

Robot basic architecture based on SSES Blueprint

Test document

V1.2 03/23/2022

Table of contents

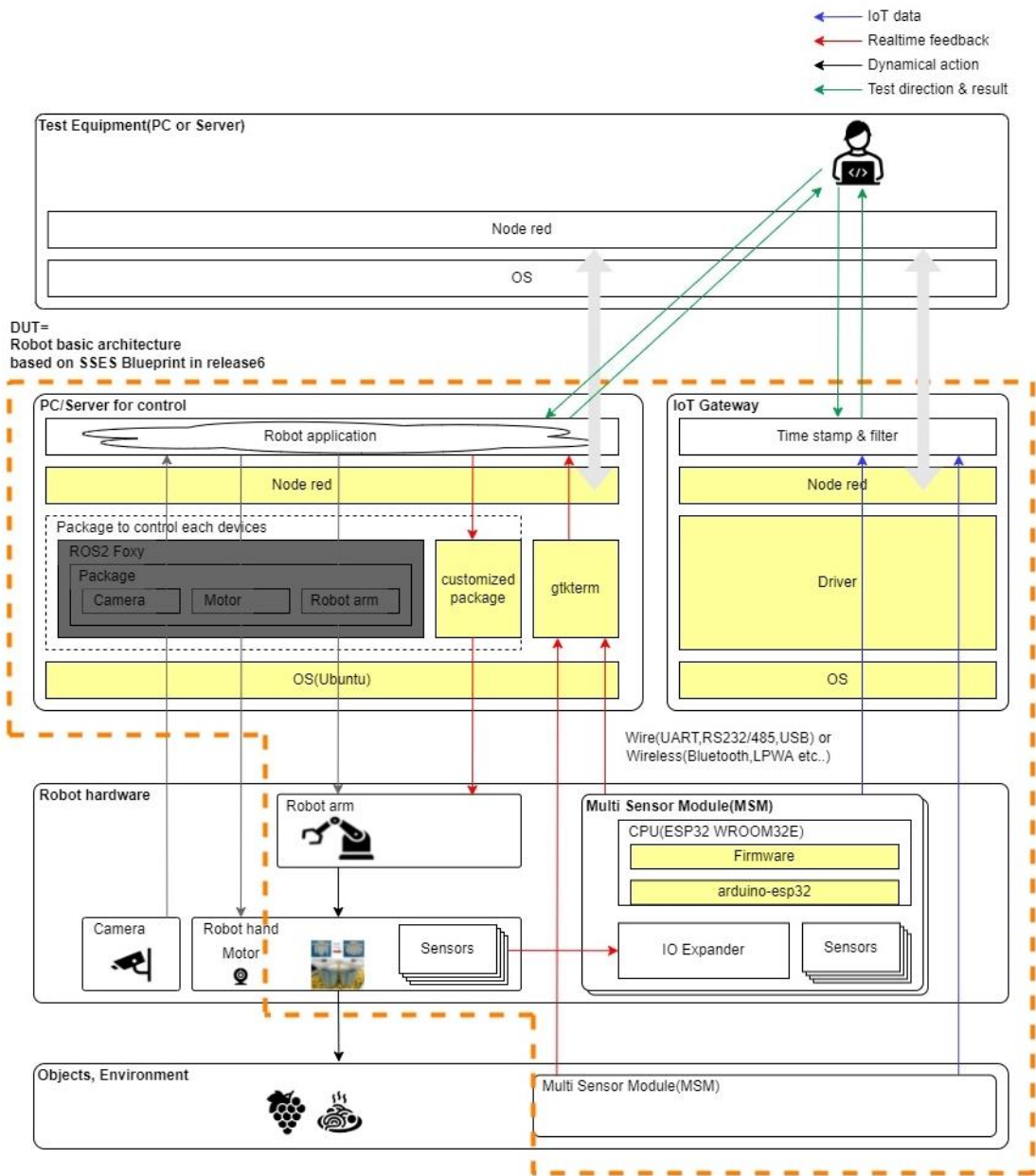
1	Introduction.....	3
2	Overall Test Architecture.....	3
3	Test API description.....	6
4	Revision history	14

1 Introduction

This document covers Test Deployment Environment and Test Case for Robot basic architecture based on SSES Blueprint. The scopes of test are installation SW to HW for robot application and connectivity between each SW and HW.

