**SRD Hexapod**

**Embedded Connectivity**

**Version 1.3**



**Author:** Rob van Eldijk, Ruud Ermers

**Organization:** Fontys Hogescholen Engineering

**Date:** April 22nd, 2013

**Place:** Eindhoven

**Index**

[1 Introduction 3](#_Toc309639846)

[1.1 Purpose of this document 3](#_Toc309639847)

[1.2 Target group 3](#_Toc309639848)

[1.3 References 3](#_Toc309639849)

[1.4 Definitions and short reference string 4](#_Toc309639850)

[1.4.1 Definitions 4](#_Toc309639851)

[1.4.2 Abbreviations 4](#_Toc309639852)

[2 Common Description 5](#_Toc309639853)

[2.1 Product perspectives of ‘Hexapod Control System’ 5](#_Toc309639854)

[2.1.1 System interfaces 5](#_Toc309639855)

[2.1.1.1 Hexapod (Device) 6](#_Toc309639856)

[2.1.1.2 Network Connectivity 6](#_Toc309639857)

[2.1.1.3 Config Tool 6](#_Toc309639858)

[2.1.2 User Interface 6](#_Toc309639859)

[2.1.2.1 Browser 6](#_Toc309639860)

[2.1.2.2 Config Tool 7](#_Toc309639861)

[2.1.3 Hardware interfaces 7](#_Toc309639862)

[2.1.3.1 Ethernet Interface (RJ45) 7](#_Toc309639863)

[2.1.3.2 USB Interface (x2) 7](#_Toc309639864)

[2.1.3.3 Serial Interface 7](#_Toc309639865)

[2.1.3.4 I2C Interface 7](#_Toc309639866)

[2.1.3.5 PWM Interface 8](#_Toc309639867)

[2.1.3.6 Power Interface 8](#_Toc309639868)

[2.1.4 Software interfaces 8](#_Toc309639869)

[2.1.4.1 Software Update Tools 8](#_Toc309639870)

[2.1.4.2 Network Connectivity 8](#_Toc309639871)

[2.1.4.3 Browser 8](#_Toc309639872)

[2.1.4.4 USB Drivers Microchip 8](#_Toc309639873)

[2.1.5 Communication interface 8](#_Toc309639874)

[2.1.5.1 PWM 8](#_Toc309639875)

[2.1.5.2 PIP 9](#_Toc309639876)

[2.2 System overview with functional demands 9](#_Toc309639877)

[2.3 Internal Block Diagram 9](#_Toc309639878)

[3 Requirements 10](#_Toc309639879)

[3.1 Requirements Identification 10](#_Toc309639880)

[3.2 Hardware Requirements 11](#_Toc309639881)

[3.3 External Interface Requirements 12](#_Toc309639882)

[3.3.1 User Interface 12](#_Toc309639883)

[3.3.2 Hardware Interface 12](#_Toc309639884)

[3.3.3 Software Interface 12](#_Toc309639885)

[3.3.4 Communication Interface 12](#_Toc309639886)

[3.4 Functional Requirements 13](#_Toc309639887)

[3.4.1 Mode: Control Hexapod 13](#_Toc309639888)

[3.4.2 Mode: Send Webpage Data 13](#_Toc309639889)

[3.4.3 Mode: Update Software / Set system parameters 13](#_Toc309639890)

[3.5 Non-functional Requirements 14](#_Toc309639891)

[3.5.1 Design restrictions 14](#_Toc309639892)

[3.5.2 Availability 14](#_Toc309639893)

[3.5.3 Security 14](#_Toc309639894)

[3.5.4 Interoperability 14](#_Toc309639895)

[3.5.5 Testability 14](#_Toc309639896)

[4 Design 15](#_Toc309639897)

[5 Links 16](#_Toc309639898)

**Version Management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Important modification** | **Author** | **Reviewed** |
| 14-03-11 | 0.1 | Initial version | RvE |  |
| 21-03-11 | 0.2 | Chapter 1 and 2 refined | RvE |  |
| 06-04-11 | 0.3 | Chapter 3 | RvE | Ems |
| 14-04-11 | 0.4 | Chapter 3 finished + 4 | RvE | Ems |
| 27-10-11 | 1.1 | Completely reworked according to template | Ems | Dui |
| 21-11-11 | 1.2 | Rework after remarks | RvE | Ems/Dui |
| 22-04-13 | 1.3 | Update for the new board, and changes to required documentation | Jdl | RvE |
| 01-04-2018 | 2.0 | Updated for the Basys MX3 and SSC-32U USB Servo Controller Board | Jdl |  |

# Introduction

## Purpose of this document

This document describes the requirements for the Hexapod Control System (HCS). This system has to be built as part of the 2nd year grade module “Embedded Connectivity”. It describes both the user requirements and the system specification.

