking

2.6.2 Change Target

If you wish to change tracking target from main view, just click on the “Change Target” button, the Object Extractor dialog will show up again for you to select new tracking template. Refer to the Choose Target section for details how to select tracking template.

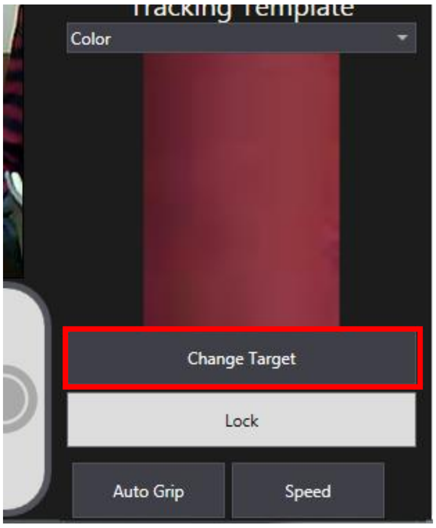


Figure F-26 Change Target button

2.6.3 Tracking

By clicking on the "Lock" button, the APOD will actually move toward the target until it reach the target (less than 40 cm). All other moving function will be locked down during the movement.

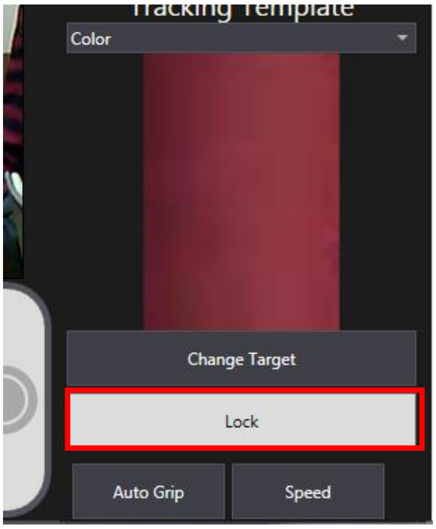


Figure F-27 Lock Button

2.6.4 Auto grip

The "Auto grip" function will make the APOD move forward and grab the object detected in front of the APOD (maximum 40 cm from APOD). The "Auto Grip" move will be trigger after clicking on "Auto Grip" button.

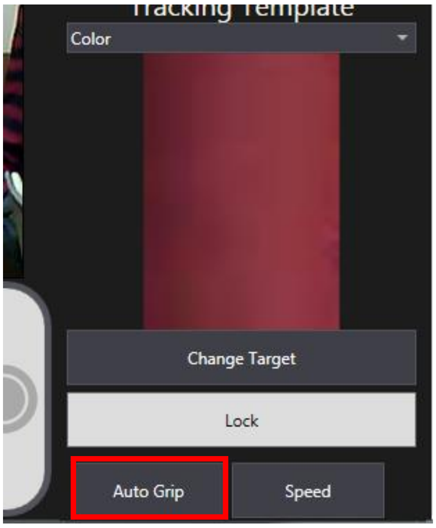


Figure F-28 Auto Grip button

3. Sequence Player

3.1. How to access?

On the top left corner, click on menu "TOOL" and select "Sequence Player"

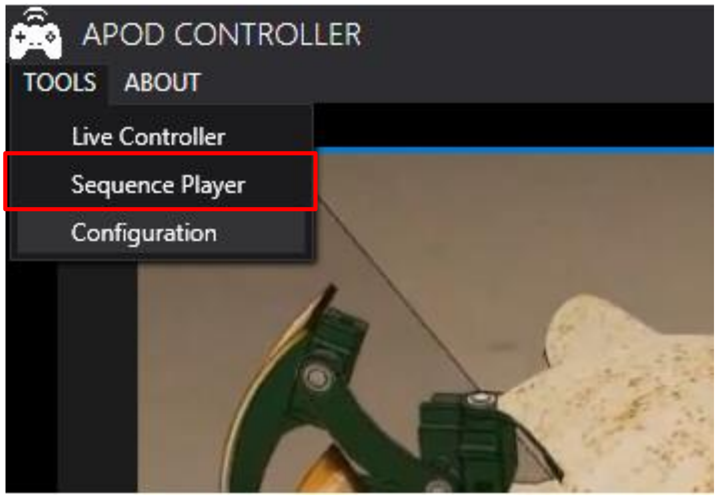


Figure F-29 Access Sequence player

This will be the main window appearance after selecting "Sequence Player"

