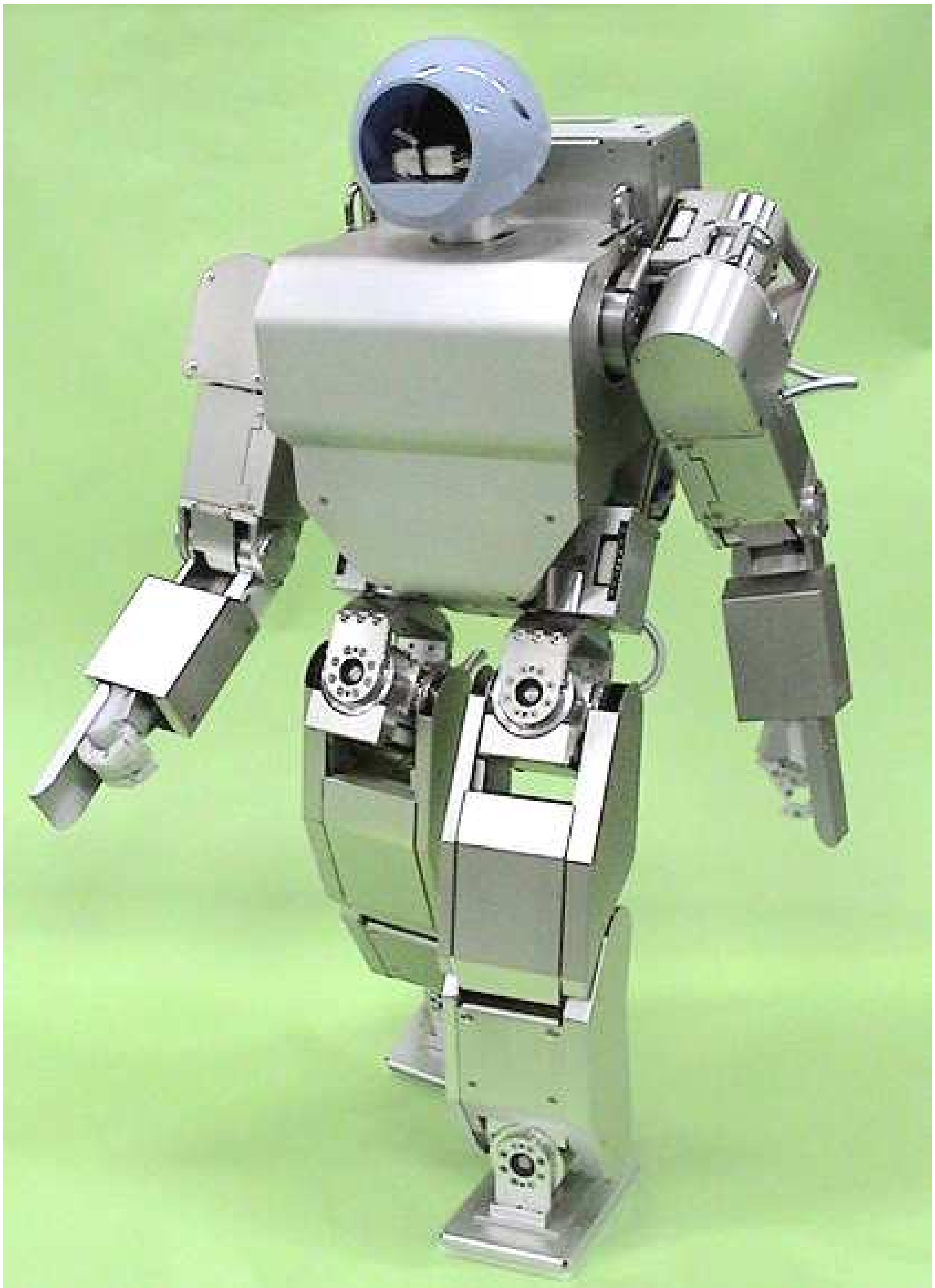
MINIATURE HUMANOID ROBOT

「HOAP-2」INSTRUCTION MANUAL



The third edition

Fujitsu Automation Co., Ltd.

						TITLE HOAP-2 INSTRUCTION MANUAL
03	041203	Kimura		Aoyama	S1.5 Add. tension data	DRAW. NO.
02	041115	Kimura		Aoyama	Add. the detail explanation	P1PW09076－A001
EDIT.	DATE	DESIG.	CHECK	APPR.	DESCRIPTION	FUJITSU AUTOMATION LIMITED / SHEET
DESIG.	040326	Kimura	CHECK		APPR. Aoyama	

CONTENTS

Appendix

１．Outline

Weighing 7kg and Standing 50cm tall, the HOAP−2 is “out of the box ready to go”and comes complete for wide application in research and development of robotic technologies. Full disclosure of hardware and software interface information makes HOAP−2 a complete, open architecture robot enabling anyone to develop their own software algorithms.

Programs developed by users are run on the RT−Linux operating system located on the command PC. Communication with the Robot body is made through a USB interface. Sensors and actuators inside the Robot also use a USB interface making it easy for their expansion capability.

２. Contents

Qty	Description	Model
1	Robot body	ＰＷ０９０７６－Ｂ００２
1	joint initial setting jig	ＰＷ０９０７６－Ｄ９１０
1	Lifting jig	ＰＷ０９０７６－Ｄ９２０
1	External power supply	ＰＷ０７１１１－Ｄ３０４
1	CD-R	Ｍ３ＰＷ０９０７６－Ｂ００２
1	Instruction Manual	Ｐ１ＰＷ０９０７６－Ａ００１
1	Command PC	ＬＯＰＷ０７１１１－１１０７

３．System Configuration

３． １ Overall System drawing

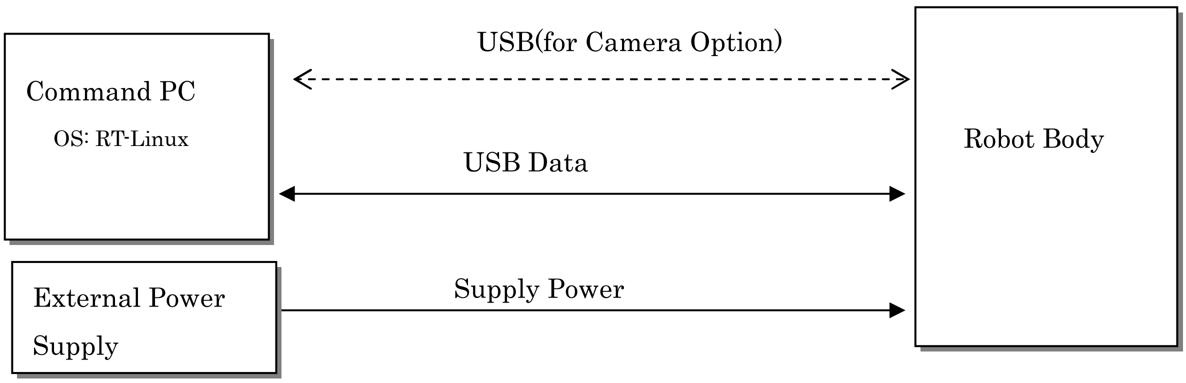
Drw. 3.1-1 and 3.1-2. illustrate communications and power connections in the wired, and wireless configuration, respectively. The Robot system is available in the wired mode and optional wireless mode. In the wired mode, high speed, real time communication is possible using USB interface between the Command PC and the Robot body. 24V power supply is used for the robot body. Using RT-Linux for the command PC OS, real time feed back control is realized using USB.

（＊） RT-Linux should be installed by user side. Refer to Appendix S3.1 installation method and operation circumstance

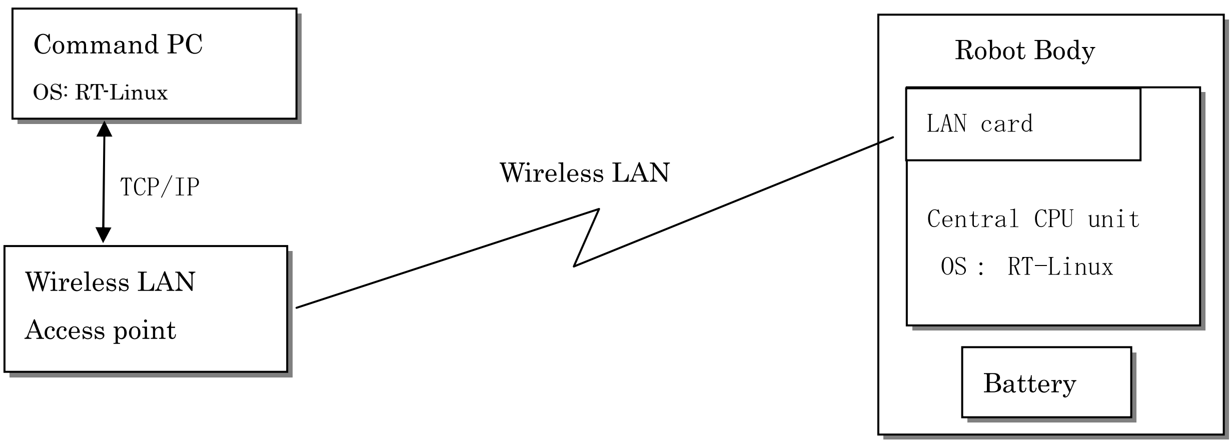
（＊＊）The USB cable of the camera option is connected to the USB port that is different route from the data communication port.

If it is connected to the same port , this robot cannot move.

Caution : There is a PC which does not have some USB ports.



Drw.3.1-1 Wired Mode system configuration



Drw.3.1-2 Wireless Mode system configuration (***)

（＊＊＊）Wireless option

３．２　Connection method

3.2.1 Wired Mode Connection Procedure

① Confirm two power supply switches are turned to off on robot.

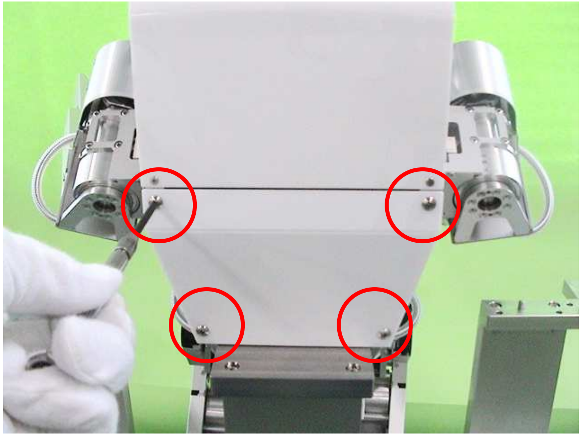
② Connect command PC and Robot body with USB cable.
The connection point is shown at Drw.S1.3-4, “Robot USB port”.
On the command PC side, a standard USB connector is attached to the USB port on the back of PC.

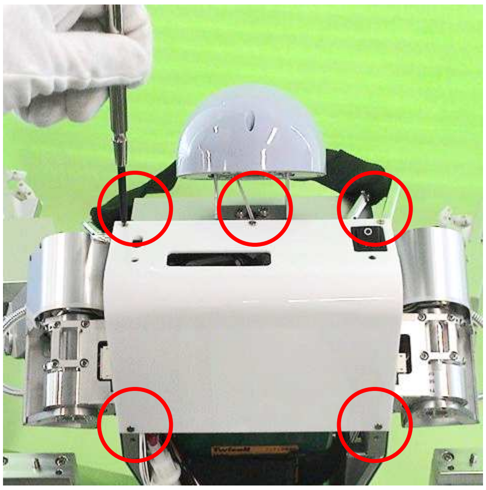
③ Connect the power supply cable with the robot body to the power supply.
See to Drw.S1.3-6 “Outside power supply input port”.
The connector on power supply unit is installed on the front side of the unit.

④ Finally, connect to power supply unit to a regular AC receptacle.

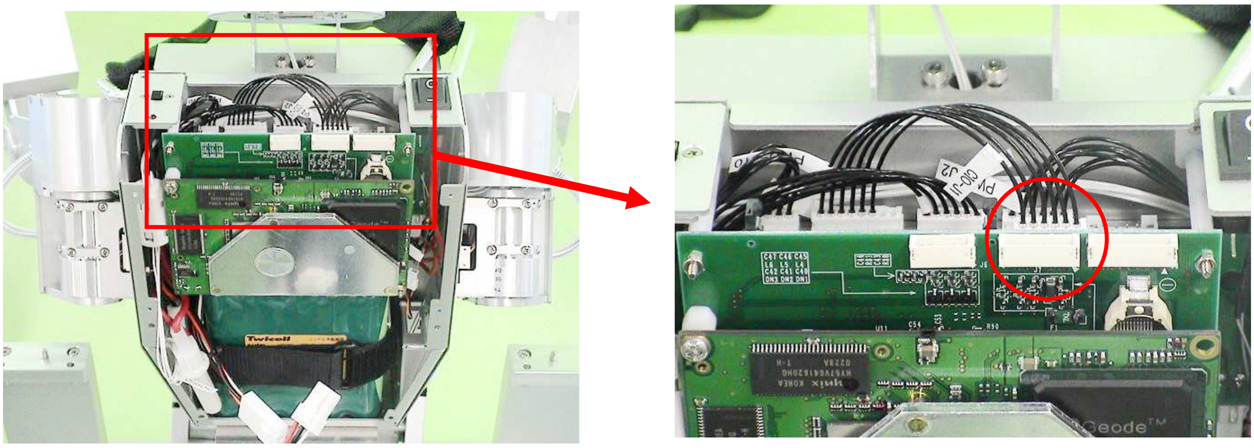
3.2.2 Wireless mode Connection Procedure (Need the Wireless Option)

① Confirm two power supply switches are turned to off on robot.
Open the both back covers.

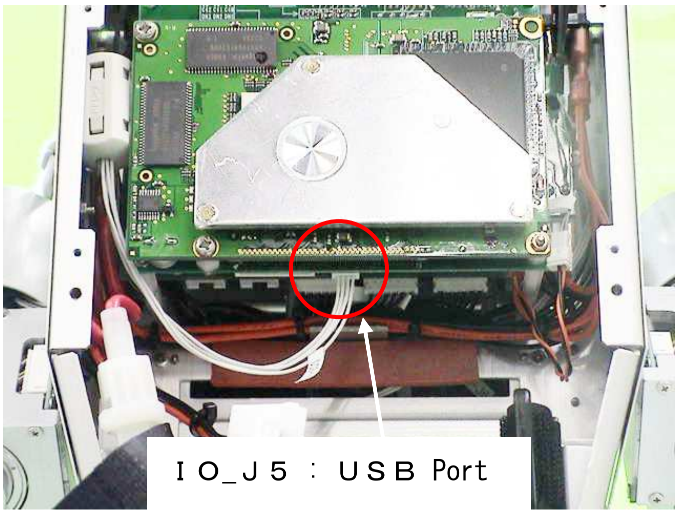




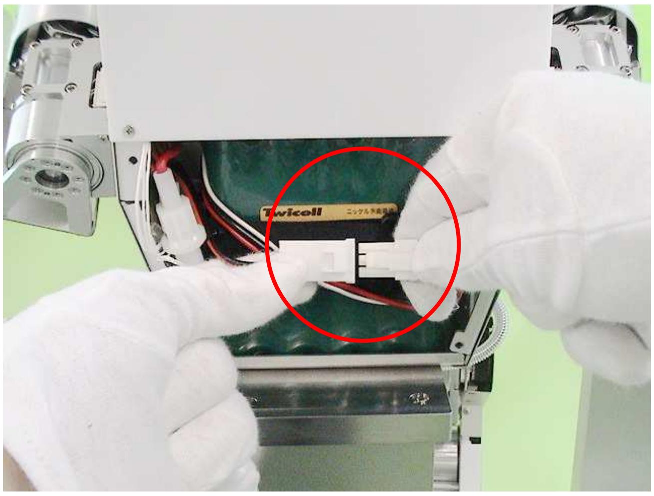
② Connect the Control CPU unit via 6 pin cable to **IO_J1** on the IO expansion board A.



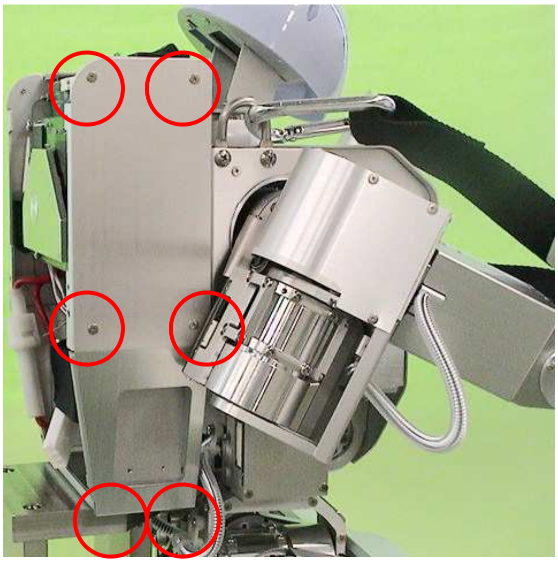
③ Remove the 4 pin USB cable relay connection , and connect to **IO_J5** on IO expansion board A



④ Battery connector refers to s. 1. 4 (should be charged).



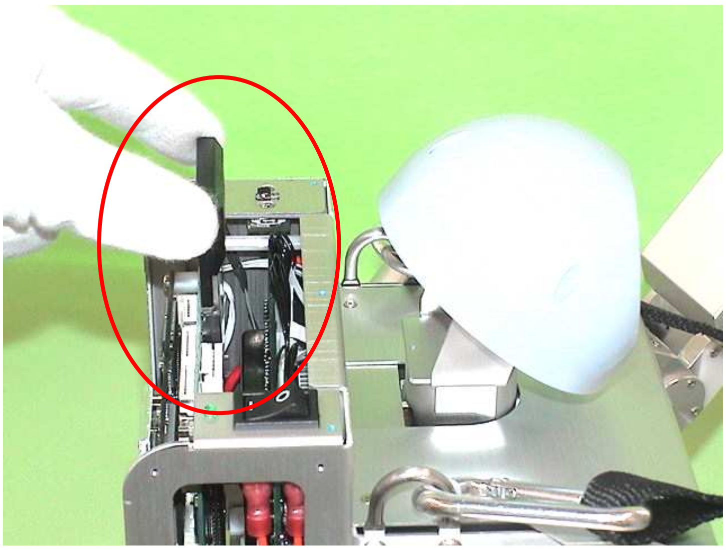
⑤ Install compact flash (CF) card into IO_J13 on IO expansion board A.



Open the side cover.

IO_J13：CFcard　Port

⑥ Install CF Wireless LAN card onto IO_J10 on IO expansion board A.



⑦ Command PC and Wireless access point (*) connect to network(**).

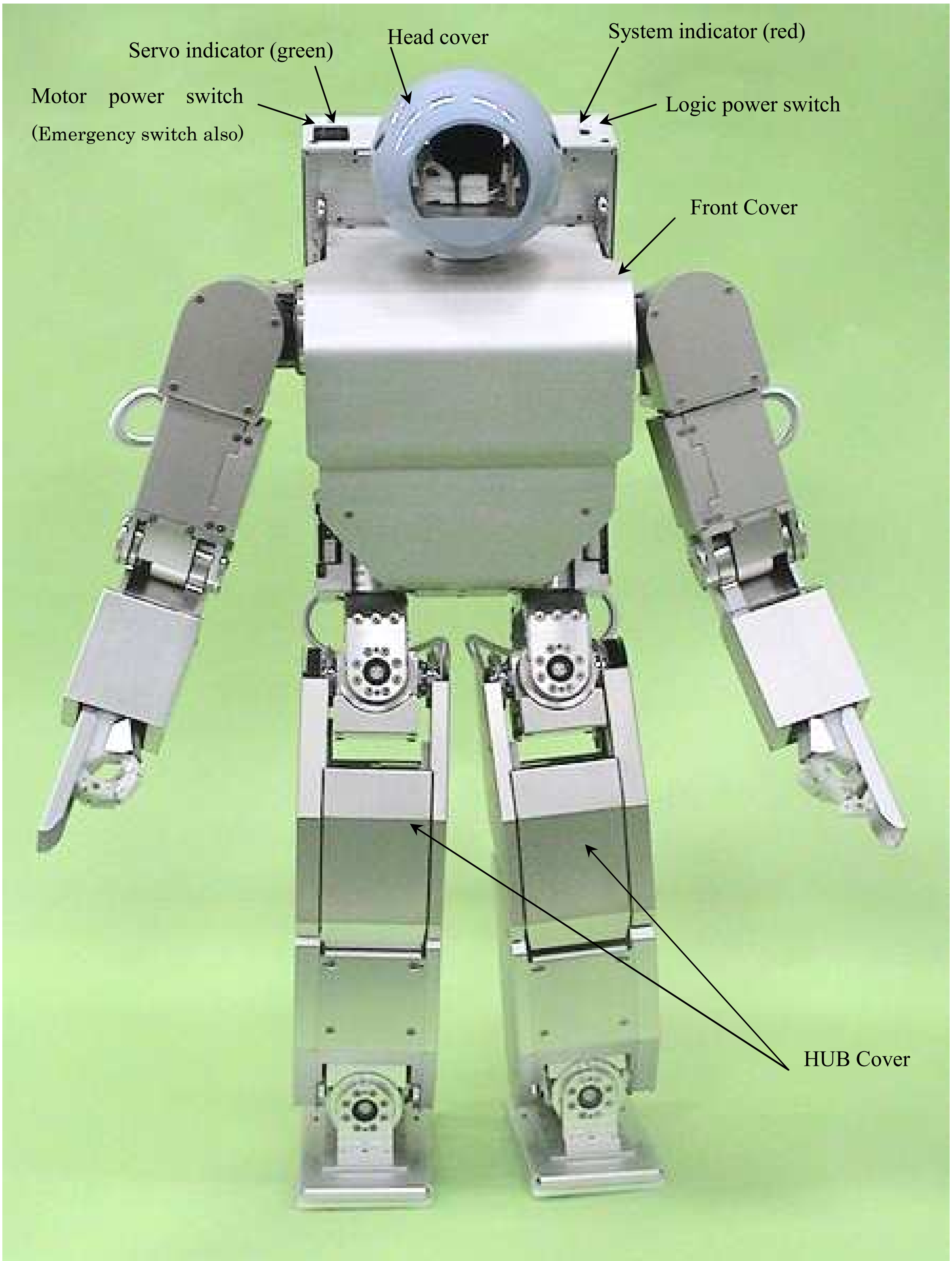
* Refer to instruction manual for Wireless access point for detail.

** Before shipment, IP address of central control is tentatively set in CF card. To avoid network interference, connect by local network by robot and command PC.