## Target group

This document is targeted at the tutor of the group of the module “Embedded Connectivity”. Moreover the members of the group are described.

|  |  |  |
| --- | --- | --- |
| **Name** | **Organization** | **Function** |
| TBD | TBD | TBD |
| TBD | TBD | TBD |

## References

References to this document:

* Plan of Action
* TBD

Document references to Lynxmotion:

* Lynxmotion SSC-32U USB Servo Controller Board, V1.1 August 2015

Document references to Basys MX3:

* Basys MX3 Board Reference Manual, Digilent Inc,
* Basys MX3 Schematics, Digilent Inc,
* Microchip Harmony V1.11 TCP/IP Stack™ Reference Design , Digilent Inc,
* PIC32MX3XX/4XX Data Sheet – DS61143G, Microchip Technology Inc.

(See the N@Tschool entry and the Links chapter in this document)

## Definitions and short reference string

### Definitions

|  |  |
| --- | --- |
| **Definition** | **Description** |
| MPLABX IDE | Microchip’s C programmer / debugger |
| MPLABX XC32 | Microchip’s C compiler |

### Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| HCS | Hexapod Control System |
| TBD | To Be Determined |

# Common Description

## Product perspectives of ‘Hexapod Control System’

Fontys Engineering has bought 7 so-called Hexapod Robots (see the cover of this document for an impression), which are used for promotional tasks. The objective of this product (Hexapod Control System, HCS) is to create an easy to maintain software/hardware platform to control the robots remotely. In order to use them at promotional events the robots should be able to walk through a simple maze, controlled by the user. To make things more difficult, the user should be able to control the Hexapod but not see it. Therefore the Hexapod systems shows the user on its screen (which might be a smart phone) part of the environment. The user then should be able to control the Hexapod through the maze.

There are some extra requirements on the product. This way it can be used in a class setting, ‘Embedded Connectivity’ in the second year grade (fourth semester). Extra requirements include using two PIC32 Embedded Systems, programmed in the C-language. Moreover, a lot of interfaces, e.g. Ethernet, Bluetooth, I2C, USB, serial communication must be used. One of the PIC32 boards will be integrated with the Hexapod; the other will be used as a web server.

### System interfaces



Figure 1: External Block Diagram HCS

#### Hexapod (Robot)

The Hexapod is a standalone robotic insect with six legs, which is able to move in four direction with a speed of approximately TBD m/s and which is able to scan the environment by means of a distance sensor (range from TBD to TBD) and a (motor controlled) moving head. Scanning is only possible in the direction of the “head” of the spider. The control of this system is done by the HCS.

#### Network Connectivity

The Network Connectivity is a network setup which is able to supply IP addresses (DHCP server) and has routing capabilities, including wireless connectivity. The user will use the Network Connectivity to make a connection from a capable browser to the HCS. A capable browser is any browser that can display a 320 x 200 HTML/AJAX based website.

Note: The Network Connectivity Environment is not part of the system, but must be built as part of the project assignment.

#### Config Tool

The configuration tool is a collection of PC based tools, which is able to

* Load code into the HCS environment
* Start/Stop the HCS
* Set the IP address of the board or to set DHCP functionality
* Set the internal Bluetooth configuration

Note: The Config Tool is not part of the system, but must be built as part of the project assignment.

### User Interface

#### Browser

Figure : User Interface HCS

The web browser provide the user interface to the HCS as shown in Figure 2.

The user interface is a website environment based on HTML and AJAX. The server-side scripting is performed according to the methods in Harmony V1.11 Microchip’s TCP/IP stack. Some bits of JavaScript may be necessary to complete the website. The website consists of a page with:

* four buttons that control the Hexapod
* three rows of five blocks that give the condition and position of the ultrasonic sensor

##### On Screen Buttons

The four buttons are needed for controlling the Hexapod from the starting line to the finish line. The buttons have the following conditions:

* Up: The Hexapod goes forward.
* Left: The Hexapod goes left.
* Right: The Hexapod goes right.
* Down: The Hexapod goes backward.

When one of the buttons is clicked, the colour of the button changes until the button is released.

When none of the buttons is clicked, the Hexapod stands still.

##### On Screen Blocks

The blocks represent the “view” of the ultrasonic sensor. By automatically panning and tilting the head it is possible to make a kind of radar with ultrasound.