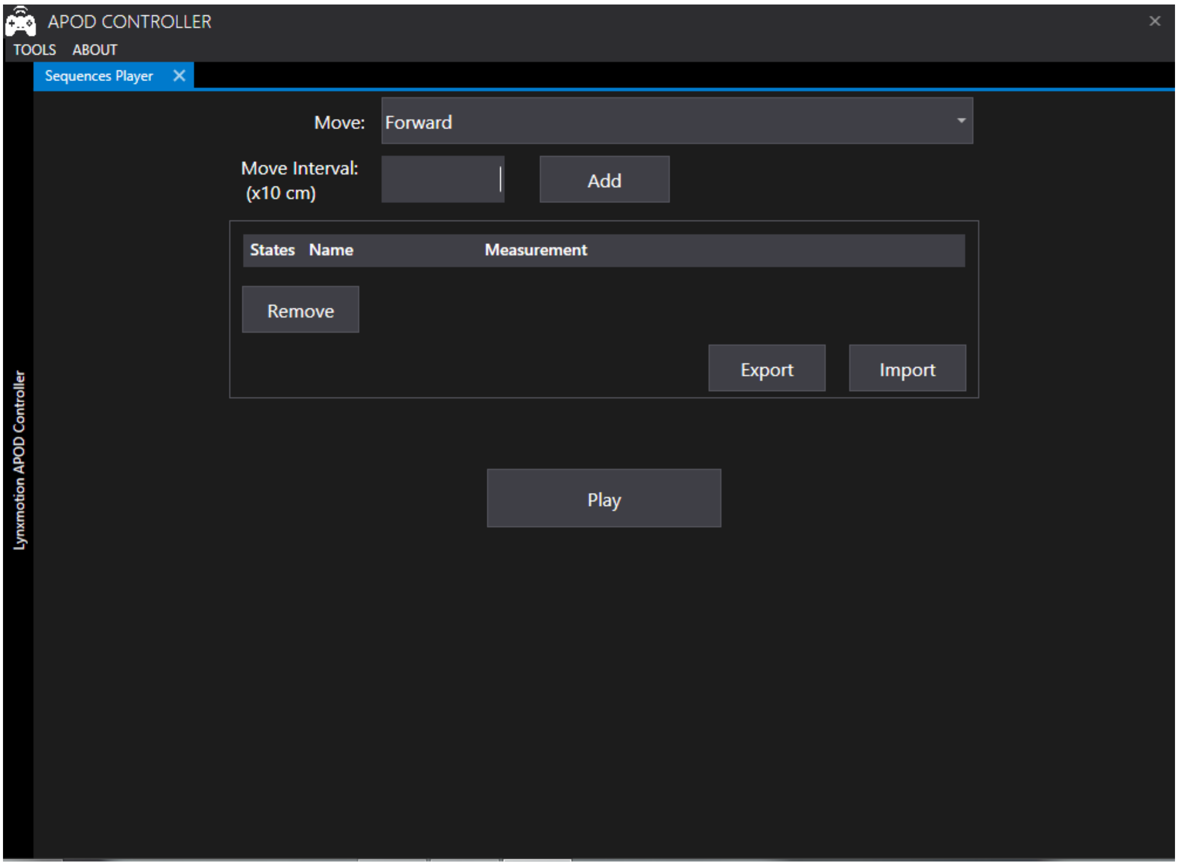


Figure F-30 Sequence Player main appearance

3.2. Add State

To add a state to a sequence, follow these 2 steps:

- Select move from the "Move" drop down list.