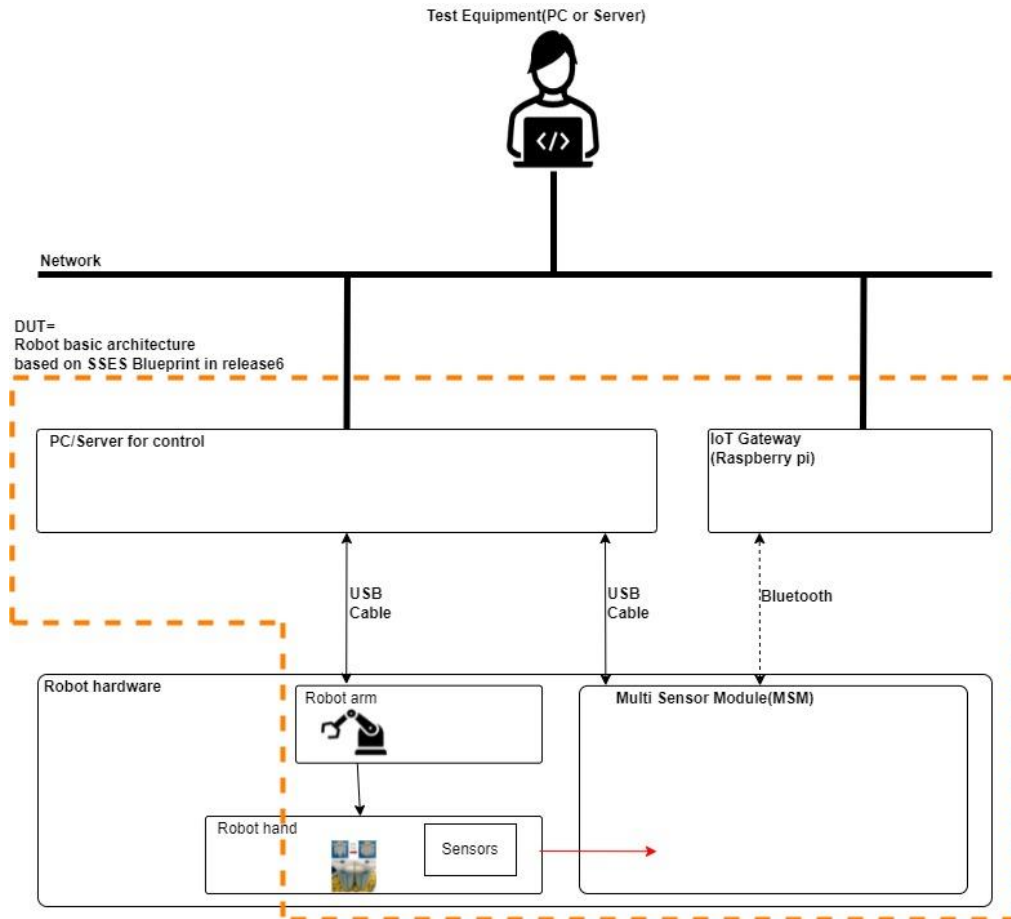
2 Overall Test Architecture

The following figure indicates overall test architecture, DUT(Device under test), and TE(Test Equipment). We will build these test bed in Ritsumeikan university.



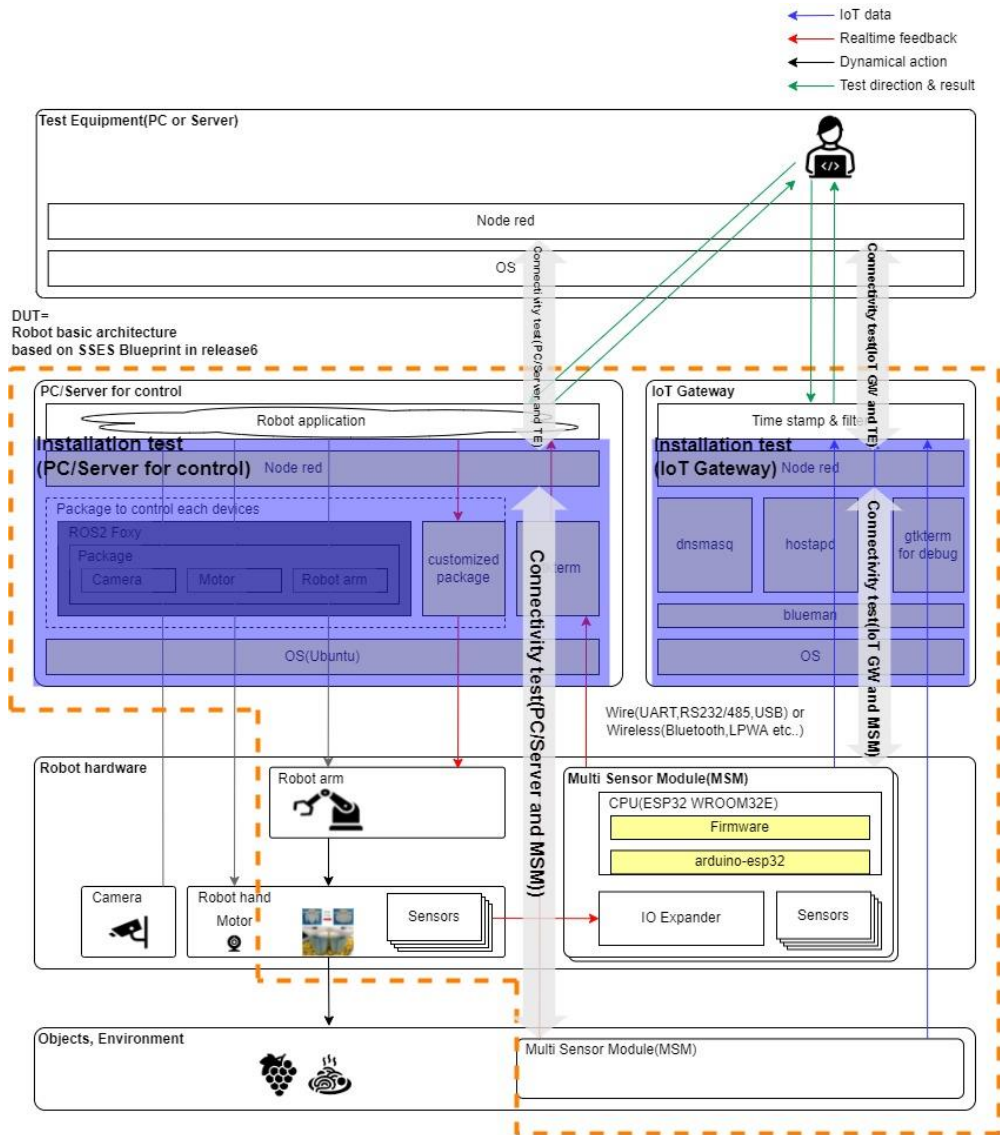
The following figure indicates HW and its connection.