It must see the difference between three heights, five widths and multiple depths.

#### Config Tool

TBD. Note: The Config Tool should be able to run on any system having the required USB drivers and tools. It should not be necessary to have the full compiler installed.

### Hardware interfaces

#### Ethernet Interface (RJ45)

The Ethernet Interface is connected to the Network Connectivity system.

#### USB Interface (x2)

The USB Interface is connected to the Config Tool PC, for configuration.

#### Serial Interface

The Serial Interface is connected to the Hexapod. It uses a micro USB/UART4.

TBD: Specify Baud rate etc.

#### I2C Interface

The I2C Interface is connected to the Hexapod. It uses a Basys MX3 board-specific pin-out and connector.

TBD: Specify I2C hardware settings and pin-outs.

#### PWM Interface

The PWM Interface is connecting the Servos to the SSC32U controller to control the movement of the Hexapod robot.

#### Power Interface

TBD [[1]](#footnote-1)

### Software interfaces

#### Software Update Tools

TBD

#### Network Connectivity

The Network Connectivity is able to supply IP addresses. Therefore no other network configuration for the HCS system is needed. However for testing facilities, fixed IP-addressing should be made possible.

#### Browser

TBD (Specify a few standard browsers and a few mobile browser. Furthermore specify AJAX/Javascript capabilities)

#### USB Drivers Microchip

TBD

### Communication interface

* TCP/IP: Controls network between server and client.
* HTTP: Transfers website information.
* HTML/AJAX: Language where the web browser gets its information.
* I²C: Interface bus between microcontroller and Ultrasonic sensor.
* PWM: PWM signals used to steering servos in Hexapod robot
* UART: Interface between Basys MX3 and the SSC32U Servos Controller.
* Sequencer: SW protocol block executing in Basys for controlling the Hexapod.

#### PWM

PWM (Pulse Width Modulation) is used for steering the servo. Specifications for the PWM are:

* Neutral position 1500μsec
* Operation angle 45°/ one side pulse travelling 400μsec
* Total pulse travel form 1100μsec to 1900μsec
* Dead band width 8μsec for HS-225 and 5μsec for HS-322

#### Sequencer

Sequencer is used for controlling the Hexapod.

It can control:

* Motion in all directions, speed and walk-gaits.
* Body movement around its axes.
* Single leg movement.
* Single servo movement.

## System overview with functional demands

The HCS controls the Hexapod by taking commands from the user (e.g. walk, left, stop), translate them, and send control information to the Hexapod. It gets the environment from the Hexapod and shows this to the user. Controlling / Showing information is performed from a browser who gets its information from the HCS. The HCS can update its software using the config tool. Moreover a few parameters can be adjusted (See 2.1.1.4).

C:\Jeedella\Fontys\Courses\SASD_SysML\Lab\Hexapod\Hexapod_Context_diag.tiff

Figure 3: System Context Diagram

The system will be tested in a race environment. For this, a racetrack shall be set up containing some obstacles, a start line and a finish board. The obstacles are at least 30 centimetres high, the finish board is at least one meter high. The area of the racetrack measures 5 x 2 meters.

As stated before, the user that controls the Hexapod should be in another room, not being able to see the race track or the Hexapod. Two Hexapods will race against each other.

## Internal Block Diagram

TBD. Here you should give a detailed overview of the HCS, in terms of hardware components. Note that we require two subsystems, one integrated with the Hexapod, one as a web server. This should become clear from the internal block diagram. The internal block diagram can be used in the design to create a mapping between hardware and software components and adding behaviour.

# Requirements

## Requirements Identification

**<Requirement Type>-<Nr> [. <Sub>]**

Where

**<Requirement Type>** specifies the type of requirement:

**EXT**  -Extern Interface

**SF** - System Feature

**PRF -** Performance

**DES -** Design Constraints

**SSA -** Software System Attribute

**HW** - Hardware

**OTH -** Other

And

**<Nr>** - Is a unique number inside the type of requirement.

**<Nr>.<Sub> -** If there is a sub-requirement than use: <nr>**.**<sub>.

Example: **EXT-12** - Requirement number 12.