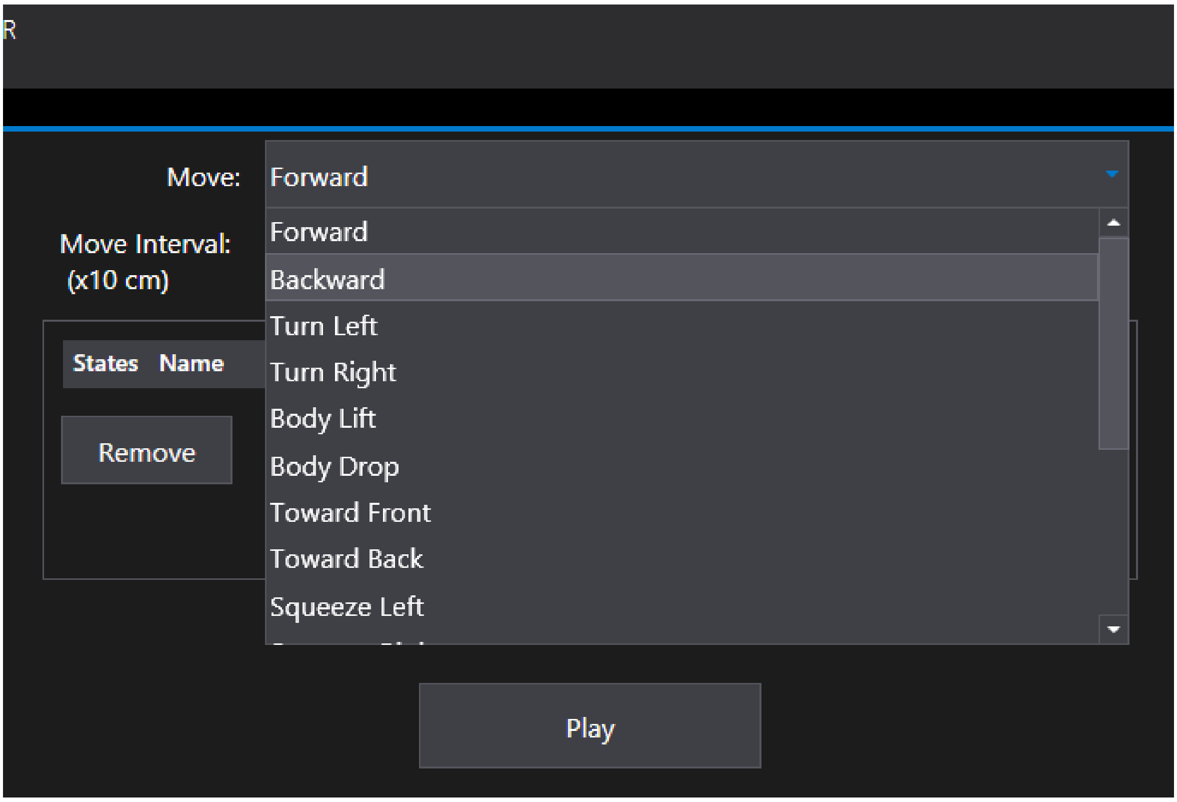
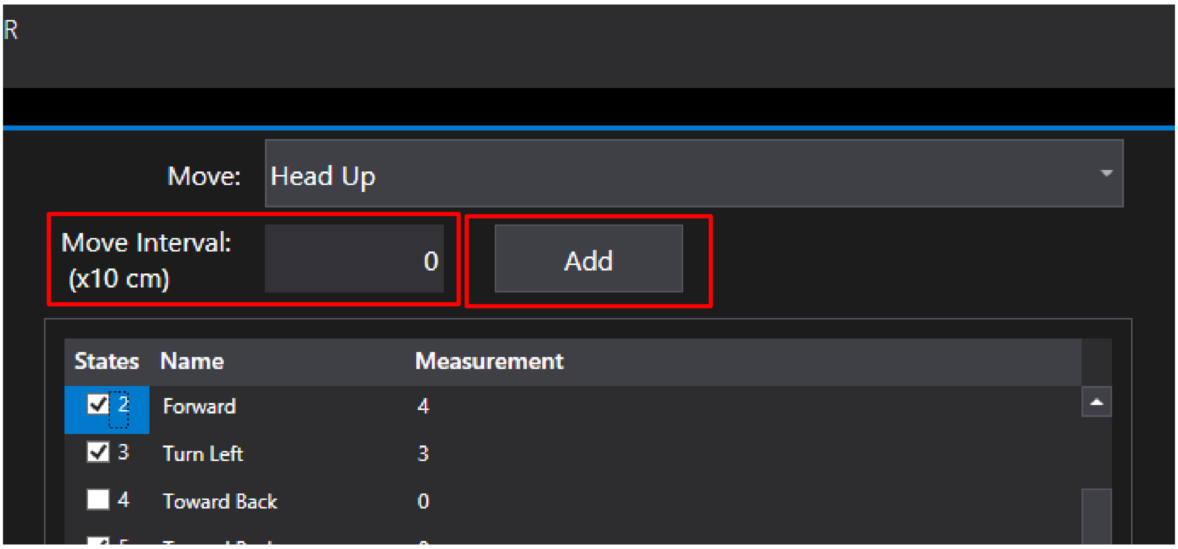


Figure F-31 Select Move

- Enter moving interval in the “Move Interval” fields.
- Click “Add” button



Note that only 4 moves that will require the move interval: Forward, Backward, Turn Left, Turn Right. When selecting any move out of those 4s, the “Move Interval” field will be auto set to “0”.

3.3. Remove States

Form the existing sequence, to remove states, follow these steps:

- Select States to be removed by the check box on the left.