All machines are on the same local area network.



3 Test API description

The following figure coverage of this test.


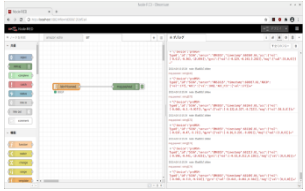
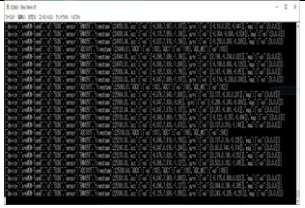


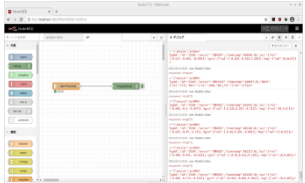
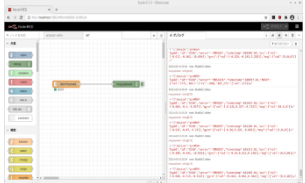
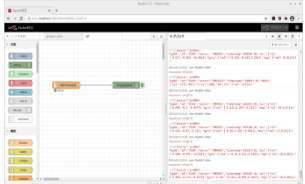
Bare Metal Deployment

No	Test Case	Test input	Test Procedure	Expected output	Test result
1	IoT gateway	-	-	-	-
1-1	system configuration	-	<p>For more details, refer to installation guide.</p> <ol style="list-style-type: none"> Log in to IoT gateway raspberry pi Display setting vi /boot/config.txt hdmi_force_hotplug=1 Set to run Node-RED when PowerON sudo systemctl enable nodered.service Allow VNC and SSH Connect Test Equipment to IoT gateway via USB. Run terminal. Set time server 		
1-2	Install GTKTerm	-	<pre>sudo apt install gtkterm \$ which gtkterm</pre>	/usr/bin/gtkterm	

1-3	Install hostapd	-	sudo apt install hostapd hostapd -v	hostapd v2.X User space daemon for IEEE 802.11 AP management, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator Copyright (c) 2002-2019, Jouni Malinen <j@w1.fi> and contributors	
1-4	Install dnsmasq	-	sudo apt install dnsmasq dnsmasq -v	Dnsmasq version 2.85 Copyright (c) 2000-2021 Simon Kelley Compile time options: IPv6 GNU- getopt DBus no-UBus i18n IDN2 DHCP DHCPv6 no-Lua TFTP conntrack ipset auth cryptohash DNSSEC loop-detect inotify dumpfile This software comes with ABSOLUTELY NO WARRANTY. Dnsmasq is free software, and you are welcome to redistribute it under the terms of the GNU General Public License, version 2 or 3.	
1-5	Install blueman	-	sudo apt install blueman \$ sudo find / -name blueman	/usr/share/blueman	
2	PC/Server for control	-	-		
2-1	Install GTKterm	-	Refer to installation guide.		
2-2	Install python	-	Refer to installation guide.		
2-3	Install Node-RED	-	Refer to installation guide.		