**EXT-12.3** - Sub-requirement 3 of requirement number 12.

Requirement priorities are:

* Essential
* Conditional
* Optional

## Hardware Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **HW-01** | **Controller** | | **Conditional** |
| Main microcontroller | A PIC32MX370F512L on a Basys MX3 must be used as a microprocessor environment to control the Hexapods and its peripherals | | |
| I²C Ultrasonic sensor | **HW-01.1** | **Ultrasonic sensor** | **Conditional** |
|  | The Hexapod Ultrasonic Sensor must be interfaced with the microcontroller over I2C | |
| Bluetooth | **HW-01.2** | **Bluetooth B** | **Conditional** |
|  | The Web-Server microcontroller must be interfaced with the microcontroller through (Serial over) Bluetooth | |
| Pan & Tilt Head | **HW-01.3** | **Pan & Tilt Head** | **Conditional** |
|  | The Hexapod pan & tilt head must be interfaced with the SSC32U servo controller through PWM | |
| Standalone | **HW-01.4** | **Hexapod Standalone** | **Conditional** |
|  | The main microcontroller must be assembled on the Hexapod.  It must be able to work standalone for at least 45 minutes, all systems active. | |
| **HW-02** | **Web-Server** | | **Conditional** |
| Web-server | A PIC32MX370F512L on a Basys MX3 must be used as a microprocessor environment for setting up a web server environment. | | |
| TCP/IP Stack | **HW-02.1** | **TCP / IP Stack** | **Conditional** |
|  | The web server connects through TCP/IP to an existing Network Connectivity environment | |
| Bluetooth | **HW-02.2** | **Bluetooth A** | **Conditional** |
|  |  | The web server microcontroller must be interfaced with the main microcontroller through (Serial over) Bluetooth | |

## External Interface Requirements

### User Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **EXT-01** | **User interface** | | **Essential** |
| Webpage | The user controls the Hexapod by a webpage. On the webpage 4 buttons and 3x5 blocks are shown. | | |
|  | Note | The buttons and blocks are specified in chapter 2.1.2.1 | |

### Hardware Interface

|  |  |  |
| --- | --- | --- |
| **EXT -02** | **I²C** | **Essential** |
| I²C Ultrasonic | TBD. Specify Description, Hardware Aspects and Protocol | |
| **EXT -03** | **PWM** | **Essential** |
| Pan & Tilt Head | TBD. Specify Description, Hardware Aspects and Protocol | |
| **EXT -04** | **RJ45** | **Essential** |
| TCP / IP Stack | Communication between the Web-Server and the TCP / IP stack.  TBD. Specify Description, Hardware Aspects and Protocol | |
| **EXT -05** | **UART** | **Essential** |
| Hexapod | TBD. Specify Description, Hardware Aspects | |

### Software Interface

|  |  |  |
| --- | --- | --- |
| **EXT-06** | **Software Update Tools** | **Essential** |
|  | TBD | |
| **SF-02.1** | **Network Connectivity** | **Essential** |
| AJAX/CGI | The web server must have a webpage onboard that can communicate with the embedded side of the web server. The webpage must be easily accessible e.g. through a domain name or a sticker on the system. | |

### Communication Interface

|  |  |  |
| --- | --- | --- |
| **SF-01** | **Main Microcontroller** | **Essential** |
| SDP | The Controller must communicate with a special designed protocol over Bluetooth.  TBD. Specify addressing settings | |
| **SF-02** | **Main Microcontroller** | **Essential** |
| Sequencer | The Controller must communicate with the Hexapod by Sequencer | |
| **SF**-**03** | **Web-Server** | **Essential** |
| SDP | The web server must communicate via a special designed protocol over Bluetooth.  TBD. Specify addressing settings | |

## Functional Requirements

### Mode: Control Hexapod

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SF-04** | **Control Hexapod** | | | **Essential** |
|  | The Up, Down, Left, and Right buttons control the movement of the Hexapod. The Hexapod will start walking with a speed (as described below) when a button is pressed and will stop within TBD msec when the button is released. Moreover, if no ‘released’ message from the button is received but another button is pressed the Hexapod will move in the new direction. | | | |
|  | **SF-04.1** | **Speed Control** | **Optional** | |
|  | TBD (Describe a method in which the Hexapods starts slowly, but gains speed after some time) | | |
|  | **SF-04.2** | **Bouncing** | **Optional** | |
|  | If the Hexapod would bounce against an object, speed will be decreased to zero. | | |
| **SF-05** | **Get Environment From Hexapod** | | | **Essential** |
|  | Reads the environment from the head of the Hexapod by moving the head and performing distance measuring with the ultrasonic sensor. The result will be send over Bluetooth with the SDP Protocol. The action will be initiated by the web server microcontroller. See the SDP Protocol for details. After the initial request the data must be available within 0.2 seconds. | | | |
|  | **SF-05.1** | **Terminal Interface** | **Optional** | |
|  | The data must be readable in a terminal program. | | |