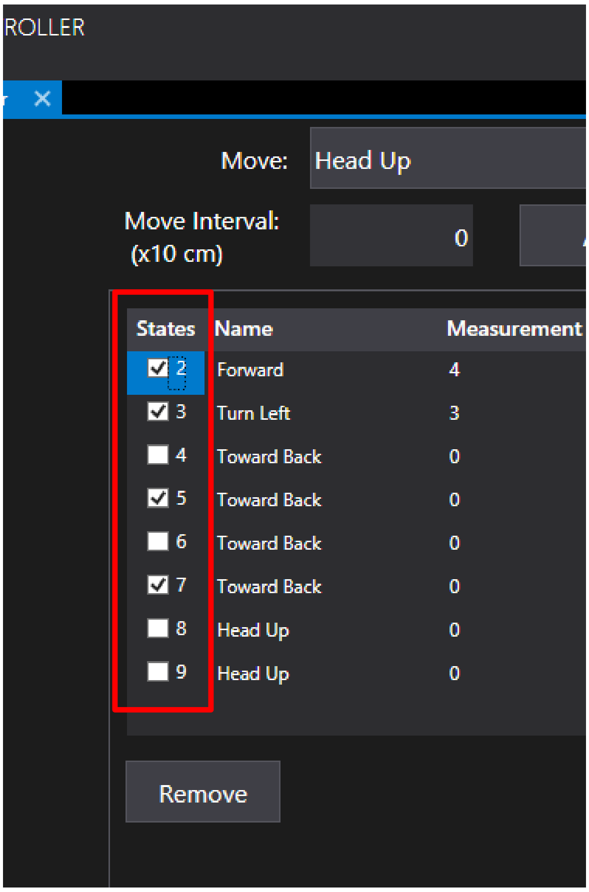


Figure F-32 Select States to remove

- Click on "Remove" button. Those states that was selected will be removed from sequence.

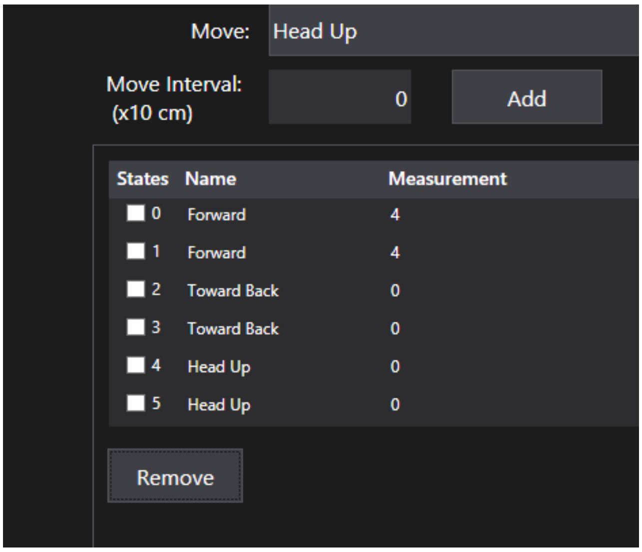


Figure F-33 After Removal

3.4. Import States

A sequence can be saved as a .xml file for later use (see Export States). To read a sequence from file, follow these steps:

- Click on "Import" button

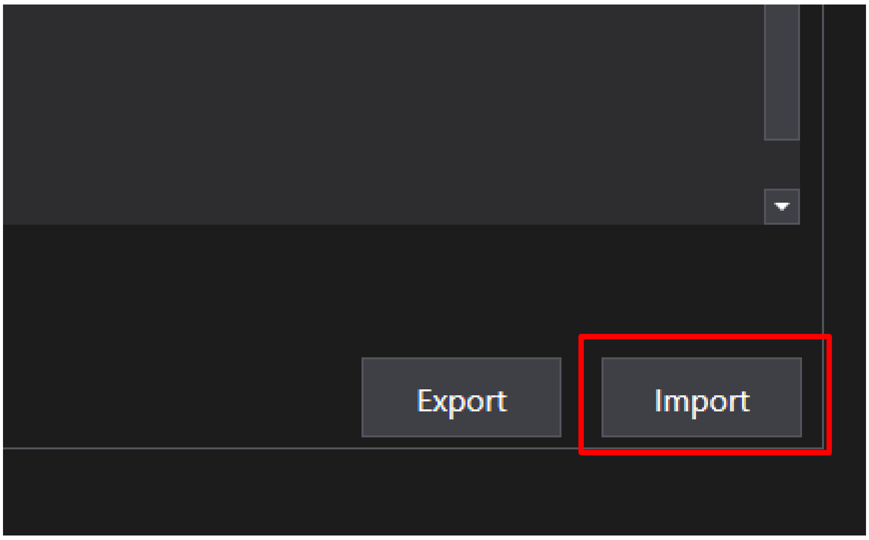


Figure F-34 Import Button

- Browse to the location of sequence file.