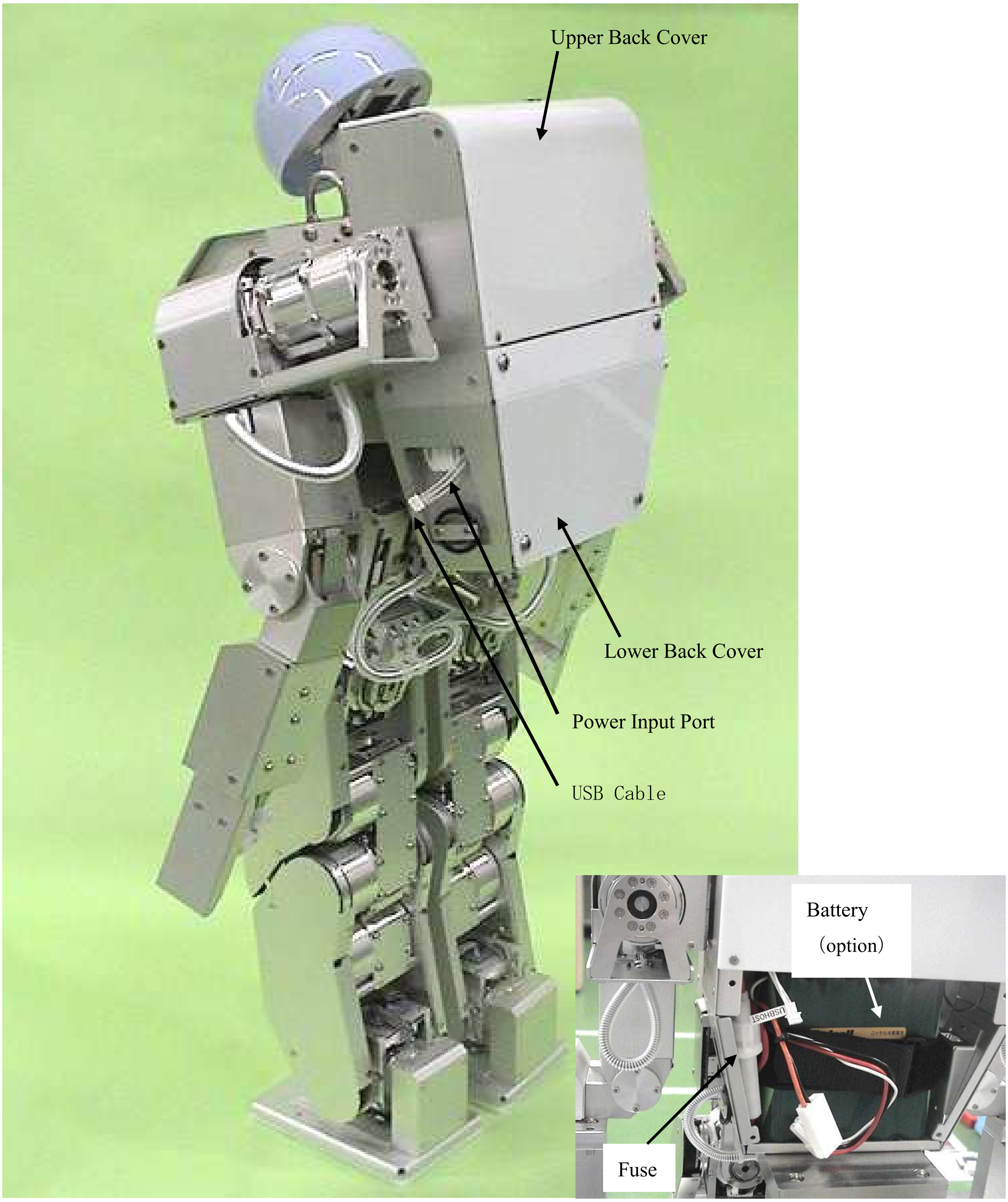
Note: When returning to wired mode, remove the cable to the IO expansion board A (IO_J1) and return the USB cable to wire mode.

3.3 Major parts position

Refer to Drw. 3.3-1, 3.3-2 to indicate position of major parts.



Drw.3.3-1. Robot body front



Inside of Lower Back Cover

Drw.3.3-2. Robot back side

4 Cautionary Items

Please read this "Cautionary Items" before use this robot system.

Please be sure to keep it at the place seen always.

4．1 Precautions and handling

HOAP-2 is a highly advanced programming unit with many degrees of freedom joints. Please follow all instructions and cautionary items to avoid breakage,other troubles or possible operator injury.

Although classified into "danger", "warning", and "cautions",please be sure to follow notes shown here, since the important contents with which all are related safely are indicated.

Danger : That it is assumed to be that a risk of a user getting death or seriously injured draws near and arises when handling is mistaken.

Warning : That a possibility that a user will get death or seriously injured is assumed to be when handling is mistaken.

Cautions: The thing it is assumed to be that a user gets a serious injury or slightly injured when handling is mistaken, and the thing which material damage generates. Or when handling is mistaken and it is assumed that the quality and the reliability of a product are spoiled.

4.1.1 Power supply and battery

Danger

・ Do not use non-standard voltage.

・ Do not use non-specified battery charger.

It becomes a liquid leak and the cause which makes it generate heat and explode about a battery.

・ Do not leave unattended during charging.

・ A battery is not thrown in in fire and it does not heat.

・The (+) and (-) terminal of a batteryare not connected with metal, such as wire. Moreover, it does not carry together with a metal necklace, a hairpin, etc.

・A battery is not disassembled and converted. Generation of heat, ignition, and the liquid of strong alkalinity emit and are dangerous.

・Don't solder to a battery directly.

・As for the battery, it opts for direction of plus and minus.

It does not connect by force. Please confirm direction of plus and minus.

・This battery is only use for HOAP robot. Don't use this battery to another use.

・When the liquid of a battery goes into an eye, wash enough with beautiful water immediately and undergo a doctor's medical treatment.

Warning

- A battery is not attached to water, sea water, etc.A terminal portion is not wet. It becomes the cause of heat and the rust of a terminal etc.
- Charge is stopped, when charge is not completed, even if it exceeds predetermined charge.
- Since injury may be caused on the skin when the liquid of a battery adheres to the skin or clothes, please wash away with beautiful water immediately.
- A battery coating tube is not removed or a crack is not given. It becomes a liquid leak and the cause which makes it generate heat and explode about a battery.
- Don't use the battery, when a battery carries out a liquid leak or it has noticed differing from discoloration.
- Don't use the battery and leave in hot places, such as a strong place of direct rays, in the car, and the front of a stove etc.

Caution

- Don't give a shock strong against a battery or don't throw it.
- Be sure to perform use (electric discharge) of a battery in 0~50℃.
- Charge the battery in 0～40℃.
- Keep a battery in a place with little humidity of 0～30℃.
- Don't charge in having got the battery cold and cold outdoors (0℃ or less) .
- When robot is not in operation, remove the battery and make sure that all power switches are in the off position to avoid battery discharge.
- When not in use, make sure that power switch is in the OFF position
- Don't use the battery in unusual.

<Abandonment of a battery>

Used nickel hydoride batteries are precious resources.
bring at a battery recycling cooperation store after sticking a tape on a terminal or a connection code and insulating.

4.1.2 Main Body

Danger

- Do not touch and keep a way inside of robot operation range.
  (for emergency stop, switch off motor power in front side)
- Keep clear when servo motors are turned ON.
- Be careful hands not to be caught between activated portion.
- Don't use a robot in an area known to have an electro-magnetic interference, static electricity discharge or possible radio frequency interference.
- The robot is likely to experience faulty operation and can be dangerous to operate under such conditions.
- Set up the joint initial setting in the right position, If it does not do so, the robot may do unusual operation and this will cause damage to the robot.

Warning

- Robot is designed to be operated in a clean, or laboratory environment.

  Do not attempt to use where dampness, dust, or unfavorable temperature conditions are present.
- The motor can become hot during operation.

  Make sure that motor power is off and motors are cooled properly prior to touching
- Do not use robot in a fashion where excessive torque is applied to joints, or beyond the allowable joint torque. This includes manually moving joints. This may cause damage to the robot.

Caution

- Tension on timing belts located in the legs should be maintained as described in the appendix. Excessive overweight can cause belts to lose tension.
- If it is left in the state where a motor does not arrive at a target position, a motor will become high temperature and this will cause damage to the robot.

4.1.3 Programming

Danger

- Knowledge of RT-Linux and robot movement principles will make programming of the robot easier
- Make sure that programmed moves are made relative to the soft-limit when developing your program
- The optimum joint motion is accomplished by using minute steps. When using small steps, the robot makes smooth movements. It is very dangerous when the movement command value between each step is increased too much. When a movement command is issued, the servos will move from current position to the newly commanded position at high speed. If the new position is greatly different from the current position it can result in a dangerous situation. It is recommended, and is the responsibility of the user to develop safeguards in their software that, within the boundary of their experiments, will reject commands that instruct movements which greatly vary from the current range and position.
- It is very dangerous to use this robot with speed control mode.

  Consult our company when you use speed control mode.

４．２ Warranty

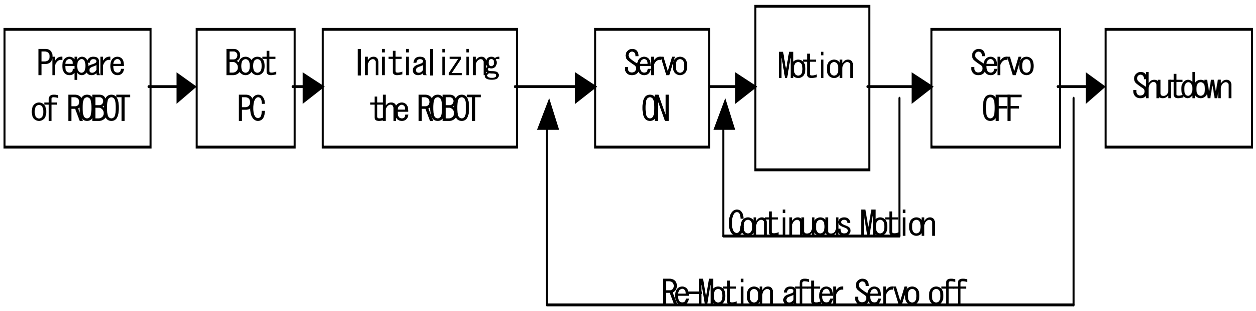
It is chargeable except for the initial defect.

This robot's appearance and specification may be changed without a preliminary announcement.

5. Confirmation of Operation

5.1 Operation Flow

The steps to operate the robot are shown in the following process flow chart. These steps are necessary in order to initialize the robot parameters before an operation begins.



5.2 Example of How to Run a Sample Motion Program

The following explains a sample operation procedure for the wired mode. This is intended to simplify operation for the user who uses a robot for the first time. Please take time to get thoroughly accustomed to these operating procedures of the robot.

5.2.1Prepare of the ROBOT

(1) Confirm that the logic power switch and the motor power switch are off.

a. It is very important to note that when powering down the robot you should always turn off the motor power switch first. After motor power is turned off, it is then safe to turn off the logic power switch.

b. Conversely, when powering up the robot, you should always turn on the logic power switch first. After the logic power switch is on, it is then safe to turn on the motor power switch.

c. Failure to power up, and power down the robot according to the above procedure could result in one or all of the following:

Unexpected movement of the robot

Damage to the robot systems

Possible injury to the operator

(2) Confirm that the external power supply is off. External power should always be turned off when connecting, or disconnecting the robot.

(3) Confirm that the **robot is fixed on the joint initial setting jig**.
Refer to S .1.2 for the method of the fixation

(4) Connect the USB cable between the robot and the USB port behind the command PC.

(5) Connect the 24VDC power supply cable between robot and the external power supply.

Attention : Make sure that the 24VDC power supply is off during this connection process.

5.2.2 Boot the command PC

(6) Turn On the PC.

(7) After Red Hat Linux starts, you will see the following prompt: and input following.

a. local host login : **root (Return)**

b. password : **default (Return)**

(8) Start terminal emulator for GNOME.

Input: #**startx (Return)**

(9) Start to『terminal emulator for GNOME』

(10) Change directory.

Input #**cd /usr/local/hoap2/modules (Return)**

(11) RT-Linux is started.

Input #**rtlinux start rt_ctlmodule.o (Return)**

When RT-Linux has started normally, the following will be displayed (+).

(+) rt_ctlmodule

:

If it is displayed (—) , Reboot the PC and restart from (6) .

(12) It is input as follows, and a communication module with the robot is started

\# **../bin/rt_ctlapp (Return)**

(13) Turn on the logic and motor power switches of the robot.

(14) Reset the USB devices. Input the "**r**" command.

0,00,00$ **r (Return)**

(15) If it is continuing pushing a Return key , the command PC starts the recognition of the USB device.

Repeat it until it is indicated below.

5, 25, xx $ (xx isn't fixed)

5.2.3 Initialing the Robot

(16) Start another terminal emulator for GNOME.

This emulator window is named The window 2 , and the window currently opened from before is named a window 1.

(17) In the window2, input as follows, and change directory to /usr/local/hoap/data .

#**cd /usr/local/hoap2/data (Return)**

(18) Confirm that the robot is exactly sitting the initial setting jig.

If the sitting position is not exactly , it is dangerous since the initial position of an encoder shifts correctly ,and normal operation becomes impossible

(19) The initial data (gain, movement range, control mode, encoder reset etc.) is set .

Input the following in the window 2.

#**../bin/sendseq < HomeReset2a.csv (Return)**

Refer to Appendix 2.2 for the details.

(20) Input the following in the window 1.

5,25,xx $ V (Return)

Confirm that it is same value csep value = nseq value.

(21) Estrange it from the initial setting jig , and hang a robot at the robot lifting jig.

Refer to Appendix 1.1. A robot's posture should extend hand and foot lightly.

5.2.4 Servo on

(22) The present encoder counter of each joint is acquired.

By Window 2, input as

```
#../bin/setpos (Return)
```

(*)Joint angle shouldn't be shifted by hand absolutely after this command transmission.

If a joint is moved, the target position value and the present position will arise.

It is danger since it will move to a target position in an instant if “Servo on” is carried out as it is. When you have moved, please perform “setpos” again.

(23) All the joints are set Servo on .

Input following by Window 2

```
#../bin/sendseq < AllServoOn.csv (Return)
```

Please move joints by hand and check the servo on.

5.2.5 Motion

(24) It is generated the file for making it shift to a starting point from the present position . Here, it is generated the file which shifts to the instruction value of the beginning of sample walk software (m01.csv) .

Input following by Window 2

```
#../bin/interpol < m01.csv (Return)
```

(25) Move to the motion starting position

The shift file(interpol.csv) created above is performed.

If this command is executed, since a robot will begin to move slowly, please do not approach a robot by any means or do not put a thing on near.

Input following by Windorw2

```
#../bin/sendseq < interpol.csv (Return)
```

Please check that a robot's joint has moved to the start position.

(26) This robot begins a sample walk movement .

Input following by Window 2

```
#../bin/sendseq < m01.csv (Return)
```

After the end of robot motion, choose a) or b) below.

a) When the same motion is performed again, Execute from (24).

b) When the operation is ended or when Servo off is executed at once, Go to “5.2.6 Servo off”.

5.2.6 Servo off

(27) Servo off.

Hung the robot in the lifting jig.

When this command will be executed, all motors will be servo off.

Input following by Window 2

```
#./bin/sendseq < AllServoOff.csv (Return)
```

In order to operate a robot again without shutting off a power supply,

Execute from (22) setpos.

5.2.7 Power off

(28) Turn off the logic power switch and the motor power switch.

(29) Turn off the External power supply switch.

(30) Close Window 2.

(31) Input following by Window 1.

```
5.25.xx$ q (Return)
```

(32) Stop RT-Linux.

```
# rtlinux stop rt_ctlmodule.o
```

(33) Shutdown linux.

```
# shutdown -h now
```

(34) Turn off the power supply of the command PC.

[Wireless mode]

Except for remote accessing from an instruction personal computer of operation, it becomes the same operation as the above using wireless LAN.

For the connection method, refer to "3.2.2 Wireless mode Connection Procedure". and for the operation method, refer to"S3.2 Wireless motion process".

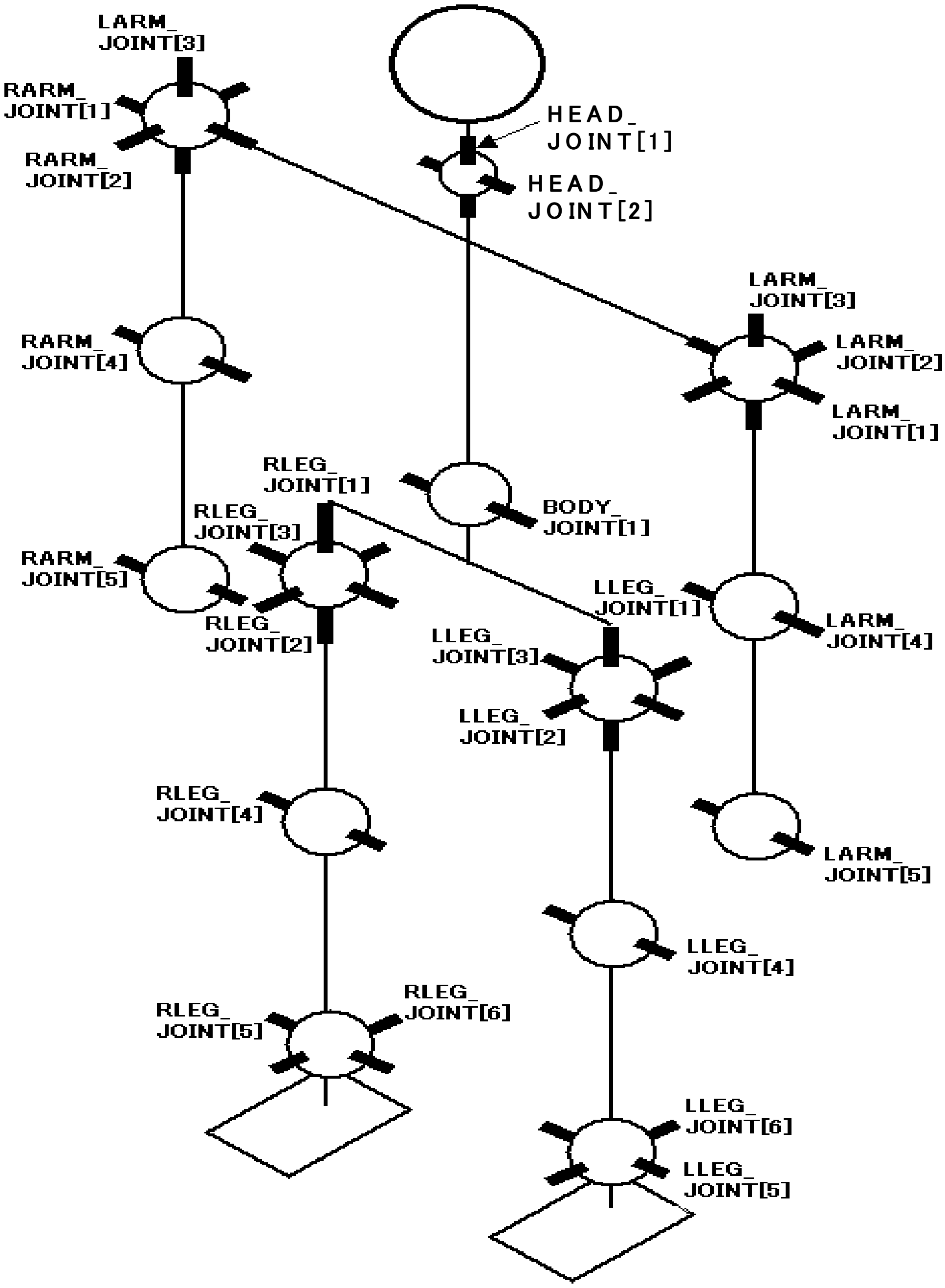
6. Mechanical

6.1 Joint composition and the definition of the angle

This robot has twenty-five joints with two arms of 5 degree of freedom , two legs of 6 degree of freedom , the Head of 2 degree of freedom , and body joint. It is the degree of rotation freedom completely, and it is a cereal link in the leg and arm.

The name of the joint and freedom occasion distribution are tables 6.1-1.

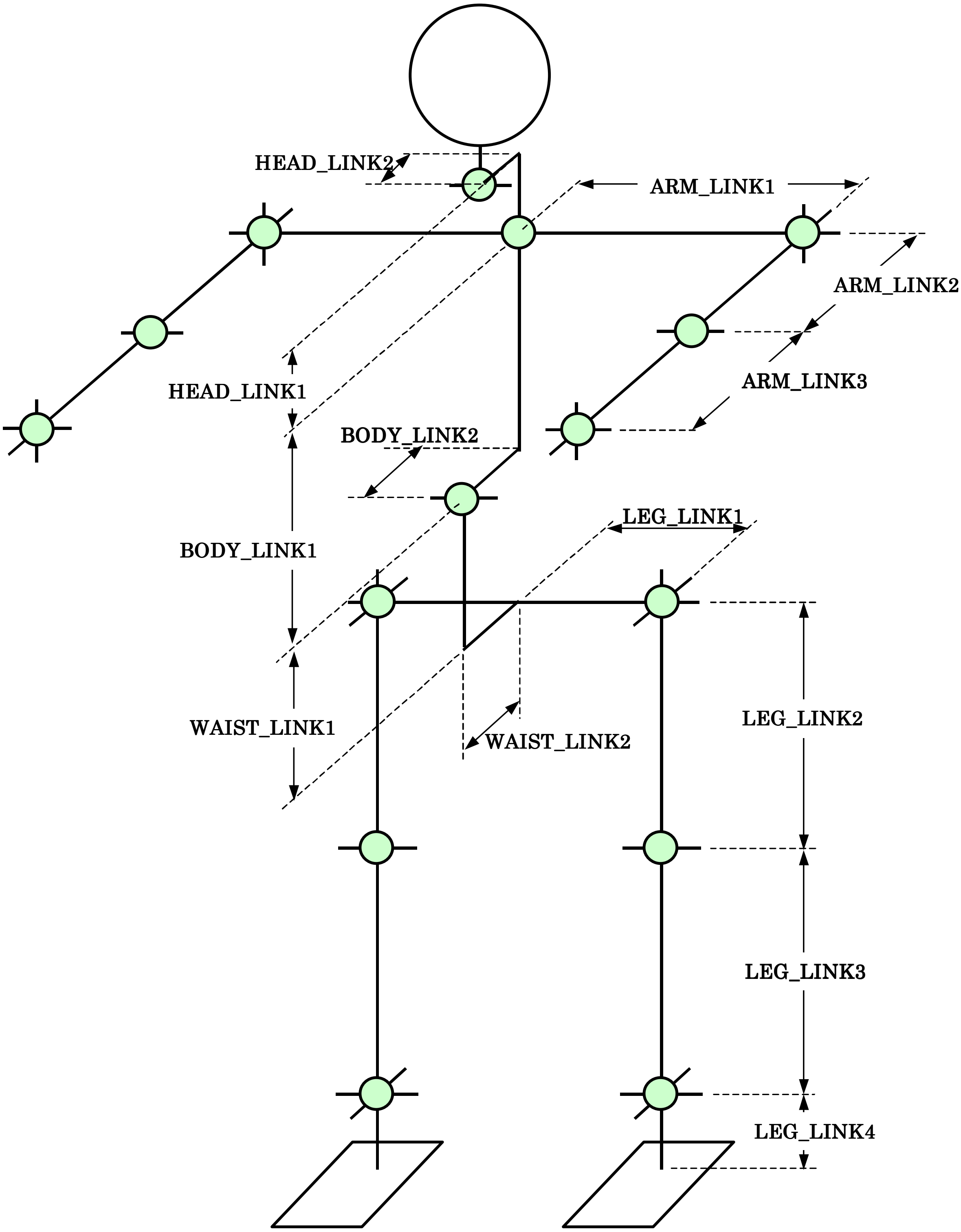
And, numerical value shows machine structure from the root like order connection order from 1 in [ ] of the joint name in the table. Moreover, the joint angle which the posture of the robot is decided as is to be defined in the link parameter of the Drw. 6.1-1,6.1.-2 and the table 6.1-2, the joint coordinate and the DH parameter, and **all the joint orders to the robot are to follow this definition**.