Connectivity test

No	Test Case	Test input	Test Procedure	Expected output	Test result
1	MSM to IoT Gateway	-	<p>For more details, refer to installation guide.</p> <p>The following commands are executed from IoT gateway terminal.</p> <ol style="list-style-type: none"> 1. bluetoothctl 2. power on 3. scan on You can detect MSM and its address. 4. scan off 5. exit 6. sudo rfcomm bind <serial port#> <address> 7. rfcomm show 0 8. ls -l /dev/rfcomm* 9. gtkterm -p </dev/rfcomm#> -s 1000000 	<p>The gtkterm shows the following message.</p> 	
2	MSM to IoT Gateway (Node-RED)	-	<p>Create Node-RED flow in IoT gateway.</p> <ol style="list-style-type: none"> 1. Run Node-RED in IoT Gateway 2. Add "serial in" with baud rate=1Mbps and port which connected to MSM. 3. Add "debug". 4. Connect the "serial in" and the "debug". 5. Deploy <p>Refer to installation guide.</p>	<p>The Node-RED shows the following message.</p> 	
3	MSM to PC/Server for control	-	<p>Execute the following commands in PC/Server for control.</p> <pre>gtkterm -p <port name> -s 1000000</pre> <p>*The port name is port which is connected to MSM via USB cable.</p>		

4	MSM to PC/Server for control (Node-RED)		<p>Create Node-RED flow in PC/Server for control.</p> <ol style="list-style-type: none"> 1. Run Node-RED in PC/Server for control 2. Add "serial in" with baud rate=1Mbps and port which connected to MSM. 3. Add "debug". 4. Connect the "serial in" and the "debug". 5. Deploy 	<p>The Node-RED shows the following message.</p> 	
5	IoT Gateway to TE		<p>Create Node-RED flow in IoT gateway.</p> <ol style="list-style-type: none"> 1. Run Node-RED in IoT Gateway 2. Add "serial in" with baud rate=1Mbps and port which connected to MSM. 3. Add "debug". 4. Connect the "serial in" and the "debug". 5. Add "UDP out" with TE IP address and port. 6. Connect the "serial in" and the "UDP out". 7. Deploy <p>Create Node-RED flow in TE.</p> <ol style="list-style-type: none"> 8. Run Node-RED in Test equipment. 9. Add "UDP in" with port which specified in the above flow. 10. Add "debug". 11. Connect the "UDP in" and the "debug". 12. Deploy 	<p>The Node-RED in TE shows the following message.</p> 	
6	PC/Server for control to TE		<p>Create Node-RED flow in PC/Server for control.</p> <ol style="list-style-type: none"> 1. Run Node-RED in PC/Server. 2. Add "serial in" with baud rate=1Mbps and port which connected to MSM. 3. Add "debug". 4. Connect the "serial in" and the "debug". 5. Add "UDP out" with TE IP address and port. 	<p>The Node-RED in TE shows the following message.</p> 	

			<ol style="list-style-type: none">6. Connect the "serial in" and the "UDP out".7. Deploy <p>Create Node-RED flow in TE.</p> <ol style="list-style-type: none">8. Run Node-RED in Test equipment.9. Add "UDP in" with port which specified in the above flow.10. Add "debug".11. Connect the "UDP in" and the "debug".12. Deploy		
--	--	--	--	--	--

Blueval test

No	Test Case	Test input	Test Procedure	Expected output	Test result
1	IoT gateway Layer:OS	-	<p>1. Clone the validation repo in TE. git clone http://gerrit.akraino.org/r/validation</p> <p>2. Fill the followings in volumes.yaml file. Location to the customized blueprint file Location to where to store the results</p> <p>3. Update variables.yaml Fill in the file with your confidential information like IP address/username/passwords and environment specific information.</p> <p>4. Run the tests. bash validation/bluval/blucon.sh [-l <LAYER>] [-o] [-n host] <Blueprint Name></p>	-	
2	PC/Server for control Layer:OS	-	<p>5. Clone the validation repo in TE. git clone http://gerrit.akraino.org/r/validation</p> <p>6. Fill the followings in volumes.yaml file. Location to the customized blueprint file Location to where to store the results</p> <p>7. Update variables.yaml Fill in the file with your confidential information like IP address/username/passwords and environment specific information.</p>	-	

			8. Run the tests. bash validation/bluval/blucon.sh [-l <LAYER>] [-o] [-n host] <Blueprint Name>		
--	--	--	---	--	--

4 Revision history

Version	Date	Editor	Contents
0.1	02/07/2022	Fukano	Draft version
1.0	02/10/2022	Fukano	Review completed and published as first edition
1.1	03/04/2022	Inoue	Minor modifications to procedures
1.2	03/23/2022	Inoue	Write test result