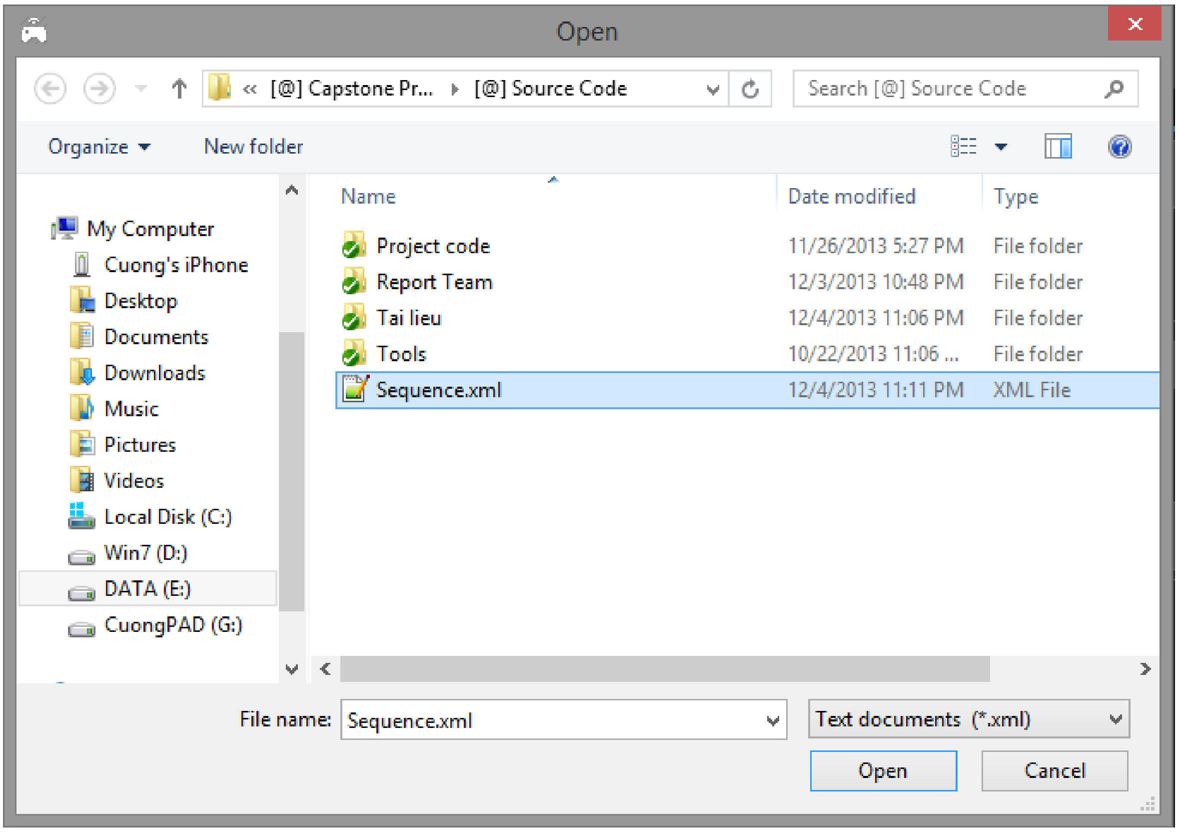


Figure F-35 Browse file to import

- Select file and click Open.

The current sequence will be replaced by file's content.