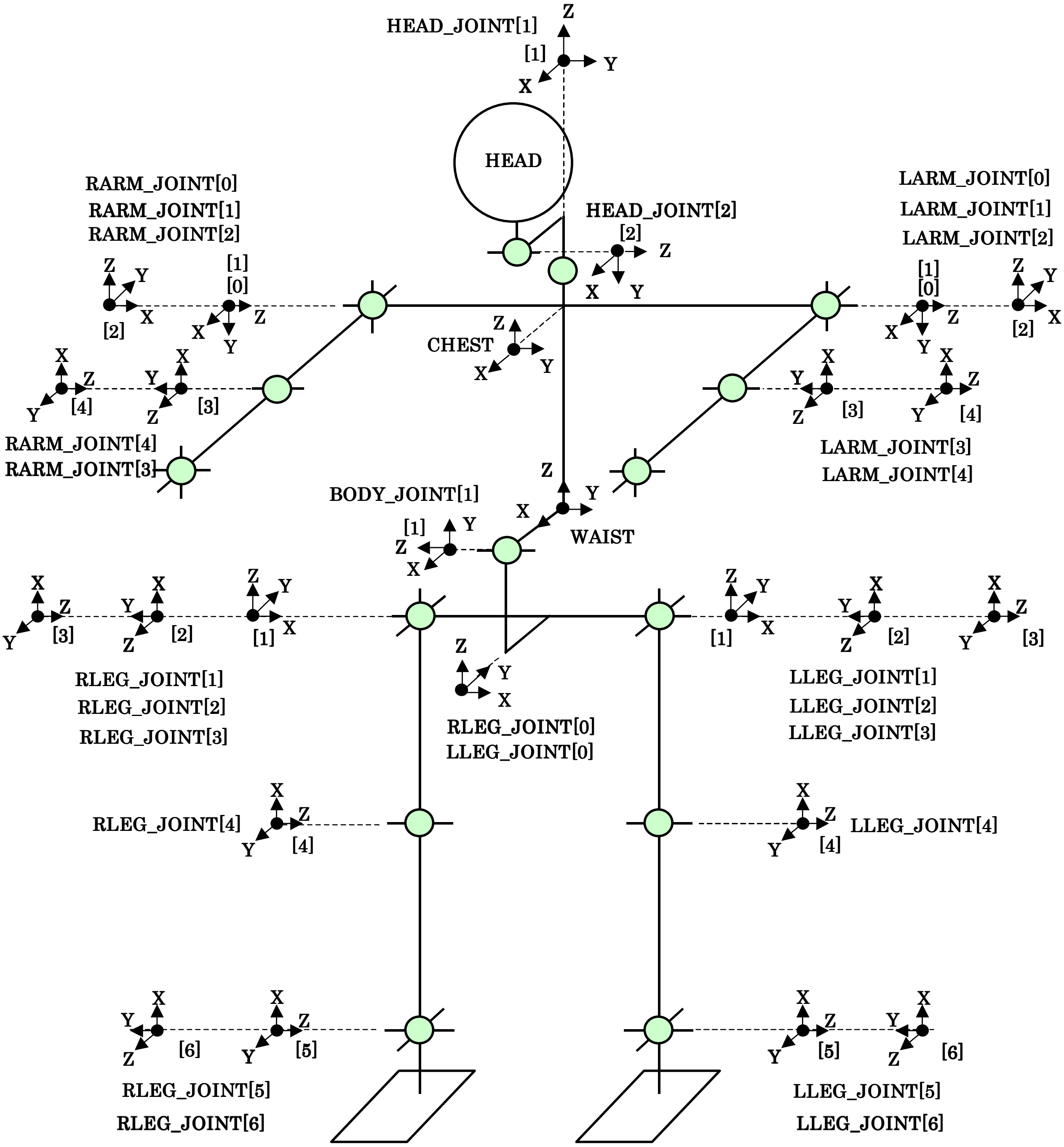
Drw.6.1-1 Joint name

Table6.1-1　Joint name

Joint name	Degree Of Freedom	Device ID
HEAD_JOINT[1]	Head pan	25(t)
HEAD_JOINT[2]	Head tilt	25(s)
BODY_JOINT[1]	waist	21
RLEG_JOINT[1]	The right thigh joint twisted.	1
RLEG_JOINT[2]	The right and left of the right thigh joint	2
RLEG_JOINT[3]	The front & rear of the right thigh joint	3
RLEG_JOINT[4]	right knee	4
RLEG_JOINT[5]	The front & rear the right ankle	5
RLEG_JOINT[6]	The right and left of the right leg ankle	6
RARM_JOINT[1]	The front & rear of the right shoulder	7
RARM_JOINT[2]	The right and left of the right shoulder	8
RARM_JOINT[3]	The right shoulder twisted.	9
RARM_JOINT[4]	Right elbow	10
RARM_JOINT[5]	Right hand	25(u)
LLEG_JOINT[1]	The left thigh joint twisted	11
LLEG_JOINT[2]	The right and left of the left thigh joint	12
LLEG_JOINT[3]	The front & rear of the right thigh joint	13
LLEG_JOINT[4]	Left knee	14
LLEG_JOINT[5]	The front & rear the left ankle	15
LLEG_JOINT[6]	The right and left of the left leg ankle	16
LARM_JOINT[1]	The front & rear of the left shoulder	17
LARM_JOINT[2]	The right and left of the left shoulder	18
LARM_JOINT[3]	The left shoulder　twisted	19
LARM_JOINT[4]	Left elbow	20
LARM_JOINT[5]	Left hand	25(v)



Drw6.1-2　The parameter definition of the link length



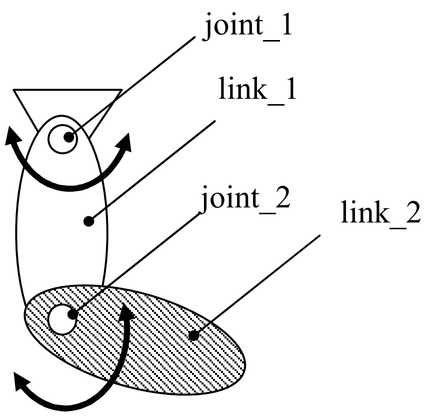
Drw.6.1-3 DH parameter definition coordinate（θ [i]=0 Posture）

Drw.6.1-2　　DH Parameter(θ [i]=0is Drw.6.1-2 Posture)

	a[I−1]	α[I−1]	d[i]	θ[i]
	(m)	(deg)	(m)	(deg)
CHEST	−BODY_LINK2	0	BODY_LINK1	0
HEAD_JOINT[1]	0	0	HEAD_LINK1	0
HEAD_JOINT[2]	HEAD_LINK2	−90	0	0
ARM_JOINT[0]	0	−90	(*)ARM_LINK1	0
ARM_JOINT[1]	0	0	0	θ[1]
ARM_JOINT[2]	0	90	0	θ[2]+90
ARM_JOINT[3]	0	90	ARM_LINK2	θ[3]+90
ARM_JOINT[4]	0	90	0	θ[4]
BODY_JOINT[1]	0	90	0	0
LEG_JOINT[0]	−WAIST_LINK2	−90	−WAIST_LINK1	90
LEG_JOINT[1]	(*)LEG_LINK1	0	0	θ[1]
LEG_JOINT[2]	0	90	0	θ[2]+90
LEG_JOINT[3]	0	90	0	θ[3]
LEG_JOINT[4]	−LEG_LINK2	0	0	θ[4]
LEG_JOINT[5]	−LEG_LINK3	0	0	θ[5]
LEG_JOINT[6]	0	−90	0	θ[6]
	(*)Take negative value at the time of the right half the body.			
	WAIST	−>	CHEST	by link
ARM_LINK1	0.0995	m		
ARM_LINK2	0.1010	m		
ARM_LINK3	0.1460	m		
LEG_LINK1	0.0390	m		
LEG_LINK2	0.1000	m		
LEG_LINK3	0.1000	m		
LEG_LINK4	0.0370	m		
BODY_LINK1	0.0900	m		
BODY_LINK2	0.0340	m		
HEAD_LINK1	0.0810	m		
HEAD_LINK2	0.0080	m		
WAIST_LINK1	0.0550	m		
WAIST_LINK2	0.0340	m		

6.2 Mass Property

mass property of the necessary link is in the attached sheet to simulate the dynamics of the robot. Mass property is defined in the link unit that it belongs to the degree of each joint freedom in accordance with the joint coordinate of the DH parameter .A link for example to belong to joint_1 in case as a ground plan becomes link_1, and mass property of link_1 by the coordinate of joint_1 is defined as it.



And, three kinds of mass property show it about the BODY link which is the body trunk of the robot by the difference in the loading thing of the cable mode and the wireless mode. Select and simulate mass property fitted to the configuration of an actual opportunity

6.3 The change of the joint angle and the velocity

The control of the joint angle of the robot and the velocity uses the thing that the position where it defines in the DH parameter, and a speed were changed into the increase and decrease value (velocity) of the pulse count value or the pulse count value (position) of the encoder of the joint and around the time as command value.It responds to the information on the position of the present joint and the speed as well by this value from the robot completely. Incremental encoder is being used for the encoder. Therefore, counter which becomes position command value must pre-set joint angle in the ordinary posture so that a change (or, reverse change) may be made in the joint angle absolute value defined in the DH parameter. It is installed in the method which a robot itself was mentioned about to S.1.2 concretely in the joint initialization jig, and pre-set counter of each joint by the value of the table 6.3-1 in this posture in the fixed value

Table6.3-1 Counter value to pre-set and a DH definition angle in the loading posture on the joint initialization jig

Joint name	Dhdefinition angle	Preset value		Motor type	Device ID
	(deg)	(decimal)	(hex)		
RLEG_JOINT[1]	0	0	0	TYPE-2	1
RLEG_JOINT[2]	0	0	0	TYPE-3	2
RLEG_JOINT[3]	-45	9405	24BD	TYPE-2	3
RLEG_JOINT[4]	90	18810	497A	TYPE-3	4
RLEG_JOINT[5]	-45	-9405	DB43	TYPE-2	5
RLEG_JOINT[6]	0	0	0	TYPE-2	6
RARM_JOINT[1]	90	18810	497A	TYPE-2	7
RARM_JOINT[2]	0	0	0	TYPE-2	8
RARM_JOINT[3]	90	-18810	B686	TYPE-2	9
RARM_JOINT[4]	-90	18810	497A	TYPE-2	10
LLEG_JOINT[1]	0	0	0	TYPE-2	11
LLEG_JOINT[2]	0	0	0	TYPE-3	12
LLEG_JOINT[3]	-45	-9405	DB43	TYPE-2	13
LLEG_JOINT[4]	90	-18810	B686	TYPE-3	14
LLEG_JOINT[5]	-45	9405	24BD	TYPE-2	15
LLEG_JOINT[6]	0	0	0	TYPE-2	16
LARM_JOINT[1]	90	-18810	B686	TYPE-2	17
LARM_JOINT[2]	0	0	0	TYPE-2	18
LARM_JOINT[3]	-90	18810	497A	TYPE-2	19
LARM_JOINT[4]	-90	-18810	B686	TYPE-2	20
BODY_JOINT[1]	0	0	0	TYPE-3	21

Table6.3-2　The change of the DH parameter definition joint angle and the position command value

θd	DH definition joint angle (deg)
P	position command value (pulse)
A	change coeficient (pulse/deg)
caution)	"P"is coded 2 byte integer
change expression :	**P=A×θd**
joint name	A(pulse/deg)
RLEG__JOINT[1]	209
RLEG__JOINT[2]	209
RLEG__JOINT[3]	−209
RLEG__JOINT[4]	209
RLEG__JOINT[5]	209
RLEG__JOINT[6]	−209
RARM__JOINT[1]	209
RARM__JOINT[2]	209
RARM__JOINT[3]	−209
RARM__JOINT[4]	−209
LLEG__JOINT[1]	209
LLEG__JOINT[2]	209
LLEG__JOINT[3]	209
LLEG__JOINT[4]	−209
LLEG__JOINT[5]	−209
LLEG__JOINT[6]	−209
LARM__JOINT[1]	−209
LARM__JOINT[2]	209
LARM__JOINT[3]	−209
LARM__JOINT[4]	209
BODY__JOINT[1]	209

Table6.3-3　The change of the DH parameter definition joint velocity and the velocity command value

ωd	DH definition joint angle (deg/s)
V	velocity command value (pulse/ms)
B	change coeficient (pulse/ms/(deg/s))
Caution)	"V" is coded 2byte integer
change expression :	**V=B×ωd**
joint name	B(pulse/ms/(deg/s))
RLEG__JOINT[1]	0.209
RLEG__JOINT[2]	0.209
RLEG__JOINT[3]	−0.209
RLEG__JOINT[4]	0.209
RLEG__JOINT[5]	0.209
RLEG__JOINT[6]	−0.209
RARM__JOINT[1]	0.209
RARM__JOINT[2]	0.209
RARM__JOINT[3]	−0.209
RARM__JOINT[4]	−0.209
LLEG__JOINT[1]	0.209
LLEG__JOINT[2]	0.209
LLEG__JOINT[3]	0.209
LLEG__JOINT[4]	−0.209
LLEG__JOINT[5]	−0.209
LLEG__JOINT[6]	−0.209
LARM__JOINT[1]	−0.209
LARM__JOINT[2]	0.209
LARM__JOINT[3]	−0.209
LARM__JOINT[4]	0.209
BODY__JOINT[1]	0.209

6.4 Joint allowable movement range

The joint movement range (DH parameter angle minimum angle - a maximum angle) that it is permitted after pre- sets joint counter of the robot becomes a table 6.4-1. Software limit of movement range can be set up in the robot. Be sure to set up software limit in less than this movement range by counter value for the prevention of damage and the safety. And, software limit is set up by the maximum and minimum of counter value. Over shoot of the control to the range of a joint angle to use in the actual practical use, and take the inside of 1deg of the maximum-minimum value of the table 6.4-1.

Sample: The utility movement ranges of RLEG_JOINT [1] become -90 - 30deg.

Table 6.4-1 Joint allowable movement range

(counter value is value with a thing after the pre-set completion.)

Joint name	DH Parameter angle min.		DH Parameter angle MAX		motor type	Device ID
	DH angle (deg)	Counter (decimal)	DH angle (deg)	Counter (decimal)		
RLEG_JOINT[1]	−91	−19019	31	6479	TYPE−2	1
RLEG_JOINT[2]	※ −31	−6479	※ 21	4389	TYPE−3	2
RLEG_JOINT[3]	−82	17138	71	−14839	TYPE−2	3
RLEG_JOINT[4]	−1	−209	130	27170	TYPE−3	4
RLEG_JOINT[5]	※ −61	−12749	※ 61	12749	TYPE−2	5
RLEG_JOINT[6]	−25	5225	25	−5225	TYPE−2	6
RARM_JOINT[1]	−91	−19019	151	31559	TYPE−2	7
RARM_JOINT[2]	−96	−20064	1	209	TYPE−2	8
RARM_JOINT[3]	−91	19019	91	−19019	TYPE−2	9
RARM_JOINT[4]	−115	24035	1	−209	TYPE−2	10
LLEG_JOINT[1]	※ −31	−6479	※ 91	19019	TYPE−2	11
LLEG_JOINT[2]	−21	−4389	31	6479	TYPE−3	12
LLEG_JOINT[3]	−82	−17138	71	14839	TYPE−2	13
LLEG_JOINT[4]	−1	209	130	−27170	TYPE−3	14
LLEG_JOINT[5]	※ −61	12749	※ 61	−12749	TYPE−2	15
LLEG_JOINT[6]	−25	5225	25	−5225	TYPE−2	16
LARM_JOINT[1]	−91	19019	151	−31559	TYPE−2	17
LARM_JOINT[2]	−1	−209	96	20064	TYPE−2	18
LARM_JOINT[3]	−91	19019	91	−19019	TYPE−2	19
LARM_JOINT[4]	−115	−24035	1	209	TYPE−2	20
BODY_JOINT[1]	−1	209	90	−18810	TYPE−3	21
RARM_JOINT[5]	−60	−	60	−	RC−Servo	25
LARM_JOINT[5]	−60	−	60	−	RC−Servo	25
HEAD_JOINT[1]	−60	−	60	−	RC−Servo	25
HEAD_JOINT[2]	−15	−	60	−	RC−Servo	25

※ The movement range of these joints may decrease, because of overlap.

6.5 Joint allowable torque

The allowable torque of the joint of the robot shows a table 6.5-1. Be careful not to exceed this value when the back drive operation of the joint by manual in direct acting servo off or the time of the movement of the robot.

Table 6.5-1 Joint Allowable torque

Item	Joint name	Joint Allowable torque [N·m]		motor type
		rating	moment	
1	RLEG_JOINT[1]	1.5	3	TYPE-2
2	RLEG_JOINT[2]	2	4.5	TYPE-3
3	RLEG_JOINT[3]	1.5	3	TYPE-2
4	RLEG_JOINT[4]	2	4.5	TYPE-3
5	RLEG_JOINT[5]	1.5	3	TYPE-2
6	RLEG_JOINT[6]	1.5	3	TYPE-2
7	RARM_JOINT[1]	1.5	3	TYPE-2
8	RARM_JOINT[2]	1.5	3	TYPE-2
9	RARM_JOINT[3]	1.5	3	TYPE-2
10	RARM_JOINT[4]	1.5	3	TYPE-2
11	LLEG_JOINT[1]	1.5	3	TYPE-2
12	LLEG_JOINT[2]	2	4.5	TYPE-3
13	LLEG_JOINT[3]	1.5	3	TYPE-2
14	LLEG_JOINT[4]	2	4.5	TYPE-3
15	LLEG_JOINT[5]	1.5	3	TYPE-2
16	LLEG_JOINT[6]	1.5	3	TYPE-2
17	LARM_JOINT[1]	1.5	3	TYPE-2
18	LARM_JOINT[2]	1.5	3	TYPE-2
19	LARM_JOINT[3]	1.5	3	TYPE-2
20	LARM_JOINT[4]	1.5	3	TYPE-2
21	BODY_JOINT[1]	2	4.5	TYPE-3
22	RARM_JOINT[5]	0.28	0.36	Rcservo
23	LARM_JOINT[5]	0.28	0.36	Rcservo
24	HEAD_JOINT[1]	0.28	0.36	Rcservo
25	HEAD_JOINT[2]	0.28	0.36	Rcservo

6.6 Sensors

6.6.1 A position of sensor loading

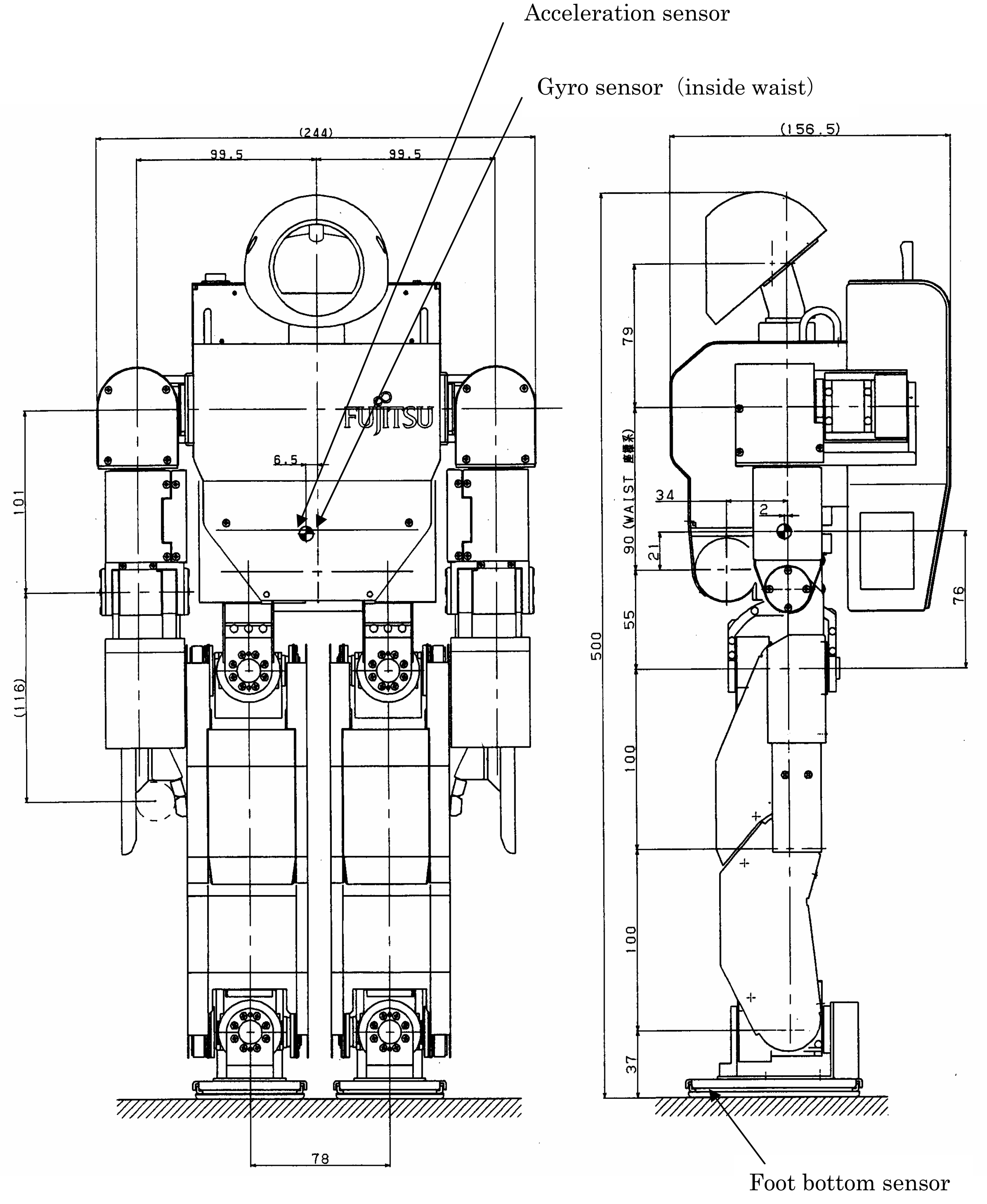
Sensors to load into Robot is

・Posture sensor 1 pc

・The foot bottom sensor 2 pcs

・USB Camera 2pcs （option）

These positions of loading show a posture sensor is combined by Gyro sensor and acceleration sensor in the Drw.6.6.1-1.