### Mode: Send Webpage Data

|  |  |  |  |
| --- | --- | --- | --- |
| **SF-06** | **Send data for making a webpage.** | | **Essential** |
|  | The web server sends a webpage with the user interface to the client. The webpage must be available within TBD second (on a stable, wired network), AJAX updates must be available within TBD seconds. | | |
|  | **SF-06.1** | **Android Interface (XML)** | **Optional** |
|  | The data sent by the web-server can be viewed by any web browser or mobile device with a web browser. The possibility of an external program on an Android mobile device, which can communicate with the web server, must be implemented. | |

### Mode: Update Software / Set system parameters

|  |  |  |
| --- | --- | --- |
| **SF-07** | **Update Software on subsystems** | **Optional** |
|  | The user must be able to select a new firmware from the PC and send it to the corresponding subsystem. | |
| **SF-08** | **Set System parameters on subsystems** | **Optional** |
|  | TBD. Describe the mechanism how parameters can be updated on the subsystems, e.g. Bluetooth parameters. Describe a way how the parameters will be stored (non-volatile). | |

## Non-functional Requirements

See e.g.

[http://www.softwarearchitectures.com/go/Discipline/DesigningArchitecture/QualityAttributes/tabid/64/Default.aspx]( http://www.softwarearchitectures.com/go/Discipline/DesigningArchitecture/QualityAttributes/tabid/64/Default.aspx )

### Design restrictions

|  |  |  |
| --- | --- | --- |
| **DES-1** | **Memory** | **Essential** |
|  | The amount of memory is 512kB. The compiler can only be in the 0 optimization level.  All web server data must be stored in flash. | |
| **DES-2** | **Processor / Programming** | **Essential** |
|  | The processor is the PIC32MX370F512L and must be programmed in standard C.  The following Microchip frameworks are used:  TBD: Specify with versions | |
| **DES-3** | **Board** | **Essential** |
|  | The developing board is a Basys MX3 (x2).  It must be used in Standalone mode. | |
| **DES-4** | **TCP/IP stack /Programming** | **Essential** |
|  | The Ethernet controller is the EN28J60 (available on the PmodNIC). As an abstraction layer for the TCP/IP stack the following Microchip library is used: TBD: Specify with version | |
| **DES-5** | **Walk speed** | **Essential** |
|  | The speed of the Hexapod must be at least 10cm/sec. | |

### Availability

There are no requirements.

### Security

There are no requirements.

### Interoperability

|  |  |  |
| --- | --- | --- |
| **SSA-1** | **Bluetooth** | **Optional** |
|  | The Bluetooth connection should be transparent, that is, any of the two subsystems should be replaceable with a stub application. (Building the stub application is not required). | |

### Testability

|  |  |  |
| --- | --- | --- |
| **SSA-2** | **Testability** | **Essential** |
|  | Testability of the system will be described in the test plan. See Chapter TBD from the test plan:  TBD. Describe how you are able to test parts of the system which are not on a functional level, e.g. logging on the system and how to retrieve data. | |

# [Design](http://www.softwarearchitectures.com/go/Discipline/DesigningArchitecture/QualityAttributes/tabid/64/Default.aspx)

TBD

See e.g. <http://www.altova.com/umodel/sysml.html>

The SDD should consist of:

* System Context Block Definition Diagram
* Use Case Diagrams
* Block Definition Diagrams for both subsystems
* Internal Block Definition Diagrams for both subsystems
* State machine diagram for each block in the system
* Protocols which were not yet defined as part of the SRD (e.g. Bluetooth communication)

# References

* <http://www.lynxmotion.com/p-1032-ssc-32u-usb-servo-controller.aspx>
* <https://nodna.de/Phoenix-3DOF-Hexapod_1>
* http://www.lynxmotion.com/p-1032-ssc-32u-usb-servo-controller.aspx
* <https://store.digilentinc.com/basys-mx3-pic32mx-trainer-board-for-embedded-systems-courses/>
* http://www.microchip.com/wwwproducts/en/pic32mx370f512l<http://www.altova.com/umodel/sysml.html>
* <http://www.softwarearchitectures.com/go/Discipline/DesigningArchitecture/QualityAttributes/tabid/64/Default.aspx>

1. Extra care should be taken for the power interfaces. It is known that, depending on the powering of the board, using the same power supply for the sonar and the servos, this might give issues, e.g. the sonar resets itself when the servo was activated (on USB power). [↑](#footnote-ref-1)