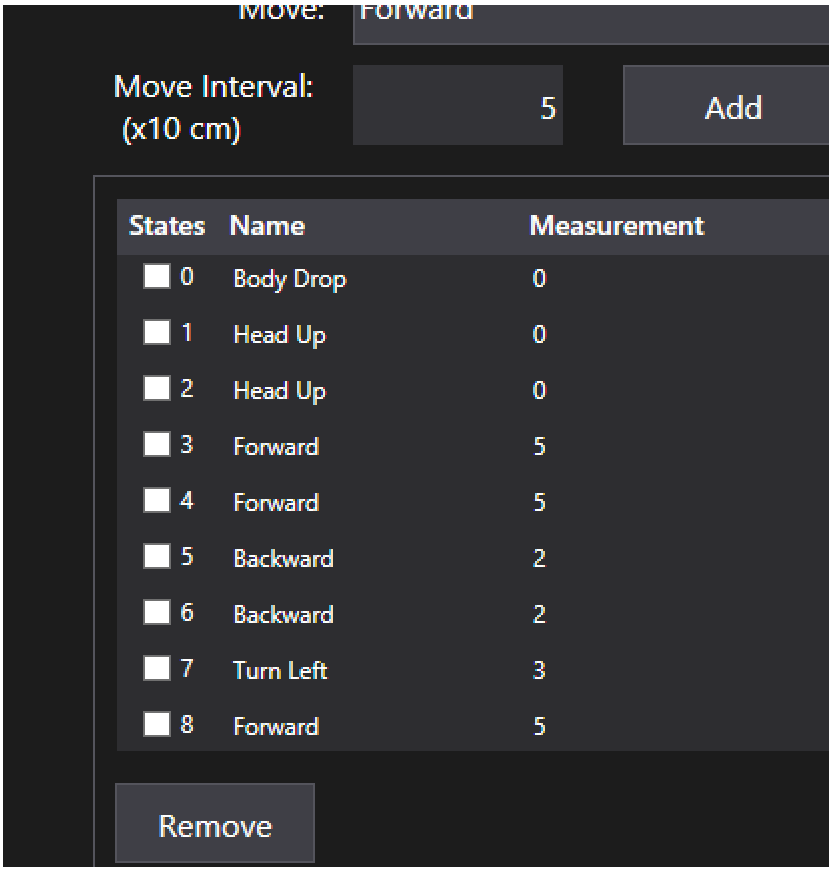


Figure F-36 After import

3.5. Export States

To export current sequence to file, follow these steps:

- Click on "Export" button

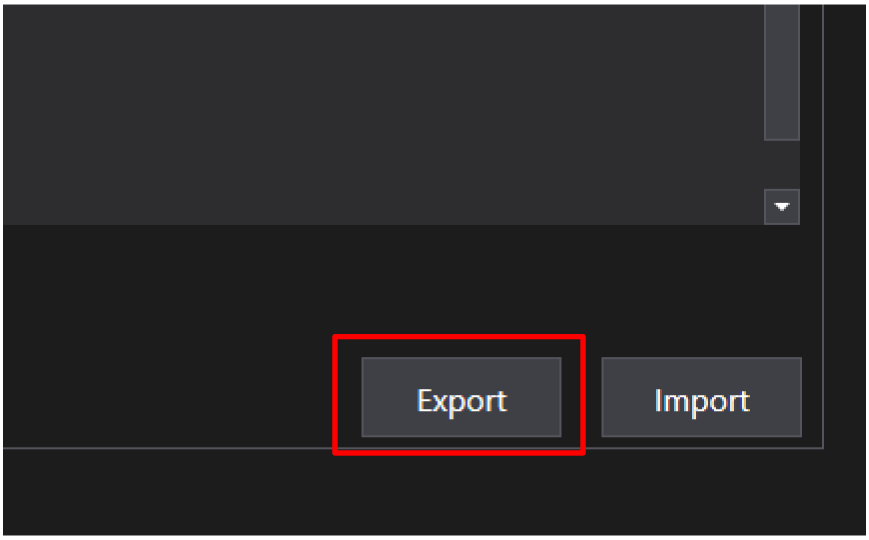


Figure F-37 Export button

- Browse to the location to save file

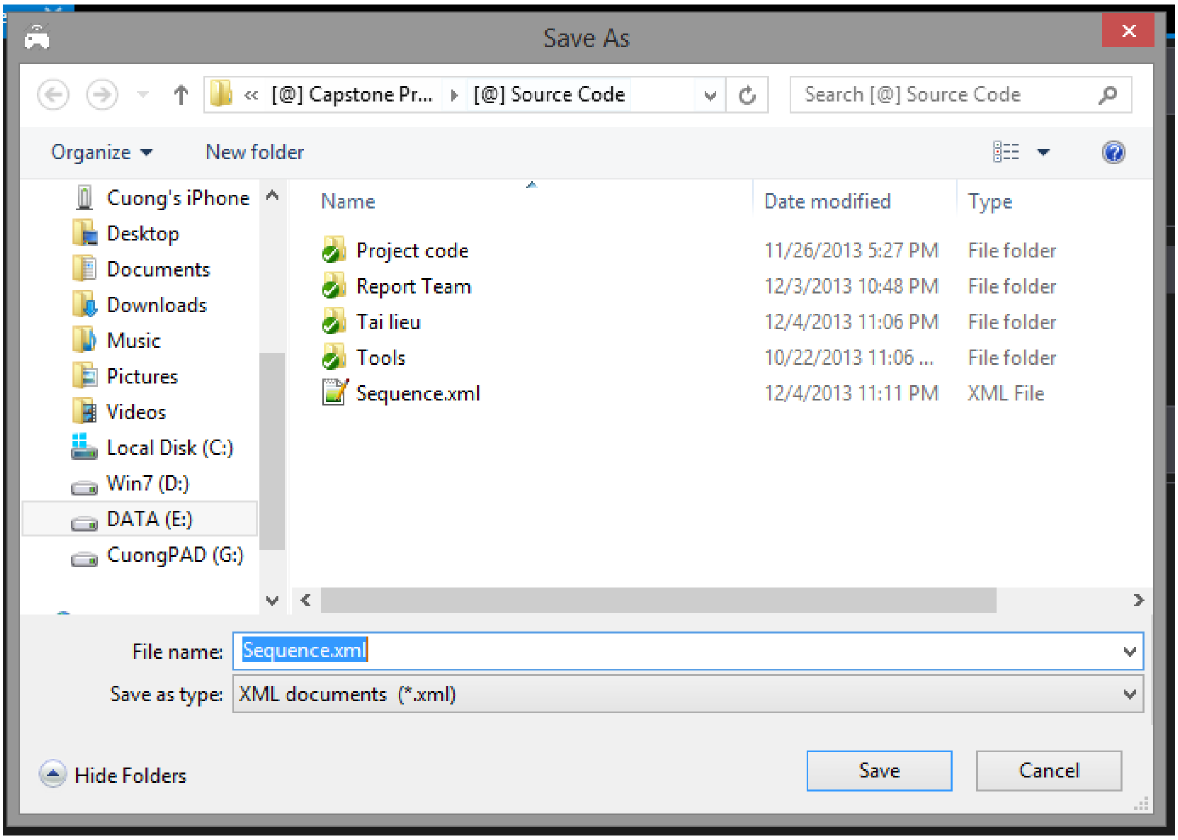


Figure F-38 Browse to save file

- Enter file name and click Save.

Current sequence states will be save to a text file (.xml) as below figure.

Sequence.xml

```xml
<Moves>
  <Move Name="Body Drop" Interval="0" />
  <Move Name="Head Up" Interval="0" />
  <Move Name="Head Up" Interval="0" />
  <Move Name="Forward" Interval="5" />
  <Move Name="Forward" Interval="5" />
  <Move Name="Backward" Interval="2" />
  <Move Name="Backward" Interval="2" />
  <Move Name="Turn Left" Interval="3" />
  <Move Name="Forward" Interval="5" />
</Moves>
```

Figure F-39 Sequence file

3.6. Execution

To execute the sequence, just click on the “Play” button, the APOD will move by the pre-set sequence.

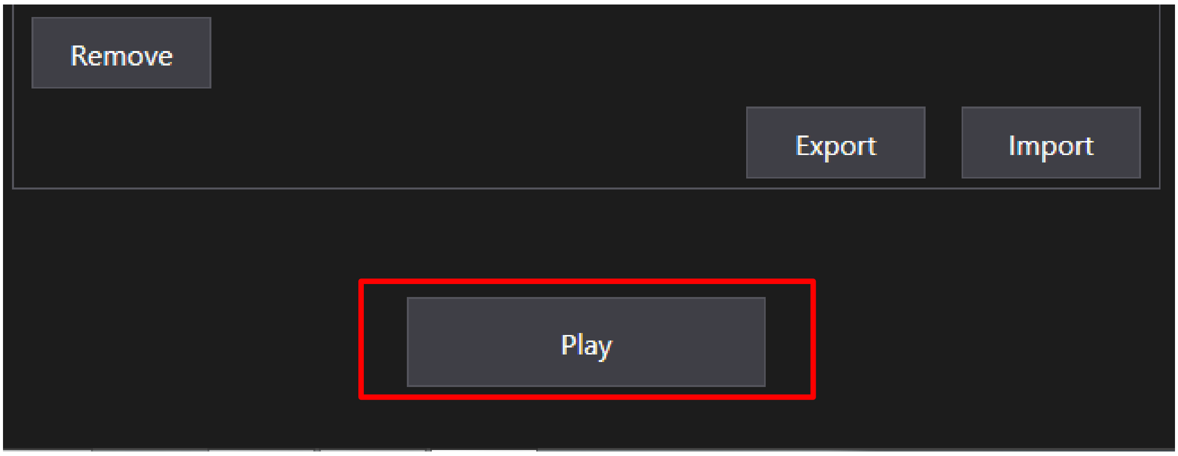


Figure F-40 Play button

Note that while the sequence is being played, all others movement control will be locked down until sequences is finish.