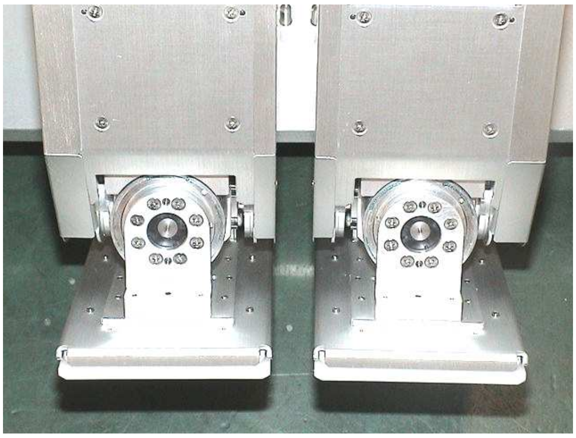
Drw.6.6.1−1 Sensor loading position

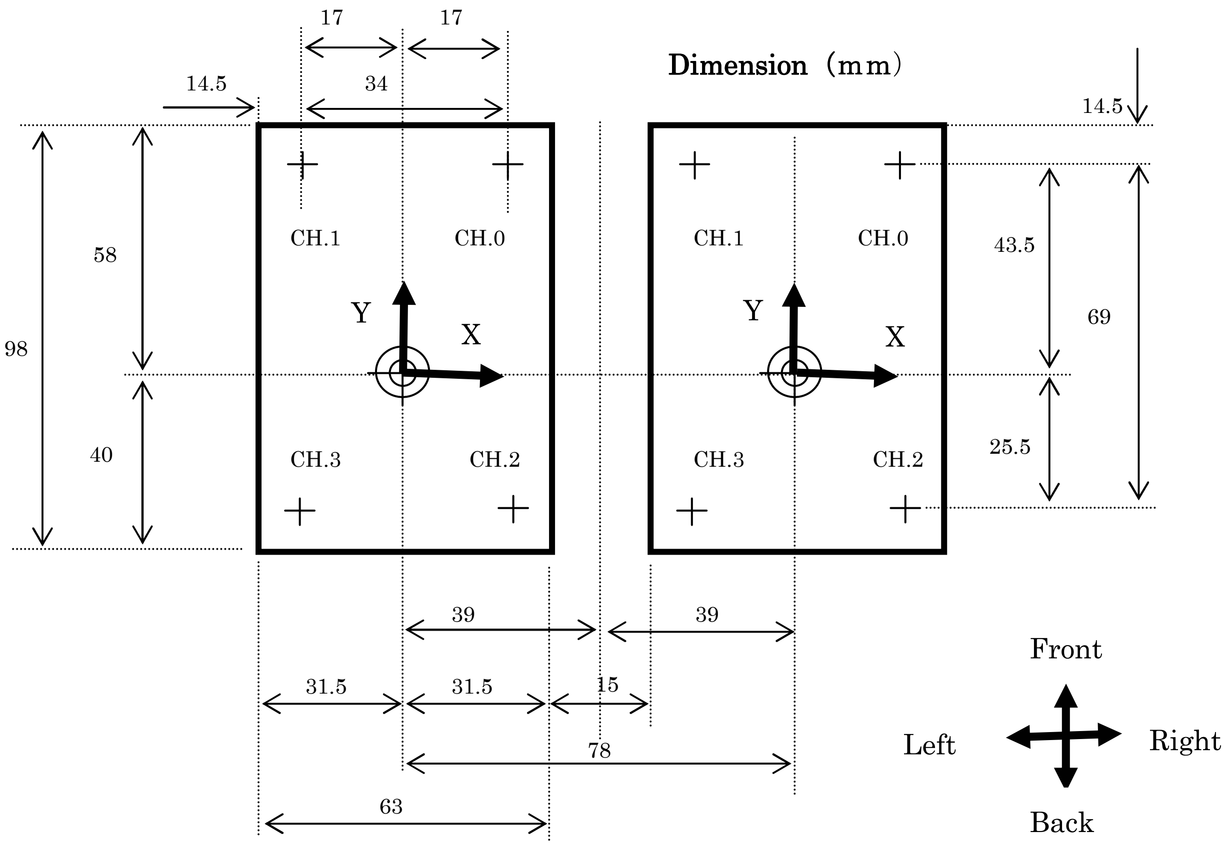
6.6.2 Posture sensor

Posture sensor is equipped inside of the upper half of the body of the robot, and composed of 3 axis angular velocity sensor (gyro) of and 3 axis acceleration sensor. to secure the precision of the posture sensor, Be sure to acquire the standard data of the acceleration and the angular velocity under the condition of standstill in the joint initialization jig adjusted horizontally and calculate the amount of physics in the amount of change from these data. The change-type to calculate physical data in the WAIST coordinate which is the standard coordinate of the robot from a result of ADC of each sensor is shown in the materials for the attached sheet. Warn that the character of the sensor is different in every individual. Especially, be careful of the ADC channel numerical order of the acceleration sensor with that polarity because they are not X, Y and Z of the order of the robot coordinate

6.6.3 Foot bottom sensor

The foot bottom sensor which detects the vertical anti-power (W) and Zero moment point (ZMP) which hangs on the foot when foot lands on the floor on both legs of the robot.A foot is equipped with four pressure sensors (FSR element) that resistance value changes corresponding to the pressure as detection element, and ZMP and W are calculated to use its resistance change. Because the character of the sensor is different in every individual of the robot, it is calculated by the independent compensation chart. Below is The form of the foot bottom sensor and a ZMP detection coordinate.





Leg angle position

Pressure sensor position
(CH.# is AD conversion channel)

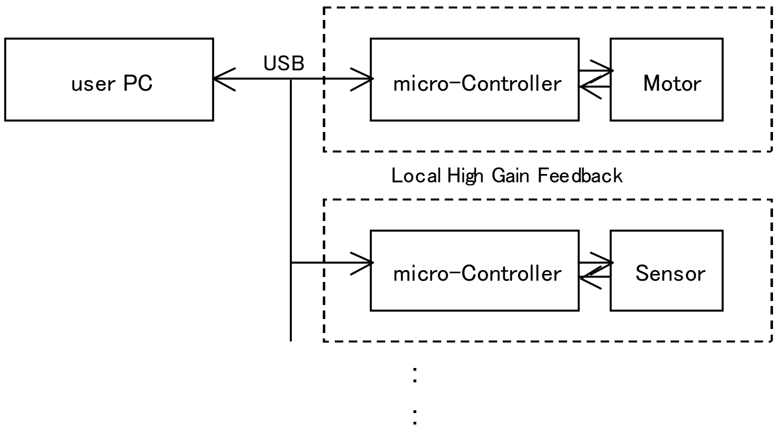
X, Y : ZMP defection coordinate

Foot bottom shape and Pressure sensor position

7. Program and Control

7.1 Local Control

As for this robot, a micro-controller is arranged to each joint, and Local High gain feedback control is done in every joint, and take the form that that movement target value is directed delicately from the motion command personal computer. Because of that, each micro-controller doesn't do an orbit calculation such as so-called trapezoid method. Control is done to meet that substitute command value promptly which a user command. <u>USB</u> which is the standard general-purpose interface of the personal computer is adopted for the communication between the personal computer and the micro-controller. <u>RT-Linux</u> is adopted as a real-time operation system in the motion command personal computer.



Drw 7.1 The control composition

There are that advantage and a restriction by having adopted USB in the communication form. Understand it about the character of USB when you handle this robot system.

Advantage:

- Communication speed is comparatively fast(12Mbps)
- Many devices can connect (up to127 units), communication system can be easily expanded.

Restriction:

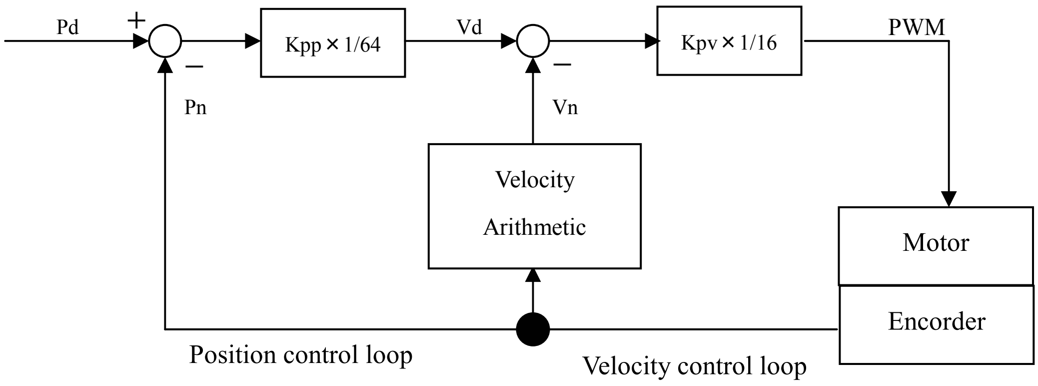
- There are a host, a general idea of host and function, and a host side controls all of the communication fundamentally. In other words, it can't communicate actively from the function side. As for this system, PC is equivalent to Host, Micro-controller is equivalent to function.
- A device for the divergence caused with HUB for the increase of the function is necessary.
- Communication method is restricted due to communicating in the frame unit of every 1mS so called SOF.

7.1.1 Motor control board & Sensor board

A robot is equipped with 22 motor control boards and 3 sensors boards which control each motor that is necessary for the posture control.

7.1.1.1 Motor Control Board

The block diagram of the firmware which equips it with a motor control board is shown in the Drw 7.1.1-1. When the speed control mode is taken, Vd to say in the following block diagram is decided to be direct directly. A speed is calculated by the number of changes in a pulse of the unit 1mS from the one for the difference of the encoder.



Drw7.1.1-1. Motor Control board block drawing

7.1.1.2 Sensor Management Board

As for the sensor management board,6 channels of AD are being output when a running average for 8 times is taken respectively for 1mS, It begins to read the condition of the IO port which it prepares for by the option, and it is condition within 1mS just before a demand for data. (But, an IO port is connected nowhere at present.)

7.1.1.3 RC Motor board

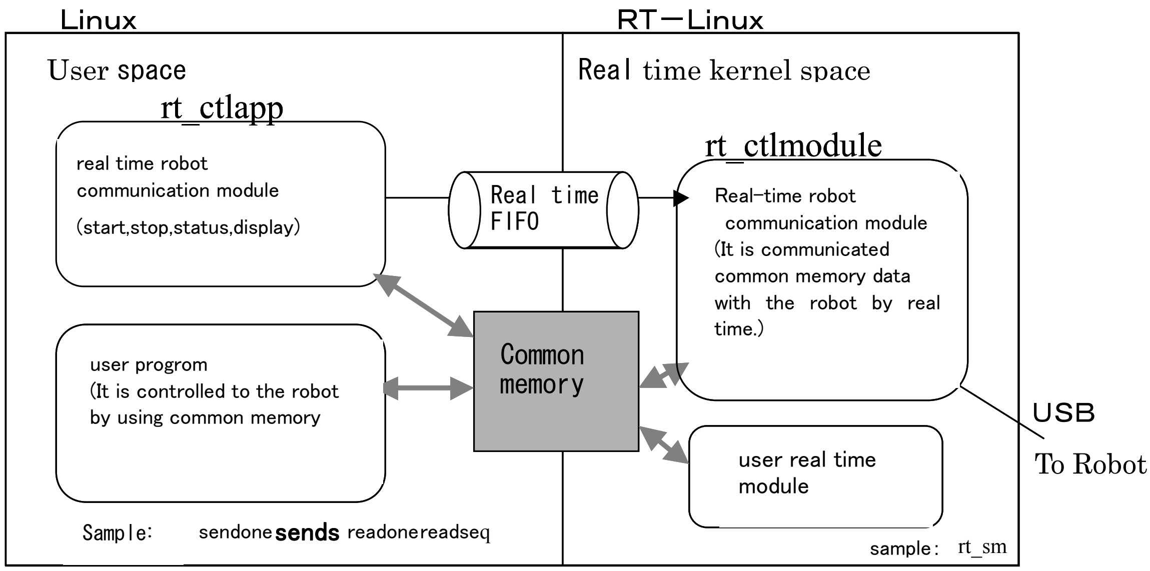
RC Motor board is connected to RC Motors of head pan, head tilt, and both hands.
When RC motors are controled, it gives motor code and moving angle.

7.1.2 Software on Motion Command PC

Explains about the construction of the software of this robot.

7.1.2.1 Whole composition

Software on the motion command personal computer of this robot is greatly divided into the real-time robot communication module carried out with real time kernel space and the program which loading ・ indicates data on the robot carried out in the user space. A data loading, indication program and a real-time robot communication module transmit and receive data and a command by using the common memory. Common memory can writing and reading from every process. Data loading program is written data and a command in the common memory to acquire from the file and the standard input. a real-time robot communication module that content of written in the common memory is transmitted and writes the result received from the robot in the common memory. A data display program indicates the result written in the common memory.
It becomes possible that a robot is made to operate by a user's program if user make program that is reading/writing of the common memory



Drw.7.1.2-1Software composition on motion command PC

7.1.2.2 Directory composition

Software of this robot is installed in /usr/local/hoap2

/bin	:Execute form file
/data	:The sequence data file of the robot
/include	:Header file
/modules	:Real Time Module
/src	:Source file
/sm_access	:User Program to access in common memory
/examples	:Sample file
/rt_sm	:Sample of real time control mode

Explain in detail for each programs at 7.3

The sources of all programs is open except real-time robot communication module and real-time robot communication module.
Refer to it in case of individual user program preparation.

7.2 Control method of Robot

It is required to control Robot to use prepared command, and under its restriction, must do the communication of the data It prepares for the three kinds of operation modes, and it is possible that the command which shows it in each of the Drw. 7.2-1 is used as for this robot system. Understand it about the name of the mode and the command and that outline in this chapter. Explain below how to set it up referring to the structure (MemMan.h and Drw.7.2-2) of the sample which is necessary for the access actual common memory

Direct Control Mode --- General command(Header = 3)

Sequence mode
- General command(Header = 3)
- Fixed command(Header = 2)

Real time control mode --- Fixed command (Header = 2)

※Header =1:Reserved

Drw.7.2-1 Relation Operation mode and Kinds of command

7.2.1 Kinds of command

Explain here for outline of commands which compose operation mode.

7.2.1.1 General Command

It is the mode to support all commands to the device. There is a command that shows it in S 2-1 in each command with a variable length corresponding. In other words, it takes at least 40mS to send 20 devices by using the general command. to the device. But, when this command is used, it takes the minimum 2mS per 1 command. In other words, it take at least 40mS to send 20 devices by using the general command.

7.2.1.2 Fixed command

It is the communicate mode of the fixed length specified for usage. Data communications can be done to all the devices (as for this basic system, 20 motor boards and 3 sensor boards) within 1mS. This velocity is applied in terms of communication when shipping.

```
typedef struct {
    USHORT   Step;    // Interval until next transmission
    UCHAR    Header;
    union {
        struct {     // Header = 2(Fixed command)

            USHORT   MtrS[MotorMax];       ←Motor command value
            UCHAR    SnsS[SensorMax];      ←Sensor command
            USHORT   MtrRp[MotorMax];      ←Position correspondence
            USHORT   MtrRv[MotorMax];      ←Velocity correspondence
            USHORT   MtrRi[MotorMax];
            union {
                UCHAR        SnsRB[SensorMax][SnsResMax];  ←Sensor return value
                USHORT       SnsRW[SensorMax][SnsResMax/2];  (read in the size of the return value)
                ULONG        SnsRL[SensorMax][SnsResMax/4];
                LONGLONG     SnsRLL[SensorMax][SnsResMax/8];
            };
        }Ctl;
        struct {     // Header = 3 (General command)
            UCHAR    DevID;
            UCHAR    CmdAsc;
            union {
                UCHAR    ArgB[ArgMax];      ←The argument of the general command
                USHORT   ArgW[ArgMax/2];     (read in the size of the return value)
                ULONG    ArgL[ArgMax/4];
            };
            union {
                UCHAR        ResB[SnsResMax];  ←Correspondence of general command
                USHORT       ResW[SnsResMax/2];    (read in the size of the return value)
                ULONG        ResL[SnsResMax/4];
                LONGLONG     ResLL[SnsResMax/8];
            };
        }Cmd;
    };
    USHORT   ResTime;  ←Correspondence time
    UCHAR    MtrMode[MotorMax];
    UCHAR    StepOver;
}DUNIT;

typedef struct {
        :
    DEVICE   Device[nDevice];     ←The condition of the connection device. It defines separately
                                  in Usbhc.h.
        :
}VARIF;

typedef struct {
    BOOL     UsbResetStart;       ←Reset USB(=1)
        :
    BOOL     InterruptSend;       ←Start general command(=1)
        :
    BOOL     ResRep;              ←Complete real-time control data communication
    BOOL     ResInt;              ←Complete general command data communication
        :
    int      Mode;                ←Mode settlement(1:real-time control,2:general command,
                                  combine trigger)
        :
    char     MtrMode[MotorMax];   ←Control mode of motor
        :
}VARIF;
```

Drw.7.2-2.Explantion of MemMan.h

The complement explanation of "the Drw.7.2-2 MemMan.h". It accesses a memory with the structure to show here, and command transmission and answer data are acquired by the sample program.

Fixed Command

(1)Motor command value MrtS[]

MrtS[MotorMax]: It appears that encoder counter value in case of position command mode.

↑

Numbers of installation numbers of motor control board: when ship out :MotorMax =21

The numerical value of [] copes with order (in bottom of the order from the left column) of the table 6.1-1 with 0 - 20.

Example) MrtS[0]: RLEG_JOINT[1] The right thigh joint twisted.
MrtS[1]: RLEG_JOINT[2] The right and left of the right thigh joint
. . . .
MrtS[6]: RARM_JOINT[1] Front & rear the right shoulder
. . . .
MrtS[10]: LLEG_JOINT[1] The left thigh joint twisted
. . . .
MrtS[16]: LARM_JOINT[1] Front & rear the right shoulder
. . . .
MrtS[19]: LARM_JOINT[4] Left elbow
MrtS[20]:BODY_JOINT[1] Waist

(2)Sensor command SnsS[]

SnsS[SensorMax]: Sensor command of (S.2.1 command list)

↑

It has the loading number # kind of the sensor management board. The sample file name which the explanation of each use and that memory are being used for is shown in the following.: When shipping :SensorMax =3

For example, when R command is put, it answers the AD data IO port data of the sensor management board at one time. These answer data can lead with SnsRB[][] to mention later, SnsRW[][], SnsRL[][] and SnsRLL[][].

(Caution)S command with the argument can't be used in case of a fixed form command.

Numerical value inside [ ] copes with the following sensor with 0 - 2.

SnsS[0]: The right foot bottom sensor
SnsS[1]: The left foot bottom sensor
SnsS[2]: The posture sensor

(3) Correspondence value of motor MtrRp[], MtrRv[ ], MtrRi[]

MtrRp[MotorMax]:A result of a position answer for the latest command value

MtrRv[MotorMax]:A result of a velocity answer for the latest command value

MtrRi[MotorMax]:NA(No meaning data)

The numerical value of [ ] copes with the same joint as the case of the motor command value of (1).

(4) Sensor return value SnsRB[][], SnsRW[][], SnsRL[][], SnsRLL[][]

This is union which it prepares for to read these correspondence data by the various data heads because it is written in the memory that answer data in form which varies in the command are the same.

For example, when a R command was transmitted to the sensor circuit board, it understands from "S .2.1 command list ", as for the correspondence data

The AD change data of CH0： ushort Data
The AD change data of CH1: ushort Data
The AD change data of CH2: ushort Data
The AD change data of CH3: ushort Data
The AD change data of CH4: ushort Data
The AD change data of CH5： ushort Data
The data change of IO port: uchar Data

Because the head address of SnsRB[][], SnsRW[][], SnsRL[][], SnsRLL[][] are all same,
In case of right foot bottom sensor,

The AD change data of Ch0: SnsRW [0] [0] defined in the ushort type
The AD change data of Ch1: SnsRW [0] [1] defined in the ushort type
The AD change data of Ch2: SnsRW [0] [2] defined in the ushort type
The AD change data of Ch3: SnsRW [0] [3] defined in the ushort type
The AD change data of Ch4: SnsRW [0] [4] defined in the ushort type
The AD change data of Ch5: SnsRW [0] [5] defined in the ushort type
The data of IO port: SnsRB[0][12]defined in the uchar type

The first 0 becomes 1 in case of the left foot bottom sensor.

In case of Posture sensor

The AD change data of Ch0: SnsRW [2] [0] defined in the ushort type
The AD change data of Ch1: SnsRW [2] [1] defined in the ushort type
The AD change data of Ch2: SnsRW [2] [2] defined in the ushort type
The AD change data of Ch3: SnsRW [2] [3] defined in the ushort type
The AD change data of Ch4: SnsRW [2] [4] defined in the ushort type
The AD change data of Ch5: SnsRW [2] [5] defined in the ushort type
The data of IO port: SnsRB[02[12]defined in the uchar type

As for the interpretation of each data, refer to the sensor character of the attached sheet and " 6.6.2 posture sensor","6.6.3 bottom sensor" as descriptions. CH.4 of the right foot bottom sensor acquires the voltage data of the battery so that it indicate"S1.4 the way of replacing battery". Remove a battery promptly when it becomes less than 550LSB. And, the least significant bit of the IO port of the posture sensor is being used for turning on LED of the breast of the robot.

General command

(1) No. of transmitted device DevID

It is the number of the device which becomes a place of transmission of a general command to transmit only to the specified device. Defined as below.

1: RLEG_JOINT[1] Motor control board of joint(capture 6.1-1reference)
2: RLEG_JOINT[2] Motor control board of joint
3: RLEG_JOINT[3] Motor control board of joint
4: RLEG_JOINT[4]Motor control board of joint
5: RLEG_JOINT[5] Motor control board of joint
6: RLEG_JOINT[6] Motor control board of joint
7: RARM_JOINT[1] Motor control board of joint
8: RARM_JOINT[2] Motor control board of joint
9: RARM_JOINT[3] Motor control board of joint
10: RARM_JOINT[4] Motor control board of joint
11: LLEG_JOINT[1] Motor control board of joint
12: LLEG_JOINT[2] Motor control board of joint
13: LLEG_JOINT[3] Motor control board of joint
14: LLEG_JOINT[4] Motor control board of joint
15: LLEG_JOINT[5] Motor control board of joint
16: LLEG_JOINT[6] Motor control board of joint
17: LLARM_JOINT[1] Motor control board of joint
18: LLARM_JOINT[2] Motor control board of joint
19: LLARM_JOINT[3] Motor control board of joint
20: LLARM_JOINT[4] Motor control board of joint
21: BODY_JOINT[1] Motor control board of joint
22: The right foot bottom treatment board
23: The left foot bottom treatment board
24: Posture sensor treatment board
25: RC motor control board (HEAD_JOINT[1], HEAD_JOINT[2], RARM_JOINT[5], LARM_JOINT[5])

（2）Command　 CmdAsc

They are command to transmit each device. Refer to「S.2.1Command list table」

（3）Command reference　ArgB[], ArgW[], ArgL[]

It is reference data to transmit to above command. Refer to「S.2.1　Command List table」 It is provided different arrangement of data length union so that the data format of the various commands may respond as well as the sensor return value.

（4）Response of General Command　ResB[], ResW[], ResL[]

It is response data from devices that transmitted. This is also declared the arrangement of the different data length with union to be able to comply with the various data formats in the same way.

（5）Response time　 ResTime

It is response time result from actual devices.

（6）Motor Control Mode　 MtrMode[]

They are the data　which show the control mode of the motor set up.
Refer to [S2.1 Command list table]

（7）Step Over　　StepOver

Not used

Explanation
VARIF

VARIF Structure

```
typedef struct {                         // Driver  User
        BOOL    UpStreamEnable;          // W       R       // It can't use.
        BOOL    UsbResetStart;           // R(W)    W       // USB bus reset transmit start
        int     ConfiguredHB;            // W       R       // setups complete HUB number
        int     ConfiguredDD;            // W       R       // setups complete device No.
        int     ConfiguredIDX;           // W       R       // setup complete index
        BOOL    ConfiguredSS;            // W       R       // It can't use
        BOOL    ConfiguredCD;            // W       R       // It can't use
        BOOL    DAction;                 // W       R       // It can't use
        BOOL    SendStart;               // W       R       // It can't use
        BOOL    SendOK;                  // W       R       // It can't use
        BOOL    InterruptSend;           // R(W)    W       // The transfer start command of the general
                                                                   command
        BOOL    Received;                // W       R(W)    // Response receiving completion
        BOOL    ResRep;                  // W       R(W)    //  〃  for fixed form command
        BOOL    ResInt;                  // W       R(W)    //  〃  for general command
        USBERR  UsbErrMon;               // W       R(W)    // Transfer status
        USBERR  UsbErrUs;                // W       R(W)    // It can't use
        int     Mode;                    // RW      W       // Transfer mode command/monitor
        int     SendLength;// W          R        // Command size（Referenced）
        int     ResLength;               // W       R       // Response size（Referenced）
        int     cRes;                    // W       R       // Response receiving counter
        int     ReceiveLength;           // W       R(W)    // Response size（Actual）
        int     ResTime;                 // W       R       // Response time
        int     ResTimeMin;              // W       RW      // Minimum response time
        int     ResTimeMax;              // W       RW      // Max. response time
        double  ResTimeAvr;              // W       RW      // It can't use
        char    MtrMode[MotorMax];       // W       R       // Current each motor mode
        BOOL    UpStreamSendStart;       // R(W)    W       // It can't use
        int     DBG;                                        // It can't use
}VARIF;
```

Explains to the one by the function group related to the command transfer about the useful thing.

(1) USB Reset

UsbResetStart Read/Write

It specially explains to the one by the function group about the command transfer for the useful thing. When a user sets TRUE, a USB bus reset sending out starts. It becomes FALSE at once after a driver completes acceptance. Emulate management of all the USB devices which contain HUB is started.

(2) Related to vice emulation

ConfiguredHB ReadOnly

Store HUB number that finished emulation.

ConfiguredDD ReadOnly

Sore device number that finish emulation.

ConfiguredIDX ReadOnly

Shows index of HUB or Device that completed emulation imminent.

(a)General command sending

InterruptSend Read/Write

When the general command transmitted by transfer start command (It is effective during the idol or only the real-time control mode.), INTERRUPT direct control mode shouldn't set up. Instead of it, set up InterruptSend = TRUE。A driver becomes InterruptSend = FALSE when this flag is received. At this time, INTERRUPT is set up in VarIF.Mode inside the driver. This is because it could handle a general command as an interruption transfer in the real-time control mode.

(b)Transfer complete flag

A user must reset each transfer complete flag in FALSE before the command transmission. These flags become TRUE when a transfer is completed. Though "transfer completion" to say here is "command transmission - a response receipt", it becomes TRUE at the moment when command transmission was completed at the case of the general command in case of the one without a response.

in case of the real-time control mode, renew command value, after Received or ResRep was made FALSET,when it confirmed that that flag became TRUE, renewed command value received by target, or response value becomes to renewal. Usually, it is possible to confirm transfer completion by using Received, but you must confirm each transfer completion by using ResRep / ResInt when an interruption transfer in general command form is specially done in the real-time control mode

Received **Read/Write**

Transfer completion (common)

ResRep **Read/Write**

Transfer completion (for fixed form command)

ResInt **Read/Write**

Transfer completion (for general command)

(c) ＵＳＢ transfer status

UsbErrMon **Read/Write**

It is constructed by the structure USBERR. cBitStuffErr - cStalled is counter which counts information such as abnormal in USB protocol that it is formed by USB Host Controller . cResTimeOut and TDEndTO is the information that it is formed by driver . A user can reset counter by writing 0.

```
typedef struct {                // Driver User
        int     cBitStuffErr;   // W    R(W)
        int     cCRCTOErr;      // W    R(W)
        int     cNAKReceiveOut; // W    R(W)
        int     cNAKReceiveIn;  // W    R(W)
        int     cBBLDetect;     // W    R(W)
        int     cDBuffErr;      // W    R(W)
        int     cStalled;       // W    R(W)
        int     cResTimeOut;    // W    R(W)    // Response receiver time out.
        BOOL    TDEndTO;        // W    R(W)    // bucket sending time out.
}USBERR;
```

(d) Transfer mode command/monitor

Mode **Read/Write**

Set up value of transfer mode fixed number definition ModeConstants. A driver does the movement to be equivalent to each mode. But, it is only IDLE, REPITITION and SEQUENTIAL that a user can set. Refer to InterruptSend in case of general command transfer.

```
enum ModeConstants{
    IDLE,           // Idle
    REPITITION,     // Real time comtrol
    SEQUENTIAL,     // Sequence
    INTERRUPT,      // Direct control (user cannot set up)
    UPSTREAM        // It can't use
};
```

(e)Transfer Data size

SendLength **ReadOnly**

Transmit command size (Referenced)

The command size defined inside the driver in advance for every transmitting command is stored.

ResLength **ReadOnly**

Response size (Referenced)

The response size defined inside the drive in advance for every transmitting command is stored.

CRes **ReadOnly**

It is Response receiving counter to use inside of driver.

Become receipt completion at cRes == ResLength

ReceiveLength **Read/Write**

Response size (Actual)

It is renewed when a response receipt is completed in the size that it was actually received.

(f)Response time

ResTime **ReadOnly**

Response time

When a transfer is started, initialization (0) is done by a driver. After a command is transmitted, transfer time to the response receipt is set in the frame (1mS) unit. On the other hand, response time ResTime exists in member of the structure DUNIT as well. The value of each ResTime of VARIF and DUNIT is shown by each condition.

A pointer is made pDunit to the DUNIT structure of the object.

1. In case receive normal response,
   VarIF.ResTime = n
   pDunit->ResTime = n
   n : Flame No. used actually.([ms])
2. In case of transmitting general command with no response.
   VarIF.ResTime = 0
   pDunit->ResTime = 1
3. When an invalid device ID was specified with a general command.
   VarIF.ResTime = No renewal
   pDunit->ResTime = -1
   ※ A command isn't transmitted and finished. Be careful in the direct control mode because a transfer complete flag isn't set, either. But, a step progresses in case of the sequence mode.
   ※ When it was abnormal finished by the host controller.

VarIF.ResTime = -1

pDunit->ResTime = -1

The contents of the abnormal can be confirmed by using counter of UsbErrMon.

4. When a time-out (1sec) was done by the NAK receiving continuation and so on.

,VarIF.ResTime = 0

pDunit->ResTime = not renewal

The occurrence of the time-out can be confirmed by using counter of UsbErrMon.

The minimum value of ResTime and maximum value are stored in ResTimeMin/ResTimeMax.。A user can initialize it by writing 0 respectively.

ResTimeMin **Read/Write**

Minimum response time

ResTimeMax **Read/Write**

Maximum response time

(g) Motor control Mode

MtrMode[MotorMax] **ReadOnly**

Each present motor control mode is stored, and [device ID - 1] of the motor can be referred to as an index. The number of the motor control mode change command transmitted at the end is stored.

('A'/ 'B'/ 'C' = Position / Velocity / <reservation>)

Refer to this for this information except for the sequence mode though it is the same as the thing of the structure DUNT inside usually. MtrMode inside DUNIT is effective when the motor control mode is confirmed in each step which has been carried out in case of the sequence mode.

The state of connection condition when ship out of a sensor treatment circuit board as following.

table Connection condition of sensor treatment board when ship out

Type	Right foot bottom Sensor treatment board	Left foot bottom Sensor treatment board	Posture sensor Treatment board	Reference
Fixed command arrangement	0	1	2	□of SnsS□ value
General command DevID	22	23	24	
AD0	FSR sensor ch0	FSR sensor ch0	Acceleration (z)	Robot coordinate xyz
AD1	FSR sensor ch1	FSR sensor ch1	Acceleration (y)	Robot coordinate xyz
AD2	FSR sensor ch2	FSR sensor ch2	Acceleration (x)	Robot coordinate xyz
AD3	FSR sensor ch3	FSR sensor ch3	Gyro (x)	Robot coordinate xyz
AD4	Battery voltage monitor	Empty	Gyro (y)	Robot coordinate xyz
AD5	Empty	Empty	Gyro (z)	Robot coordinate xyz
IO7	Empty	Empty	Empty	
IO6	Empty	Empty	Empty	
IO5	Reserved	Reserved	Reserved	
IO4	Empty	Empty	Empty	
IO3	Empty	Empty	Empty	
IO2	Empty	Empty	Empty	
IO1	Empty	Empty	Empty	
IO0	Empty	Empty	The output for LED of the breast	

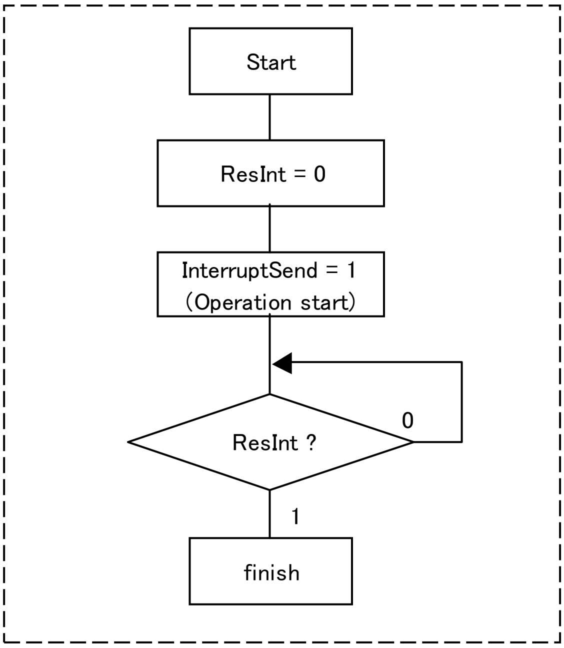
7.2.2 Kinds of operation mode

The outline is explained about the operation mode corresponding to the user's use situation. Each operation mode starts a movement by establishing the value of Mode. The correspondence of the operation mode and that value is as following.

Operation mode	Mode value	Motion start Trigger
Direct control mode	0(= IDLE)	InterruptSend = 1 set
Sequence mode	1	Mode = 1 set
Real time control mode	2	Mode = 2 set

7.2.2.1Direct control mode

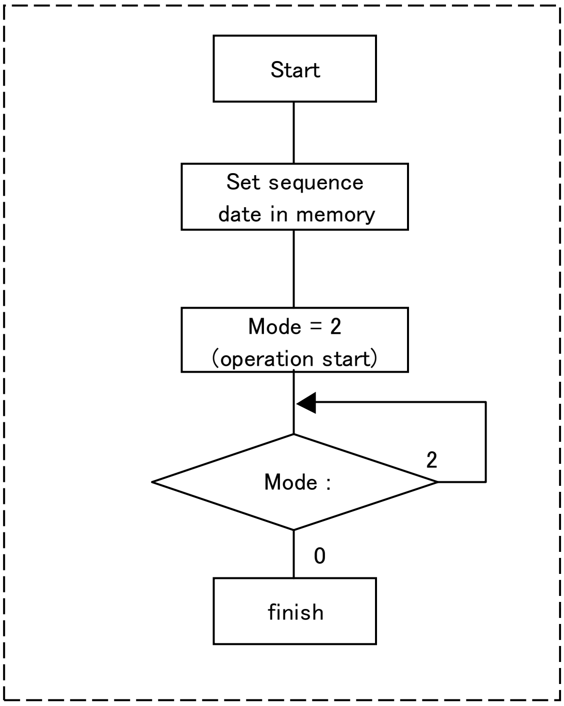
It is the mode to use when general operation is done in the device unit. A specified command is carried out in this mode on every 1 line. Only a general command can be directed to the command. Surely, the next data are set after it confirms that a general command data communications complete flag (ResInt) stands. You must stand an interrupt flag (InterruptSend) more to transmit a message actually. A flow chart in this mode is shown in the Drw 7.2-3.



Drw. 7.2-3. Direct control mode

7.2.2.2 Sequence mode

It uses when it wants to do operation continuously without manual input making a command process with macrocosm in the direct control mode . A command is enumerated in the file, and a command is carried out in the time interval specified as turn by setting it on the memory. It is possible that a fixed form command is made to be mixed with the general command in this mode. But, the processing of 1 line takes the time of at least 2mS at the general command and at least 1mS at the fixed command. It is required to stand a sequence mode start flag to carry out the mode more.

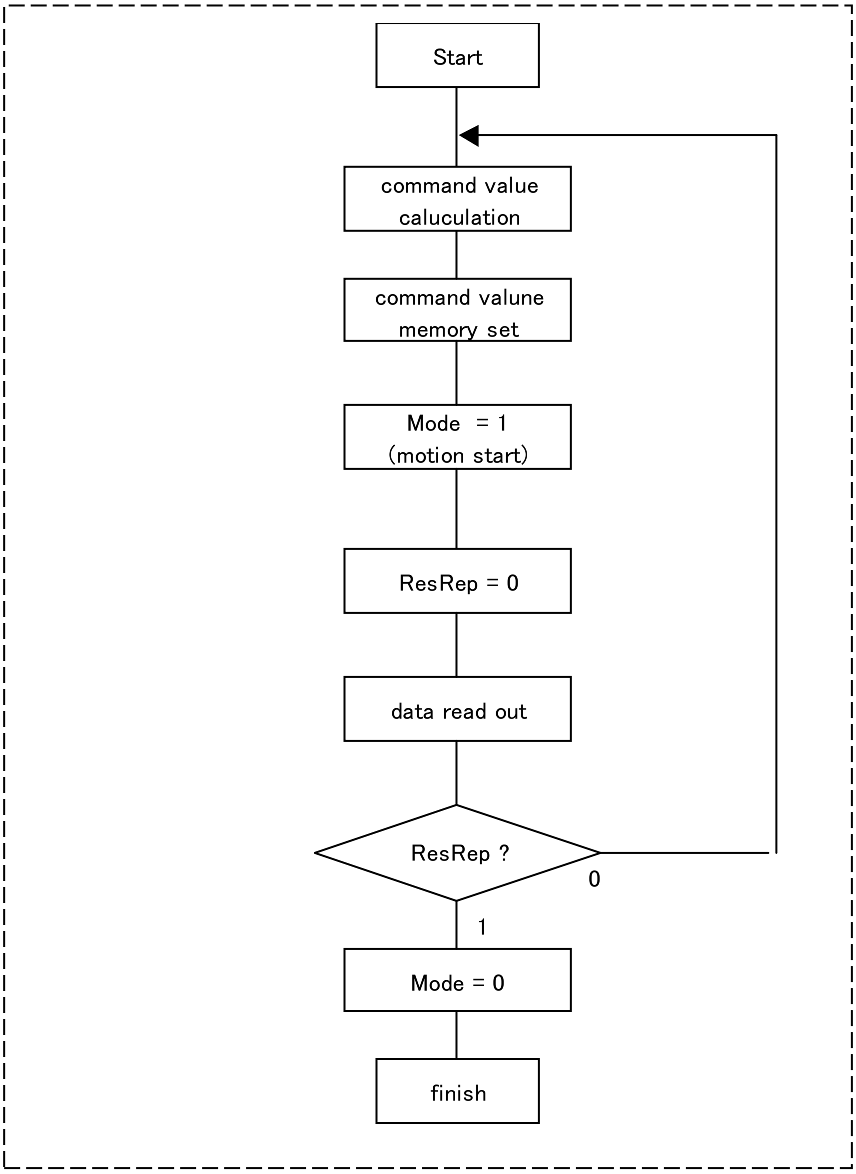


Drw. 7.2-4 Sequence mode

7.2.2.3 Real time control mode

It is the mode to use for a user when to do real-time operation such as closed loop control. A user must control with seeing a fixed form command data communications complete flag exactly the matter whether there was an effective answer or Command value can be written in .

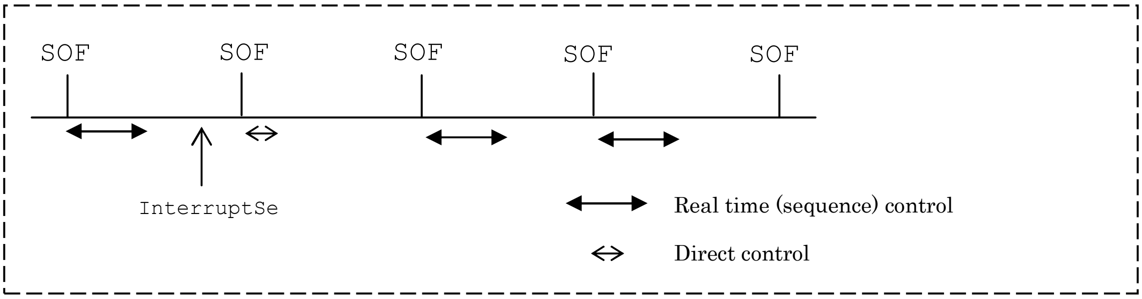
It can be controlled within 1mS if the next command value is written within 100us after command value is written in and a fixed form command data communications complete flag stands. And, you must stand a real-time control start flag to make this mode effective.



Drw. 7.2-5 Real time control mode flow chart

7.2.2.4 Mixture of Operation mode

While continuance in the real-time control mode or the sequence mode works, there is time which it wants to operate by the case to the specific device. In this case, interruption operation can be done by the direct control mode. The direct mode isn't started movement by Actually, data are transmitted with the next frame that stands an InterruptSend flag, and it is returned from the next frame again for the movement of the usual control mode. the settlement of the value of Mode, but it is to start transmission by standing a flag on InterruptSend.



Drw.7.2-6.Mixture of Operation mode

7.3 Explanation of sample soft ware

Explain software for robot on motion command PC

rt_ctlapp	
explanation	It communicates with the real-time robot communication module rt_ctlmodule.o, and it is the program which gets the practice of the start/stop of the real-time robot communication module and information on the robot.
How to use	。A RT-linux module and a robot control real-time module rt_ctlmodule.o are started by entering command as rtlinux start rt_ctlmodule.o and a command. Next, when rt_ctlapp is carried out, it becomes the prompt of *.*.*$.(*is number) First * is in the number of USB hubs, the second * is in the number of USB controller, their * indicate USB controller in the recognition. The following command can be input. r：USB reset v：Common memory status indication h：Help d：i：Direct control mode monitor r：Real time control mode monitor 2：Setting to real time control mode 3：Setting to direct control mode
Caution	It cannot carry out if rtlinux module is not operated. rCarry out by root.
Binary	/usr/local/hoap2/bin/rt_ctlapp
Source	Non

Sendone	
Explanation	It is the program which transmits a general command to the robot. A command is written in the common memory. A general command is carried out by transmitting the contents that a real-time robot communication module was written in the common memory to the robot itself.
How to use	Carry out from command line、 Shows prompt as command: A general command is written in the common memory when a command is written and a return. When the practice of the command is completed, return value is written down in the standard output. It is indicated continuously as command : a command can be input in succession. if practice isn't completed like a robot isn't connected , return to command: prompt by pushing any key. Finish if input with quit. Input command is <sec>,<header>,<devID>,<command>,<param1>. <sec>:Whatever is input, it doesn't care in the general command mode because it is meaningless. <header>:Input "3" which shows a general command <devID>:It is specified which USB controller of the robot a command is sent to. <command>:Input to refer to general command list <param1>..:It is a parameter to different from <command>。Input to refer to general command list. For example command: 2,3,4,J is to send general command as J in device 4.
Caution	It can't be carried out if a rtlinux module doesn't start it. Carry out by root
Binary	/usr/local/hoap2/bin/sendone
Source	/usr/local/hoap2/src/sm_access/sendone.c, sm_access.c

Sendseq	
Explanation	It is a program to send sequence data for robot. Write sequence data in common memory. A sequence motion is carried out by transmitting the contents that a real-time robot communication module was written in the common memory to the robot itself.
How to use	From command line sendseq < <filename> as、If input file name by re-direct, that file is loaded, and written in as the sequence data, and the practice of the sequence motion is indicated. If input file name by re-direct, that file is loaded, and written in as the sequence data, and the practice of the sequence motion is indicated.
Caution	It can't be carried out if a rtlinux module doesn't start it. Carry out by root
Binary	/usr/local/hoap2/bin/sendseq
Source	/usr/local/hoap2/src/sm_access/sendseq.c, sm_access.c

Readone	
Explanation	It is the program which indicates all the main things of the contents being written in the common memory at present. Contents of command transmission and that result can be confirmed.
How to use	From command line Input as readone it is indicated content of data of common memory at present.
Caution	It can't be carried out if a rtlinux module doesn't start it. Carry out by root
Binary	/usr/local/hoap2/bin/readone
Source	/usr/local/hoap2/src/sm_access/readone.c, sm_access.c

Readseq	
Explanation	While a sequence motion is carried out, it is a program that reads a result to the place where a movement was finished one after another from the common memory and to indicate. Present contents of command transmission and that result can be confirmed. It is the program which indicates the contents being written in the common memory.
How to use	From command list Input as readseq Contents of transmission to the sequence movement being carried out at present and a result are indicated. On and after, it is indicated every time the next movement during the sequence movement is finished.　Finish if input anything by keyboard.
Caution	It can't be carried out if a rtlinux module doesn't start it. Carry out by root
Binary	/usr/local/hoap2/bin/readseq
Source	/usr/local/hoap2/src/sm_access/readseq.c, sm_access.c

rt_sm	
Explanation	It is the sample program　which writes motion data in the common memory which uses a real-time robot communication module from the real-time module.
How to use	From command line Input as rtlinux start rt_ctlmodule.o rt_sm_module.o Start sample real time module together with real-time robot communication module. Start rt_sm_app 、sample real time module start, and data is written in common memory. rt_sm_app stops a sample real-time module, and finished when motor encoder value is received from the sample real-time module 100 times. The contents being written in the common memory can be confirmed by the readone　program.
Caution	It can't be carried out if a rtlinux module doesn't start it. 　Carry out by root. A robot doesn't work with this source because it is a sample to write in the common memory from the real-time module.
Binary	/usr/local/hoap2/examples/rt_sm/rt_sm_app, rt_sm_module.o
Source	/usr/local/hoap2/examples/rt_sm/rt_sm_app.c

Setpos	
Explanation	Current value is made to correspond to the target command value of the robot
How to use	Input as" setpos" from command line
Caution	It can't be carried out if a rtlinux module doesn't start it. Carry out by root.
Binary	/usr/local/hoap2/bin/setpos
Source	/usr/local/hoap2/src/sm_access/setpos.c

Interpol	
Explanation	It is the program which makes the sequence data file that interpolate the space to the initial posture of the sequence data file which will be read next from the present posture of the robot.
How to use	Interpolate sequence data file name is " outputfile"、sequence file name which will carry out next is "nextfile", then, input " interpol outputfile < nextfile" from command line Make sequence data file which interpolates with two seconds to the current position specified with a head of next file from the current position of the motor of the robot. outputfile change the file of the name of interpol.csv when it is omitted.
Caution	It can't be carried out if a rtlinux module doesn't start it. Carry out by root.
Binary	/usr/local/hoap2/bin/Interpol
Source	/usr/local/hoap2/src/sm_access/interpol.c, sm_access.c

８．Contact address

Customer service

Fujitsu Automation Co., Ltd. Kawasaki sales office
Sales Division

In Charge： Yoshino,　Kobayashi

Phone ： 81-44-754-3805

E-mail ： hoap@fja.fujitsu.com
(yoshino@fja.fujitsu.com , masashi@fja.fujitsu.com)

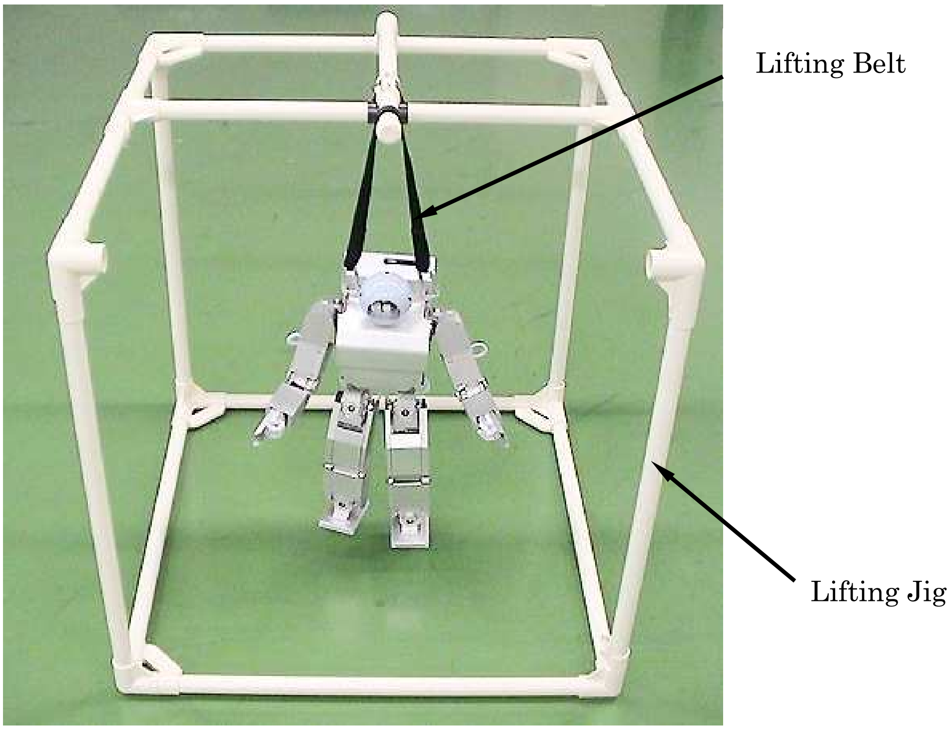
S. 1 Mechanical

S.1.1 Lifting jig

Lifting jig is provided to lift robot temporary as a accessory . The purpose of jig is : When a robot is kept without giving a load to the joint under the condition that it is hung in the air、or when a joint is controlled tentatively, or when It uses for the case that the output of the sensor is confirmed, the exchange work of the part, and so on.