G. APPENDIX

I. Glossary

Acronym	DEFINITIONS	Notes
ADC	Analog - digital converter	
APOD	Name of the hexapod	
AVR	Microcontroller which was developed by Atmel in 1996	
CMS	Content management system	
COM	Communication port/ serial port	
DLL	Dynamic-link library	
DMIPS	Dhrystone Microprocessor without Interlocked Pipeline Stages	
DOF	Degree of freedom	
EEPROM	Electrically erasable programmable read-only memory	
ES	Embedded system	
FPU	Float-Point Unit	
FS	Half speed	
GND	Ground	
HEXAPOD	A structure with 6 legs	
HMI	Human machine interface	
HS	Full-speed	
IS	Information system	
ISP	In-system programming	
IT	Information technology	
MCU	Microcontroller unit	
MISO	Master out slaver in	
MOSI	Master in slaver out	
NiCad	Nickel–cadmium battery	
Ni-MH	Nickel–metal hydride battery	
OTG	On-the-go	
PIC	Programmable intelligent computer	
PVC	Polyvinyl chloride	
PWM	Pulse width modulation	
RISC	Reduced instruction set computing	
RS232	Physical serial interface standard	
SCK	Source clock	
SDD	Software design description	
SDM	Systems development method	
SPI	Serial peripheral interface	
SPMP	Software project management plan	
SRAM	Static random-access memory	
SRS	Software requirement specification	
SSC-32	Serial servo controller board (32 servos)	
STD	Software test document	

SUM	Software user manual	
USART	Universal asynchronous receiver/transmitter	
USB	Universal serial bus	
VCC	IC power-supply pin	

Table G-1 Glossary

II. List of Figures

III. List of Tables

IV. References

All you need to know about AVR fuses . (2013, May 30). Retrieved from Embedds: http://www.embedds.com/all-you-need-to-know-about-avr-fuses/

Atmel. (n.d.). *ATmega328p*. Retrieved from Atmel Website: http://www.atmel.com/devices/atmega328p.aspx?tab=overview

Atmel. (n.d.). *ATmega328p.* Retrieved from Atmel Web site: http://www.atmel.com/devices/atmega328.aspx?tab=documents

Electrical and Computer Engineering. (n.d.). *C Coding Standard* . Retrieved from Electrical and Computer Engineering Web site: http://users.ece.cmu.edu/~eno/coding/CCodingStandard.html

Frye, J. (2010, June 16). *SSC-32 Manual*. Retrieved from Lynxmotion Web site: http://www.lynxmotion.com/images/html/build136.htm

ITead Studio. (n.d.). *Ultrasonic Ranging Module HC-SR04.* Retrieved from ITead Studio Web site: http://imall.iteadstudio.com/im120628012.html

Jenkins, P. (n.d.). Retrieved from http://mahapps.com: http://mahapps.com/MahApps.Metro/

Kirillov, A. (n.d.). Retrieved from aforgenet.com: http://www.aforgenet.com/framework/

Lötzsch, W. (2012, December 9). *Visual Studio 2012 Metro Styles for WPF*. Retrieved from Codeproject Web site: http://www.codeproject.com/Articles/442856/Visual-Studio-2012-Metro-Styles-for-WPF

NXP . (n.d.). *74HC595PW.* Retrieved from NXP Web site: http://www.nxp.com/products/logic/shift_registers/74HC595PW.html

Oricom Technologies . (2003, January). *Analysis of Multi-Legged Animal + Robot Gaits*. Retrieved from Oricom Technologies Web site: http://www.oricomtech.com/projects/leg-time.htm#Hex2

STMicroelectronics . (n.d.). *STM32F4DISCOVERY*. Retrieved from STMicroelectronics Web site: http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419#

STMicroelectronics. (n.d.). *STM32F407VG.* Retrieved from STMicroelectronics Web site: http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN11/PF252140#

STMicroelectronics STM32F407VG. (n.d.). Retrieved from Keil Web site: http://www.keil.com/dd/chip/6103.htm

Tushar. (2013, January 25). *Controlling servo motor with AVR* . Retrieved from Embedds: http://www.embedds.com/controlling-servo-motor-with-avr/

Tushar. (2013, July 26). *Interfacing shift register with AVR* . Retrieved from Embedds Seb site: http://www.embedds.com/interfacing-shift-register-with-avr/

Tushar. (2013, August 4). *Interfacing Ultrasonic Rangefinder with AVR* . Retrieved from Embedds: http://www.embedds.com/interfacing-ultrasonic-rangefinder-with-avr/