It passes into picked up through attached lifting belt in the top (the back of the head) of the back of the robot, and it is hung the opposite side at the tip of the pipe in the center of lifting jig, and a lifting belt.

(＊)Be careful not to fall the robot to fall by the mistake when fixed the chain and so on.



Drw s1.1-1 Lifting jig

S1.2 The method for fixed and remove of joint initial setting jig

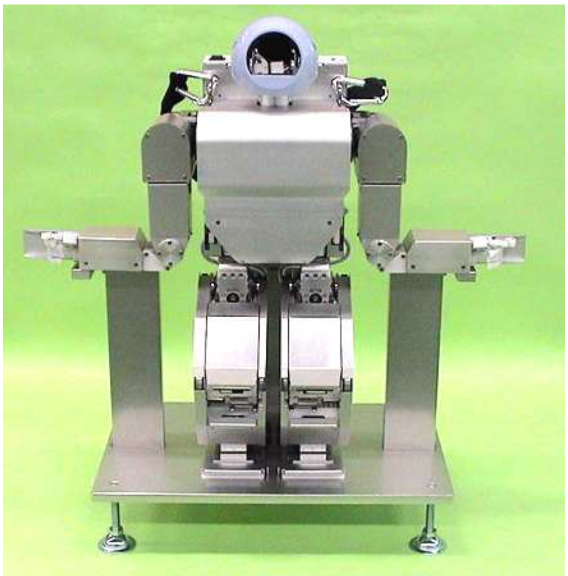
This Item is to explanation for method for fixed and remove of joint initial setting jig (Drw S1.2-1)

Set up the joint initial setting in the right position, If it does not do so, the robot may do unusual operation.

Drw. S1.2-1 Joint initial setting jig

S1.2.1 Fix

The purpose of this work fixes the corner of each joint of the robot on the initial posture, and pre-sets encorder counter of the joint in that posture. After job, become like drawing S1.2-2.



Drw.1.2-2 Initial posture of robot to be standard when fixed on joint initial setting jig.

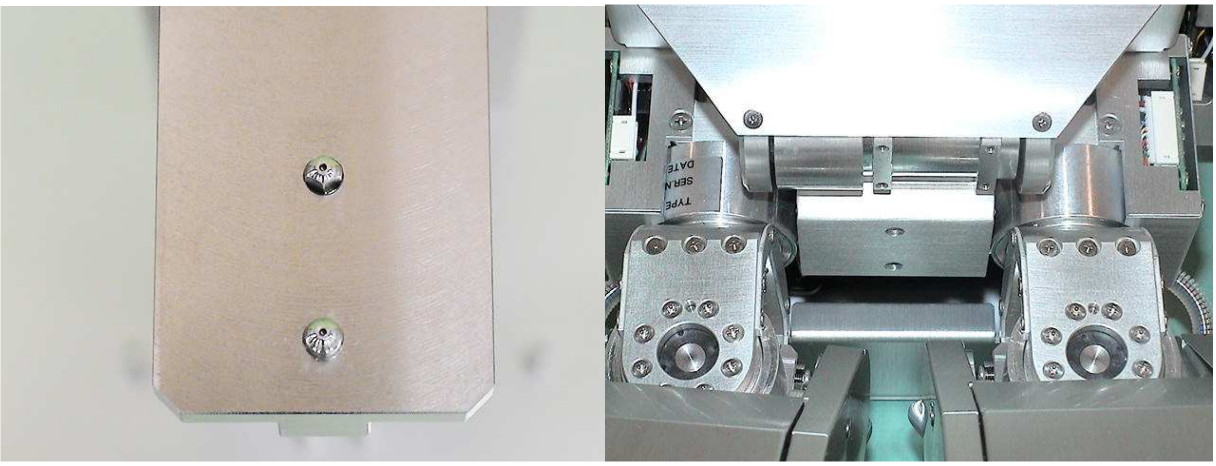
S1.2.2 The process of the fixation

A fixation is done with the following process after it confirms that the servo of the joint is off.

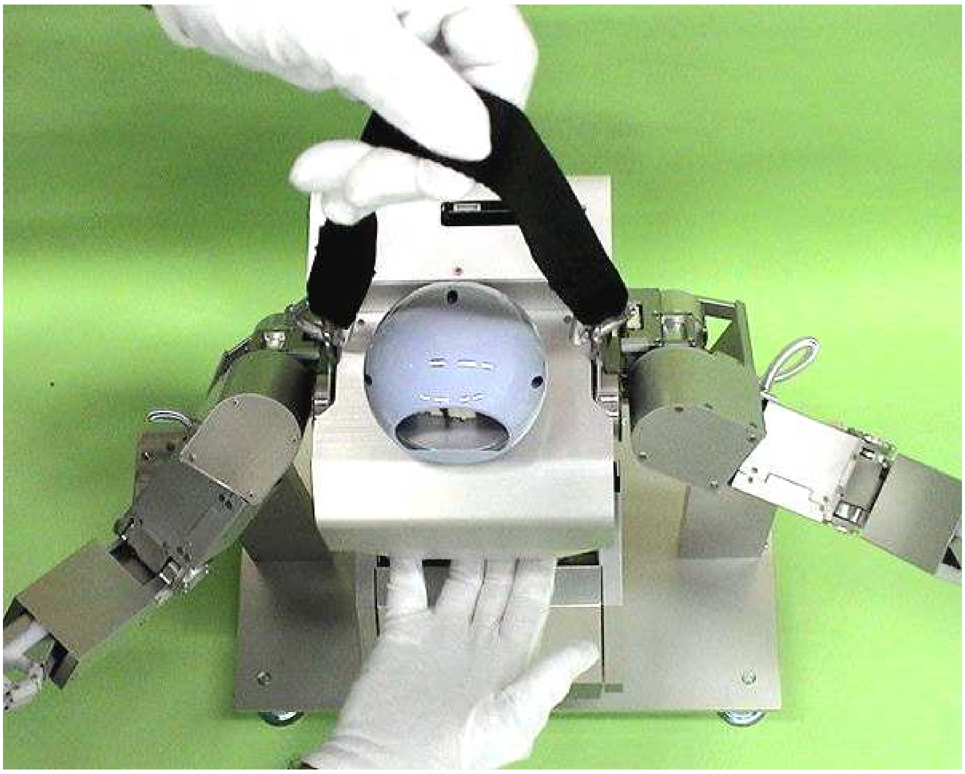
[The temporary installation of the thigh]－－－＞[The temporary installation of the arm] －－－＞[The temporary installation of the leg]－－－＞[Actual fix of the thigh] －－－＞ [Actual fix of the legs] －－－＞[Actual fix of the arms]

［Setting of thigh］

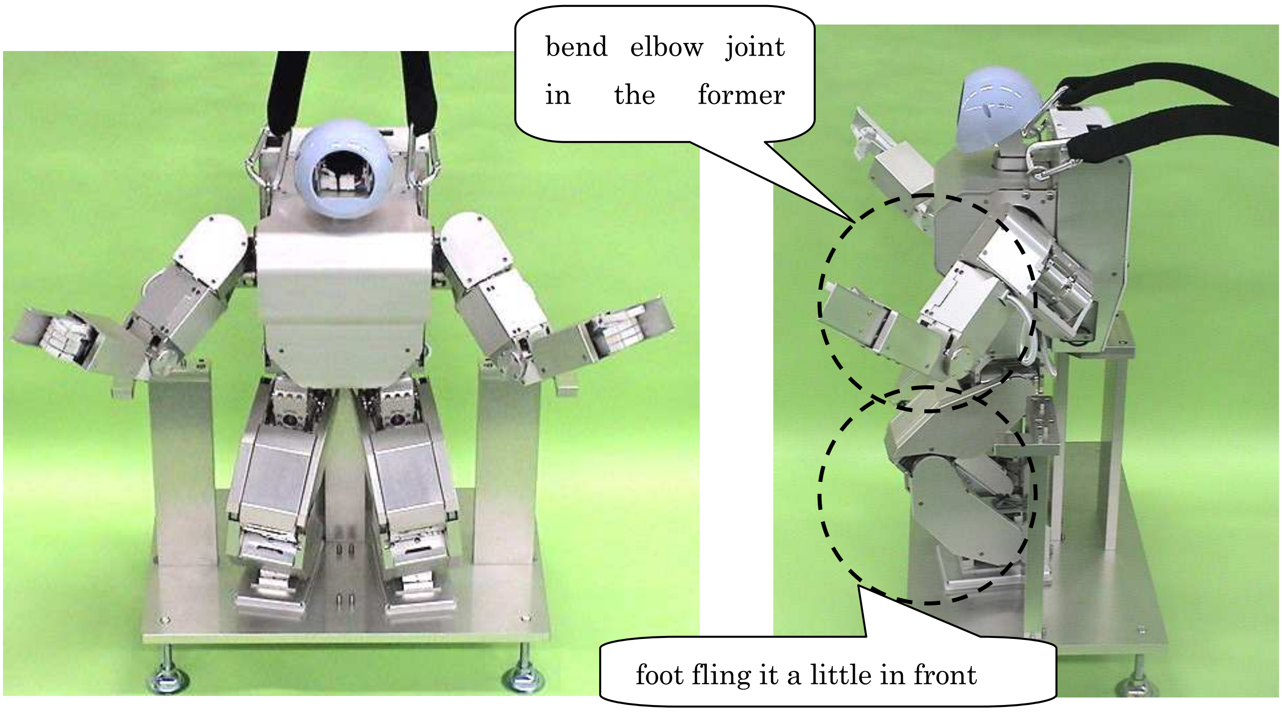
A positioning holes under the thigh of the robot is set on the pins of the joint initialization jig.（drw.S1.2-3）。**Caution：Be sure to keep pick-up the lifting belt and the bottom of the front cover in case of establishment（drw.S1.2-4）.**
On setting the position, bend elbow joint in the former direction, and make a foot fling it a little in front.
Be careful not a breakage of the cables and deformation of the truss. Align an eye with a position of insertion, and work carefully.（drw.S1.2-5）。



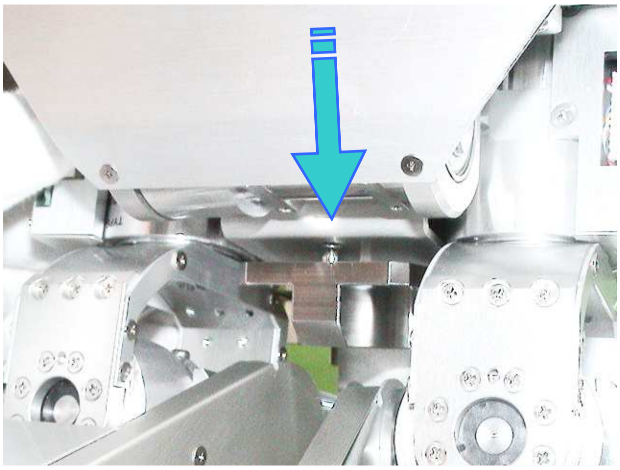
Drw. S1.2-3　Setting position of thigh



Drw.S1.2-4　 position to keep for setting



Drw.S1.2-5　Arm and foot on the setting thigh



Drw.S1.2- 6　The position of the inserting pins

Drw.S1.2- 7　Finish setting position

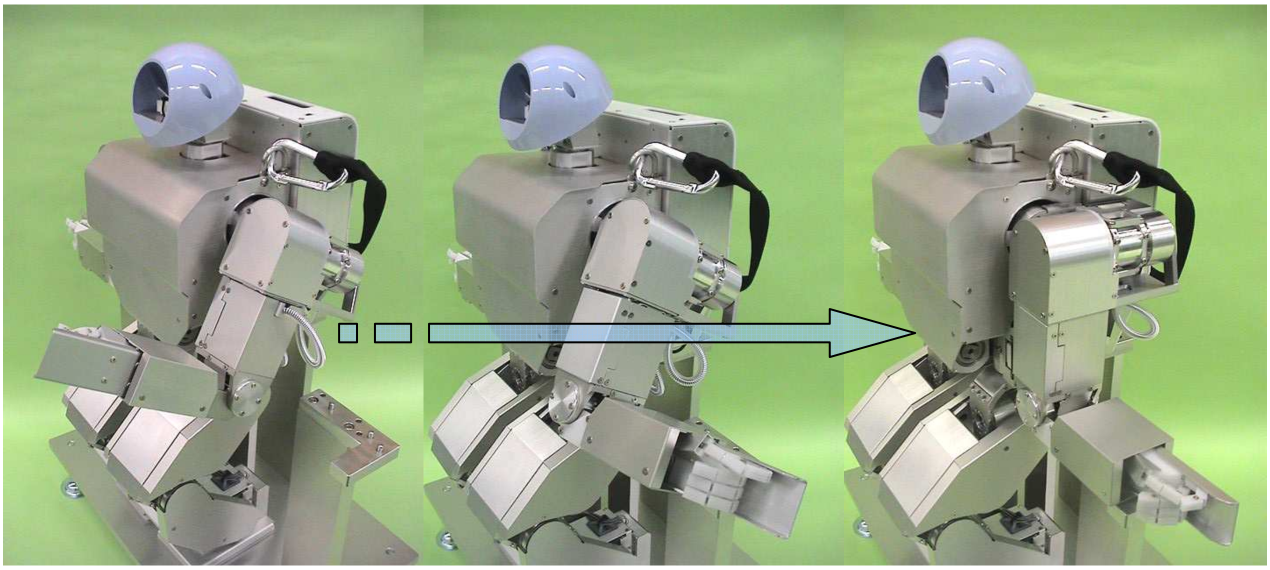
[Temporary installation for Arm and leg]

Next, an arm and a leg are set up in the form which is close to the installation posture (drw. S1.2-2).

Move each joint slowly not to hang an excessive load.

An arm is set up in an orbit like a the drw.S1.2-8.

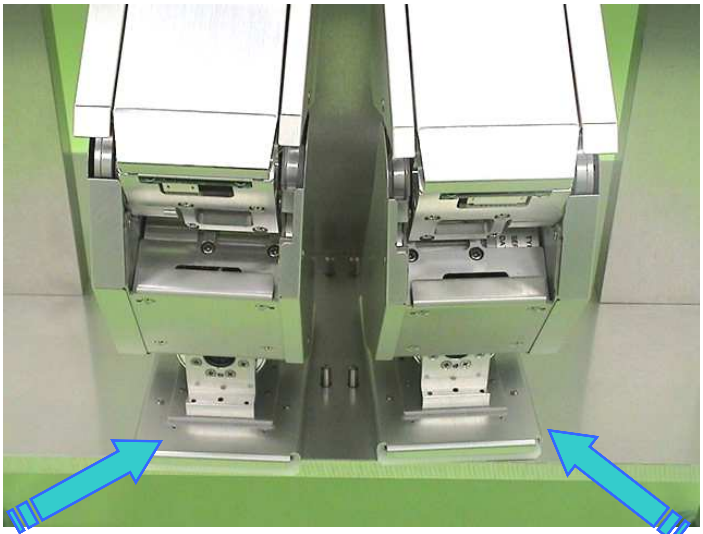
A leg is enough if it becomes the condition which the reverse side of the foot touch on base seat with stretched out a foot in the front.



Drw. S1.2-8　Process of motion when arm install temporary

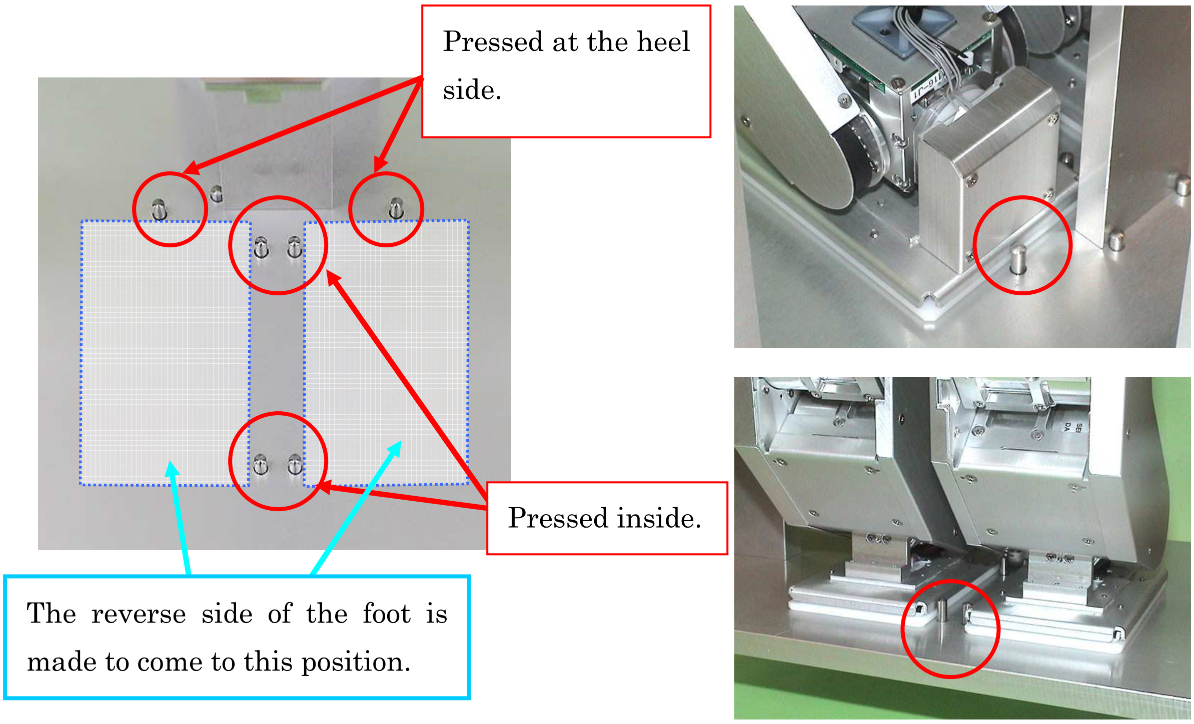
[Fixation of　leg]

Next, the end of the foot is pushed alternately as the drw. S1.2-9 little by little with a state of contact of the thigh maintained, and positions edge of foot.



Drw. S1.2-9　　Position and direction when leg is fixed

When a hand is left, it is OK if it touches a pin in the form whose position of a foot was equal in parallel under the condition that a joint initialization jig touched the back of the front cover as the contact Drw. 1.2-10 and the reverse side of the foot is not in the gap with the floor.



Drw. S1.2-10 Completion of installation of legs

[Fixation of arm]

It is moving adjusted that arm may touch a pin with pressing .
Contact an arm with the pin of the joint initialization jig as shown in the drw. 1.2-11

Drw. S1.2-12 Positioning of arm

Were you installed in the completely suitable position of the thigh, the leg and the arm?

Confirm it again.

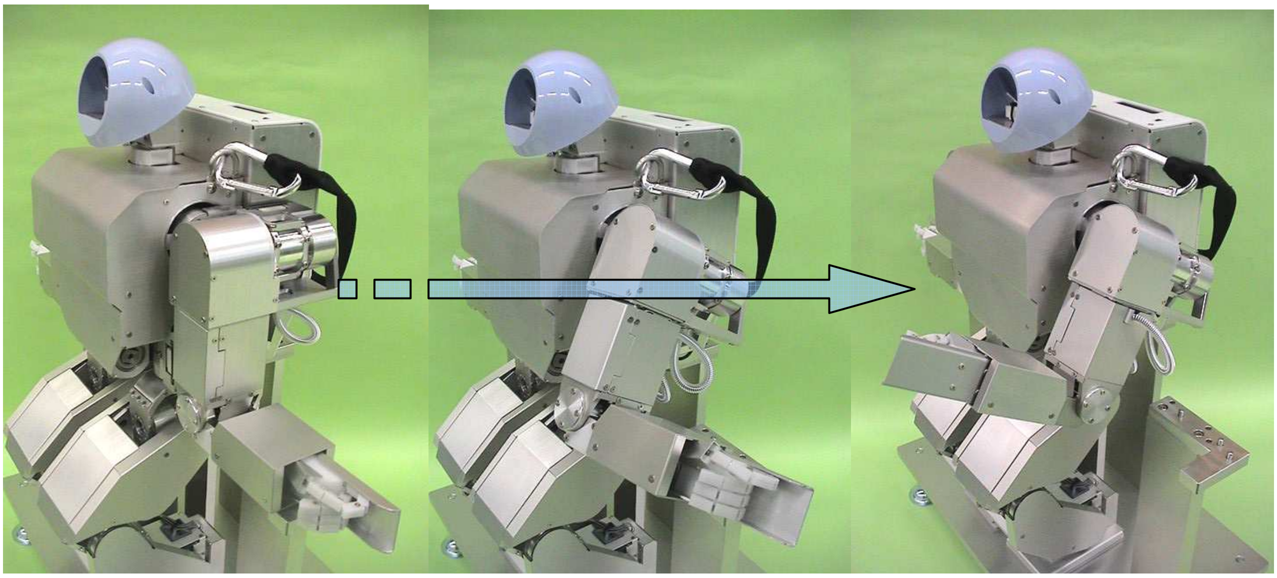
It becomes pre-set condition of the count of the encorder in the fixed numerical value and the power supply is supplied to the robot in this posture

S1.2.3 How to remove

Do for the joint under the condition of servo-off.

[Remove arm]

Move an arm, and remove it from the joint initialization jig with a process of drw. S1.2-12 .



Drw. S1.2-12 Remove arm

[Remove thigh]

It moves slowly the robot upward a little, and remove it in front .

Be careful that a robot doesn't hit an obstacle.

Caution item : When removing , be careful that robot doesn't hit any obstacle, such as joint initialization setting jig. Especially,

- **Do not make an unusual over load on the arm joint that an arm hits a positioning pin.**
- **Do not make an unusual overload to foot joint that reverse side of foot drag in the ground.**
- **Do not make unusual overload to each joint when remove thigh joint from plastic part of joint initialization jig.**
- **Do it work with careful confirmation**

S1.3 How to connect a connector

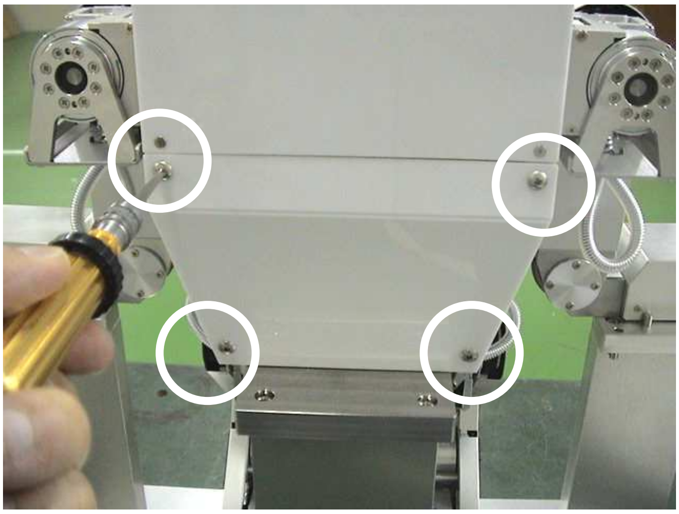
This chapter is an explanation about the connection of the connector at the back cover.

S1.3.1 How to remove back cover

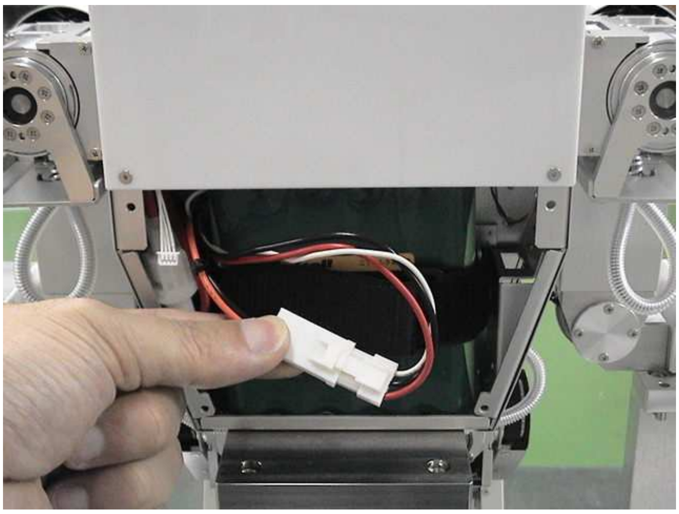
It is required to remove back cover when connectors are connected or disconnected inside of back cover.

a） A back cover is fixed on the symmetrical ○ place of the Drw. S1.3-1 with total four screws. Loosen four screws with a driver.

b） Removed a back cover as the Drw. S1.3-2 .



Drw.S1.3-1 Screw to fix back cover



Drw. S1.3-2 Removed the back cover

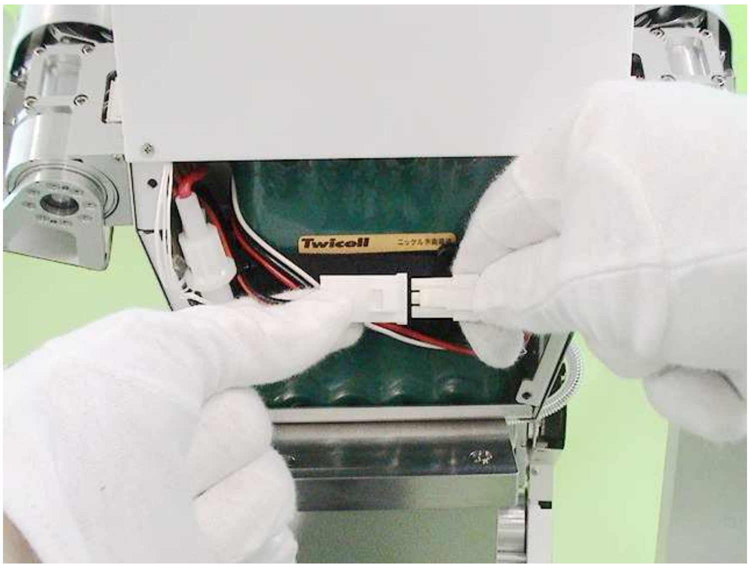
S1.3.2 How to connect and disconnect for battery connector

Battery connector should connect and disconnect on the battery inputport that is the relay connector inside of robot as shown in the drw. S1.3-3 .

When you don't use a battery in a long time , be sure to remove a connector.

It causes battery deterioration.

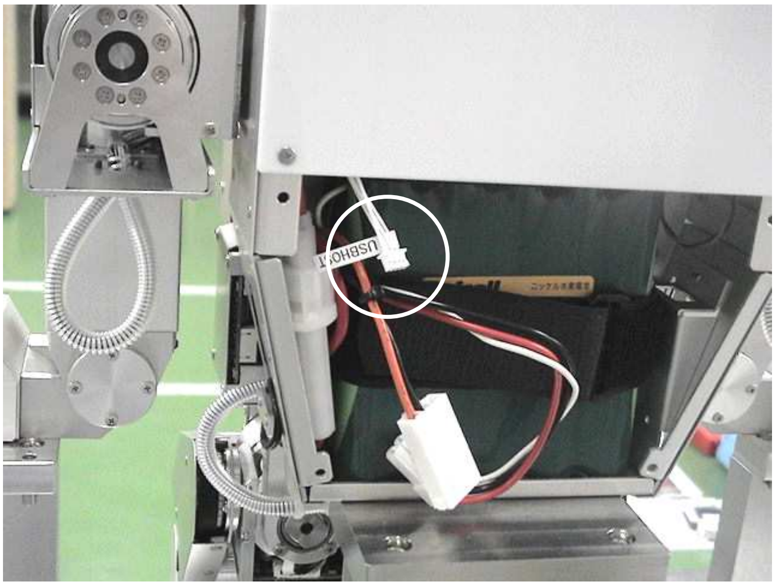
And even if it is a short time, turn off both two switches.



Drw. S1.3-3 Connection of battery connector

S1.3.3 Robot USB port

Robot USB port is shown below on Drw. S1.3-4



Drw. S1.3-4 Robot USB port

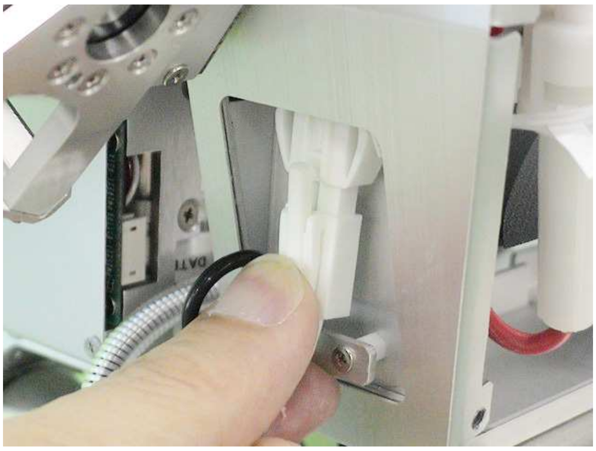
S1.3.4 Attachment of the back cover

The attachment of the back cover is the reverse procedre of the removal (drw. S1.3-5) , and tightens four screws.（drw. S1.3-6）.

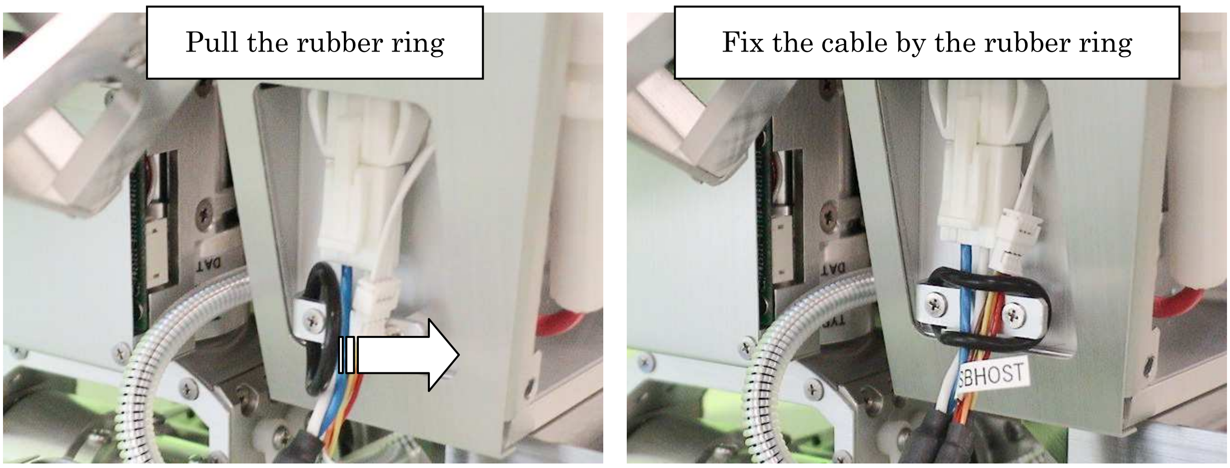
Drw. S1.3-5　Screw to fix back cover

S1.3.5 Input port for outside power supply

A power supply connector is connected to the outside power supply input port (drw. S1.3-6) at the wired mode.



Drw. S1.3-6　Outside power supply input port



Drw. S1.3- 7　Fixing the power supply cable

S1.4 How to change the Battery

This chapter is an explanation about how to change the battery . (Drw. S1.4-2)

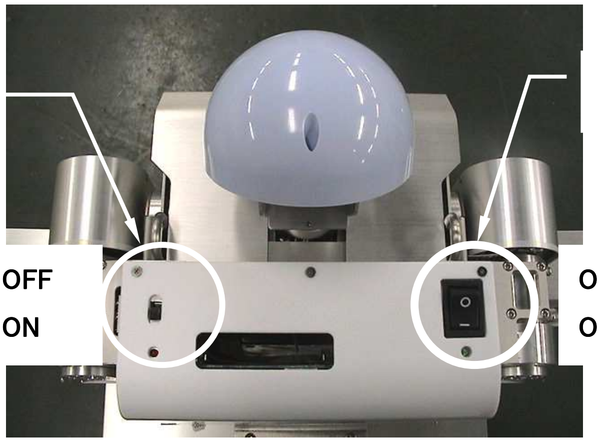
Note) Battery residual quantity can be detected by CH.4 of the A/D converter CH.0-5 of the right foot sensor.

If this result of a change becomes less than 550LSB (20V, very). , stop an experiment promptly and exchange the battery.

And before replace a battery, turn the power switches off .

S1.4.1 Turn off the power switch

A power supply is turned off in order of the motor power switch and the logic power switch.(drw.S1.4-1)



Drw. S1.4-1 Motor power switch and logic power switch

S1.4.2 Remove the back cover

Remove a back cover for the battery exchange because a battery is installed inside of back cover.

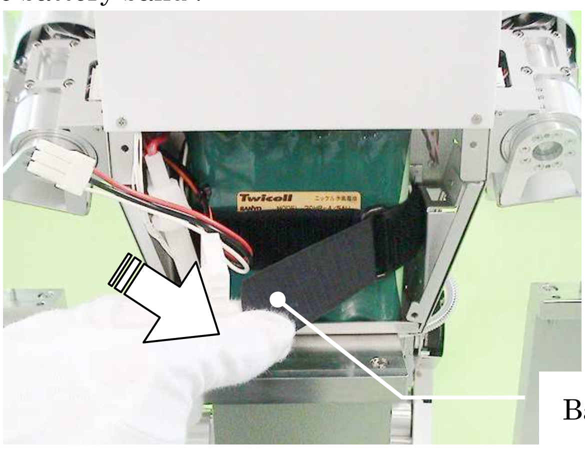
Refer to S1.3 how to remove back cover

S1.4.3 Exchange of battery

Following process is how to remove battery.

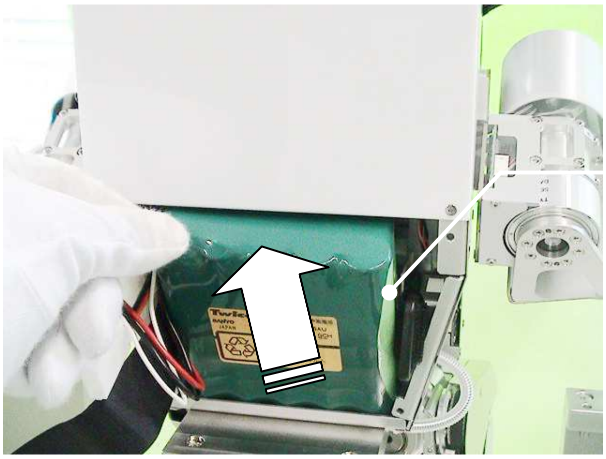
・Remove battery connector (refer to S.1.3)

・Remove the battery band .



Drw. S1.4-2 Battery band

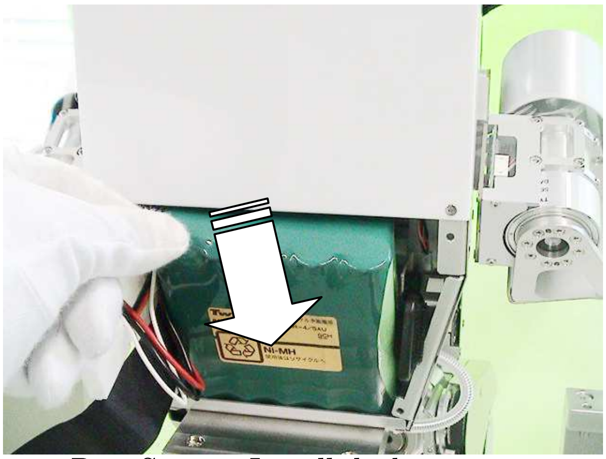
・ Remove the battery



Drw. S1.4-3　Remove the battery

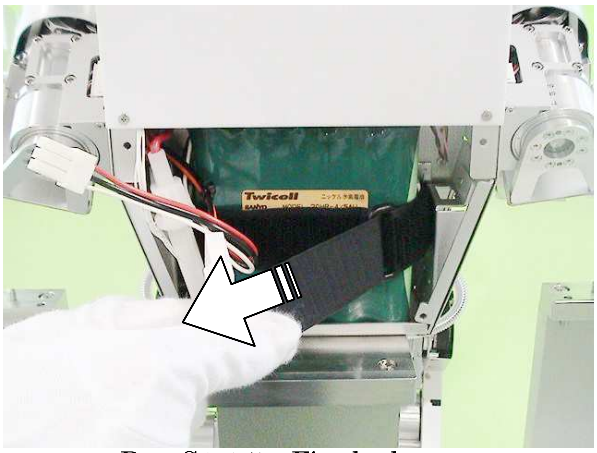
Following process is how to fix battery.

・ Install the battery



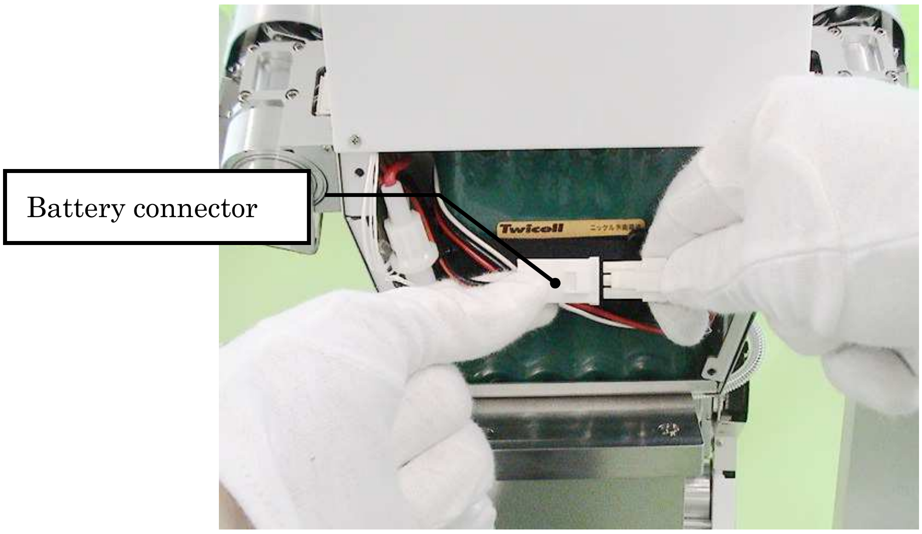
Drw. S1.4-4　Install the battery

・ Fix a battery firmly with an attached band.



Drw.S1.4-5　Fix the battery

・Connect the battery connector（refer to S.1.3）



Drw. S1.4-6　　Connect the battery connector

S1.4.4 Install the back cover

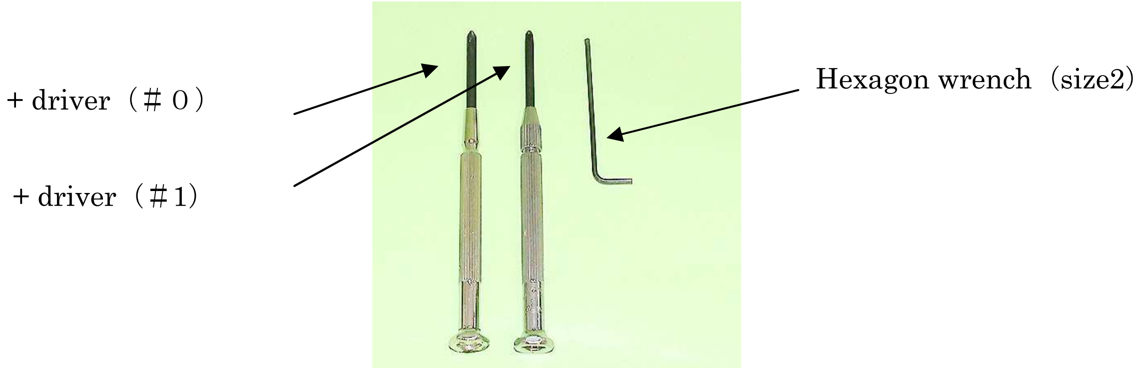
Refer to S1.3 how to install back cover

S.1.5　How to adjust a timing belt

This robot is using a timing belt for the joint of the leg with　2 places/leg .

The tension of the timing belt sometimes becomes loose by the case that an excessive load and the use of the repetition.

It is necessary that the + drivers (#0 & #1) and hexagon wrench.



Drw. S1.5-1 tool

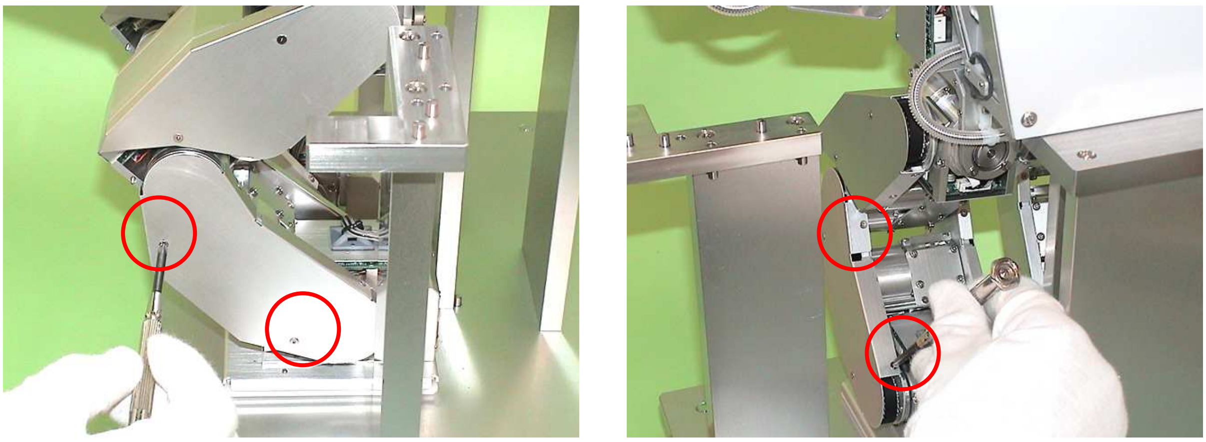
［ｓｔｅｐ1］

Remove the 4 screws of the ○mark places of the outsidethigh cover by + driver #0 .(Drw. S1.5-2)

Drw. S1.5-2　Fix position of the outside thigh cover

［ｓｔｅｐ2］

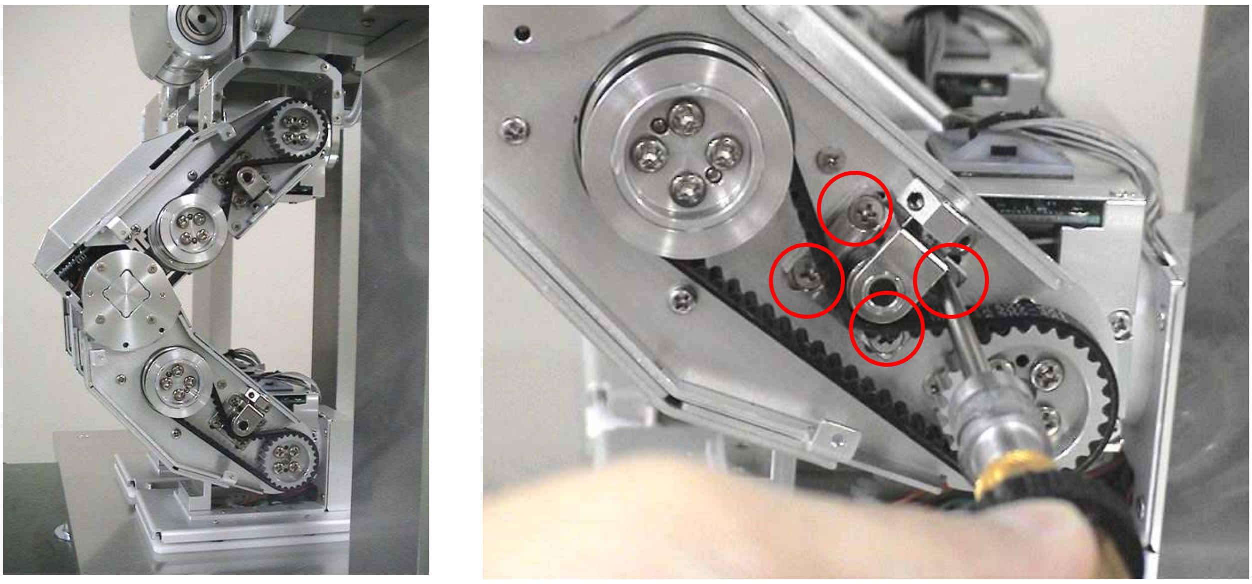
Remove the 4 screws of the ○mark places of the outside leg cover by + driver #0 .(Drw. S1.5-3)



Drw. S1.5-3　Fix position of the outside leg cover

［ｓｔｅｐ３］

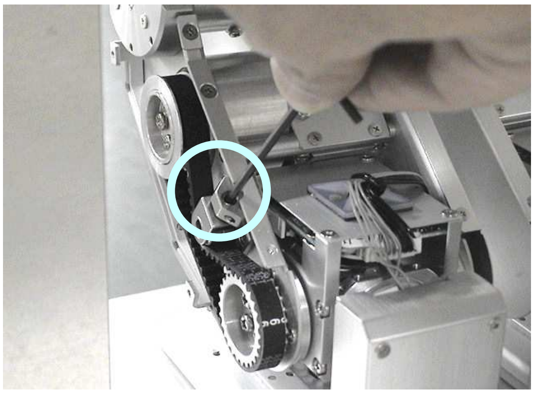
Loose the 4 screws of the ○mark places of Drw.S1.5-4 at rotated 90° to anti-clock.



Drw. S1.5-4 Adjustment place of timing belt

［ｓｔｅｐ４］

Adjust the belt using belt adjusting jig like Drw. S1.5-5 .



Drw. S1.5-5 How to use belt adjustment jig

Reference： At our company, the tension is adjusted by 80～100[N].
Please adjust according to the conditions of use.
(Used measuring instrument: Sonic tension meter U-505 by Gates Unitta Asia)

［ｓｔｅｐ 5］

Check that the belt has fully stretched.
And tighten the screw that loosed at [step3].

［ｓｔｅｐ 6］

Install the outside thigh covers and the outside leg covers.

S.2.1 Command list

command	Parameter (byte)	(C)	response (byte)		meaning
RC motor Control board					
s	1	unsigned char	0	-	Head Tilt , range is 165～240(-15° ～60°)
t	1	unsigned char	0	-	Head Pan , range is 120～240(-60° ～60°)
u	1	unsigned char	0	-	Right Hand , range is 120～240(-60° ～60°)
v	1	unsigned char	0	-	Left Hand , range is 120～240(-60° ～60°)
Sensor board					
R	0	-	13	AD0～AD5(short×6)+IO(uchar×1)	Received the AD conversion data and the IO port status.
S	1	IO(uchar×1)	0	-	Set the IO port.
X	0	-	6	ID version (uchar×6)	Received the Device ID and rhe firmware version.
(Y)	-	-	0	-	Reserved
Motor Control board					
A	2	Kpp(short×1)	0	-	Position proportional gain
B	2	Kpv(short×1)	0	-	Velocity proportional gain
(C)	2	Kpi(short×1)	0	-	Reserved.
D	2	Kpi(short×1)	0	-	Current proportional gain
F	2	short×1	0	-	Control mode 0x41:Position, 0x42:Velocity, 0x43:Current
H	2	short×1	0	-	Velocity limit (0～57)
I	0	-	0	-	Servo On
J	0	-	0	-	Servo Off
L	2	short×1	0	-	Setting the Encorder origin (= 0)
N	4	short×2	0	-	Encoder Limit (+Limit ,-Limit)
a	0		6	short×3	Response of the present position & the present velocity & the present current
d	0		6	short×3	Received the setup value of the Position proportional gain & Velocity proportional gain & Current proportional gain.
e	2	short×1	6	short×3	Target Position value , Response of the present Position & Velocity & Current
h	2	short×1	6	short×3	Target Velocity value , Response of the present Position & Velocity & Current
k	2	short×1	6	short×3	Target Current value , Response of the present Position & Velocity & Current
n	0	-	2	ushort×1	Alarm (1: + limit over , 2 : - limit over , 3 : ±limit over)
o	0	-	0	-	Alarm RESET
X	0	-	6	ID version (uchar×6)	Received the Device ID and rhe firmware version.
(Y)	-	-	0	-	Reserved

S.2.2 Setting file sample

Regarding contents of initial setting file

Below is sample of sequence mode. Before starting a direct acting servo, such an initialization file is surely transmitted, and initialize the inside of firming.

In case of general command,

Form：[communication interval(ms)],[Header=3],[device No.],[command],[number1],([number 2])

In case of fixed command,

Form：[communication interval],[Header=2],[motor 1command],[motor 2command],...,[sensor 1 command=R],R,R

As for the motor command value, meaning varies in the control mode. An command is set up automatically by the control mode.

Send command value corresponding to the control mode. For example, when a motor 2 is set up in the speed control mode, a command to the motor 2 here becomes a target for a speed.

Explanation of HomeReset.csv

2,3,1,A,12	<- Recommendation position proportion gain
2,3,1,B,350	<- Recommendation velocity proportion gain
2,3,1,C,0	<- Fixed setup value (Be sure to setup this value)
2,3,1,D,1	<- Fixed setup value (Be sure to setup this value)
2,3,1,E,0	<- Fixed setup value (Be sure to setup this value)
2,3,1,H,57	<- Velocity limit(Be sure to setup below thin value)
2,3,1,N,6479,-19019	<-Set it up within that value after you confirm allowable movement range of each shaft
2,3,1,F,65	<-The control mode is specified.(position controls is As example)
2,3,1,L,0	<- Encoder zero setting value
2,3,1,e,0	<-Fitted a target value to the encoder zero point.
2,3,2,A,12	<- Set as well as the rest to the motor number 2.
2,3,2,B,350	
:	
:	

Explanation for HomeToWalkInit.csv

10,2,0,0,9405,......,R,R,R	<- Request sensor data to feed command value to each joint in the interval 10mS.
10,2,0,0,9400,......,R,R,R	
10,2,0,0,9396,......,R,R,R	
:	
:	

S2.3 About the renewal of the built-in firmware

A robot is supporting built-in firmware via USB, but if customer by themselves renewal program imprudently, that makes a cause of communication impossibility and re-program becoming impossible, in the worst case, Robot will stop operation. Please contact FJA when renewal firmware.

Also customer is requested to provide below tools when renewal firmware.

- Compiler for the firmware development
  It is necessary when firmware is cording/ compiled. It responds separately, and an incircuit emulator (ICE) may be necessary.
  Supplier：Hitachi ULSI system Model：PS008CAS4-MWR

- HOAP Down load
  It is necessary when down loading firmware via USB. Supply it for Windows98 from Fujitsu.
  Contact FJA for details

S3 Relation of PC for instruction to operate

S3.1 Installation of RTLinux

The procedure of Installation of RTLinux to PC with HOAP is explained.
And, as for Installation to PC except for HOAP, it would be users' (customers) responsibility. In the case, the questions about Installation would not be able to be asked.
The operation except for Attached PC is not confirmed in our company.

S3.1.1 Necessary items for Installation

(1) Attached personal computer
cpu, mouse, keyboard, display, cdrom

(2) LinuxOS
Package RedHatLinux7.3(It would be ready by users.)

(3) Linux Kernel source file archive
linux2.4.18.tar.gz (It would be ready by users.)

(4) Kernel configuration file for HOAP
fja.config(It is included in CD with HOAP.)

(5) RTLinux source file archive
rtlinux-3.2-pre1.tar.bz2(It would be ready by users.)

(6) LAN driver of E1000 series
e1000-5.2.30.1.tar.gz(It would be ready by users.)
Caution: These are not necessary for the users except for the ones of FMV-610 or 620.

(7) PCMCIA-CS source file archive
pcmcia-cs-3.2.7.tar.gz(It would be ready by users.)
Caution: It is necessary in he case of using PCMCIA card adopter with wireless options.

(8) HOAP program for kernel2.4.18
hoap2.tar.gz or hoap2c610cpu1.tar.zg(It is in the CD with HOAP.)

Note:It is possible to download the file necessary for setup from the site as follow.

Package RedHatLinux7.3	http://www.redhat.com/
Linux kernel source files	http://www.kernel.org/
RTLinux source files	http://www.rtlinux.org/
E1000 source files	http://support.intel.com/support/go/linux/e1000.htm
PCMCIA-CS source files	http://pcmcia-cs.sourceforge.net/

S3.1.2 The flow of operation

The installation of RTLinux and operation of confirmation are performed by procedure as follow.

(1) Linux installation

(2) Starting of Linux and GNOME terminal

(3) Preparation of file

(4) Setup of BIOS

(5) RT patch

(6) Compile of Linux Kernel

(7) Compile of RTLinux Modules

(8) Installation of LAN Driver（for FMV-610 ,620）

(9) Installation of PCMCIA Card Service（for PCMCIA Card Adapter）

(10) Installation of HOAP programs

S3.1.3 Operation to install

(1) Linux installation

RedHatLinux7.3 is installed to PC with HOAP.

a) Preparation to install

Please obtain「Package RedHatLinux7.3」from the download sites, etc. and enter to the media of CD-R and etc., then create installation CD.

Note:Plural CD-Rs might be installation CDs. (About 3 CDs)）

b) Start from CDROM

⇒Please push [F12] from starting monitor after turning on the switch of PC.(In case of FMV)

⇒Please insert the 1st CD created to install to CDROM drive

⇒Please select the「CD-ROM」from keyboard

⇒Please push [Enter]KEY.

Caution:The method(How to change the starting drive) is different. It depends on PCs.

c) The start of Installation

⇒Please push [Enter]KEY when boot: prompt is showed

⇒Please install by the instruction of the monitor.

Please check the site as following about the method(How to install).

http://www.redhat.com/

Note:It is installed and the operation is confirmed in the conditions as following in our company.

Language Selection japanese

Keyboard Selection jp106

Mouse Selection Generic-2 Button Mouse(PS/2)

Installation Type Custom

Disk Partitioninig Setup Disk Druid

Mount Point: / File system type: ext2 Size: over 2G

Mount Point: File system type: swap Size: over 256

Boot Loader Configuration

Use GRUB

MBR

LBA32N

Firewall Configuration No firewall

Language Support Japanese

Time Zone Selection GMT

Package Group Selection

Sound and Multimedia Support

Network Support

Software Development

Kernel Development

(2) Starting of Linux and GNOME terminal

Installed Linux from (1) is opened, the file necessary for operation is copied.

a) Starting of Linux

PC is opened, Please select Linux by [cross Key] on the GRUB loader monitor in the time to start.

b) LOGIN

Login is carried out by root.

local host login : **root** ↓

password: ******* ↓

↑Password setup on the time to install Linux

Note:When the login is performed in the user's name except for root and after started the GNOME terminal, it is possible to change the right to “root” by “su command”.

c) Starting of GNOME

After the login, the command as follow is input, and GNOME is started.

startx ↓

Note:Though the warning message for login by “root” is showed, please push[Enter]Key and continue.

d) Starting of GNOME terminal

GNOME terminal is started after the starting GNOME window.

⇒Please click the GNOME terminal icon by.

Subsequent work is done in form of inputting a command into a GNOME terminal from a keyboard.

The command for input to the terminal is written in bigger (wider) size character.

And the last "↓" on the command line to input means "[Enter] Key.

(3) Preparation of the files

The files necessary for the installation are prepared.

a) Download of files

Please obtain the file as follow from the download site, and enter the file to the CD-R.

- linux-2.4.18.tar.gz
- rtlinux-3.2-pre1.tar.bz2

b) Mount of CDROM

Prepared CD is mounted to CDROM drive.

⇒Please insert CDROM to the drive.

mount /dev/cdrom /mnt/cdrom ↓

c) Copy of source file

Done download file is copied to "/usr/src".

cp /mnt/cdrom/linux-2.4.18.tar.gz /usr/src/. ↓

cp /mnt/cdrom/rtlinux-3.2-pre1.tar.bz2 /usr/src/. ↓

d) Unmount of CDROM

Mount of CDROM is taken off and it is taken out from the drive.

umount /mnt/cdrom ↓

⇒Please eject CDROM from the drive.

e) Unpack the source files.

The copied files are decompressed.

cd /usr/src ↓

tar zxvf linux-2.4.18.tar.gz ↓

bunzip2 -d rtlinux-3.2-pre1.tar.bz2 ↓

tar xvf rtlinux-3.2-pre1.tar ↓

Note:When an error comes out at the time of decompression of a file, please redo from download.

f) Movement and the link of a kernel source file

Kernel sauce is moved to RT sauce and a link is created.

mv /usr/src/linux /usr/src/rtlinux-3.2-pre1/. ↓

ln –s /usr/src/rtlinux-3.2-pre1 /usr/src/rtlinux ↓

ln –s /usr/src/rtlinux/linux /usr/src/linux ↓

g) Preparation of configuration file

Configuration file is copied from CD with HOAP to Kernel source.

⇒Please put CD of attachment in HOAP into a CDROM drive.

```
mount /dev/cdrom /mnt/cdrom ↓
cp /mnt/cdrom/fja.config /usr/src/rtlinux-3.2pre1/linux/.config ↓
umount /mnt/cdrom ↓
```

⇒Please eject CDROM from a drive.

(4) Setup BIOS

a) Display of BIOS monitor

When FMV is started, push [F2] key, and BIOS monitor is displayed.

b) The contents of a setting of BIOS

[Adbanced]
  [Serial/Parallel Port configurations]↓
    Serial Port1:[Disabled]
    Serial Port2:[Disabled]
    Parallel Port:[Disabled]
  [Internal Device Configurations]↓
    USB Controller:[Both]
    USB2.0 Controller:[Disabled]
  USB Legacy Emulation[Disabled]

c) Saving and Exit

[Exit]
  [Exit Saving Changes]↓

(5) RT patch

Patch the kernel

a) Start of Linux

⇒Please select Red Hat Linux(2.4.18-3) from GRUB.

b) Unpack the files

```
cd /usr/src/rtlinux-3.2-pre1/patches ↓
bunzip2 -d kernel_patch-2.4.18-rtl3.2-pre1.bz2 ↓
```

c) Patch

```
cd /usr/src/rtlinux-3.2-pre1/linux ↓
patch –p1 < ../patches/kernel_patch-2.4.18-rtl3.2-pre1 ↓
```

(6) Compile the RTLinux kernel

Compile and install the kernel and modules.

a) Start of Linux

⇒Please select Red Hat Linux(2.4.18-3) from GRUB.

Note:If it has been already starting on RT Linux(2.4.18-rtl3.2_pre1), it is unnecessary to re-start it.

b) Configure the kernel

cd /usr/src/rtlinux-3.2-pre1/linux ↓

make xconfig ↓

⇒It is setup as follow from configuration menu.

①[Processor type and features] (Refer to "Caution")
Set " Symmetric multi-processing support " to [n]

②[General setup] ⇒ [PCMCIA/CardBus support] (Refer to "Caution")
Set " PCMCIA/Card Bus Support " to[n]

③[File systems]
Set " Ext3 journalling file system support " to [y]

④[Code maturity level options]
Set " Prompt for development and/or incomplete code/drivers " to [y]

⑤[IEEE1394(FireWire)]support(EXPERIMENTAL)]
Set " IEEE1394(FireWire)]support(EXPERIMENTAL) " to [m]
Set " Texas Instruments PCILynx support " to [m]
Set " OHCI-1394 support " to [m]
Set " OHCI-1394 Video support " to [m]
Set " SBP-2 support(Hard disks etc) " to [m]
Set " Raw IEEE1394 I/O support to " [m]

⇒ Please select [Save and Exit].

⇒ Please select [OK].

Caution:①Please setup it in the case of FMV-C610.

②Only when a PCMCIA card adapter is used.

c) Compile and install

make clean ↓

make dep ↓

make bzImage ↓

make modules ↓

make modules_install ↓

make install ↓

d) Default Setup of BOOT OS

GRUB configuration file is edited and RTLinux is setup to OS of default.

gedit /boot/grub/grub.conf ↓

⇒The line of「default」is changed.

default=0

⇒Plesse select the [save] Button.

⇒Plesse select the [file(F)]⇒[Exit(E)]

Note: In the time after the second compile, three lines are deleted as follow.

Title Red Hat Linux(2.4.18-rtl3.2-pre1)
root (hd0,1)
kernel /vmlinuz-2.4.18-rtl3.2-pre1 ro root=/dev/hda3 hdc=ide-scsi

e) reboot

It reboots from a new kernel.

reboot ↓

(7) Compile and install the RTLinux modules.

a) reboots from a RTLinux kernel

⇒Please select the RT Linux(2.4.18-rtl3.2_pre1).

b) Configure

cd /usr/src/rtlinux-3.2-pre1 ↓

make xconfig ↓

⇒Do you agree to the terms of this license[y/n]?

y ↓

⇒Please select [Save and Exit].

⇒Please select [yes].

c) Compile and install

make clean ↓

make dep ↓

make all ↓

make dirs_install ↓

make install ↓

d) The check of RTLinux of operation

rtlinux start ↓

⇒Please check the following displays.

Scheme(-)not loaded, (+) loaded

(+)mbuff

(+)rtl

(+)rtl_fifo

(+)rtl_posixio

(+)rtl_sched

(+)rtl_time

e) The stop check of RTLinux

rtlinux stop ↓

⇒Please check the following displays.

Scheme(-)not loaded, (+) loaded

(-)mbuff

(-)rtl

(-)rtl_fifo

(-)rtl_posixio

(-)rtl_sched

(-)rtl_time

(8) Installation of a LAN driver（FMV-610,620 only）

a) reboots from a RTLinux kernel

⇒Please select RT Linux(2.4.18-rtl3.2_pre1)

b) The copy of E1000 driver

⇒Please insert CD in a CDROM drive.

mount /dev/cdrom /mnt/cdrom ↓

cp /mnt/cdrom/e1000-5.2.30.1.tar.gz /usr/local/. ↓

umount /mnt/cdrom ↓

⇒Please eject CDROM from a drive.

c) Unpack the source files.

cd /usr/local ↓

tar zxvf e1000-5.2.30.1.tar.gz ↓

d) Installation

cd /usr/local/e1000-5.2.30.1/src↓
make install↓

e) Setup

⇒[Program]⇒[System]⇒[Network setup]
⇒Start
⇒Close

(9) Installation of PCMICA Cart Service（card adapter use only）

a) reboots from a RTLinux kernel

⇒Please select the RT Linux(2.4.18-rtl3.2_pre1).

b) Copy pcmcia-cs

⇒Please insert CD in a CDROM drive.。

mount /dev/cdrom /mnt/cdrom ↓

cp /mnt/cdrom/pcmcia-cs-3.2.7.tar.gz /usr/src/. ↓

umount /mnt/cdrom ↓

⇒Please eject CDROM from a drive.

c) Unpack the source files

cd /usr/src ↓

tar zxvf pcmcia-cs-3.2.7.tar.gz ↓

d) Compile and Installation

cd /usr/src/pcmcia-cs-3.2.7↓
make config
⇒All[Enter]
make all ↓
make install ↓

e) Setup

gedit /etc/sysconfig/pcmcia ↓
⇒It changes into below.
PMCIA=yes
PCIC=i82365
PCIC_OPTS="irq_mode=0 pci_csc=0"
CORE_OPTS=

f) The check of pcmcia

/etc/rc.d/init.d/pcmcia stop ↓
/etc/rc.d/init.d/pcmcia start ↓

⇒Please insert CF in a PCMCIA slot.

mkdir /mnt/cf ↓ (When the directory is not being created)

mount /dev/hde1 /mnt/cf ↓

less /mnt/cf

⇒It checks that the folder in CF is displayed.

umount /dev/hde1 ↓

cardctl eject ↓

⇒Please eject CF from a slot.

(10) Installation of HOAP program

a) Copy of HOAP program

⇒Please insert the HOAP attachment CD in a CDROM drive.

mount /dev/cdrom /mnt/cdrom ↓

cp /mnt/cdrom/hoap2.tar.gz /usr/local/. ↓

(cp /mnt/cdrom/hoap2c610cpu1.tar.gz /usr/local/. ↓)

umount /mnt/cdrom ↓

⇒Please eject CDROM from a drive.

b) Unpack the HOAP program

cd /usr/local ↓

tar zxvf hoap2.tar.gz ↓

(tar zxvf hoap2c610cpu1.tar.gz ↓)

Caution:( ) FMV-610 only.

c) The check of HOAP program

cd /usr/local/hoap2/modules ↓

rtlinux start rt_ctlmodule.o ↓

⇒Please check the following displays.

(+) rt_ctlmodule

/usr/local/hoap2/bin/rt_ctlapp ↓

⇒Please check the following displays.

0,0,00:$

$q

S3.2 Wireless mode process

Preparation

1. Read the manual of the wireless LAN access point, and SSID of the wireless LAN access point is set up in "GeoWave”.
2. The Compact Flash which puts LinuxOS is inserted into the slot of the robot.
   Insert Wireless LAN card into Robot.
3. Switch ON robot power source.
   Provide the PC for remote access which connected a wireless LAN access point and a network.
   It cares about either Windows or Linux.
   Refer to [S3.2.1 wireless motion process note] regarding motion process of Linux.

※Software loading CompactFlash is setup to communicate with the access point of SSID is "GeoWave" and IP address is 10.25.184.151 by the default.
Network setting can be re-write in case of necessity. Refer to [S3.3 : Compact Flash setup] .

Wireless motion

1. Remote Login to the PC card in the robot from remote access PC.
   In case of Windows ： Start command prompt.
   Input as “**telnet 10.25.184.151**”, then return.
   In case of Linux ： Start terminal emulation.
   Input as **“rlogin -l guest 10.25.184.151”,t**hen return.
2. Login as User **guest**、Password **guest**. After Login、
   Input as **$ su**
   Then return, input password **default**, log in with root authority.
   The rest can be operated with a same command that is operating method from local user PC.
3. after finishing operation, input as
   /sbin/shutdown -h now
   and shutdown OS at robot side before shout off power supply.
   It has the possibility that the contents of Compact Flash crash when a robot power supply is turned off without doing this. In case compact flash crashed and do not start up, re-install process is referred to [S3.3 compact flash set up].

S3.2.1 Wireless mode process

1.Turn to the “ X” of 「MDI/MDI-X」switch of the access point. And connect between the remote access PC and the access point by LAN cable.　Turn on the PC power.

2.Confirm that the joint angle of the robot is an initial position (Refer to S1.2)
Turn on the robot switches.

3.When Red Hat Linux is started on PC, login like the wired mode.

Local host login ： **root**

Password ： **default**

4.Input # **startx**

5.Check the LED of the wireless LAN card on the robot is ON.
There is this several minutes case that after turning on the robot until LED lights up.

6.Start terminal emulator for GNOME

7.Input the following in this window,

```
# rlogin -l guest 10.25.184.151
password : guest
$ su
password : default
```

8.Input the following in this window,

```
# cd /usr/local/hoap2/modules
# rtlinux start rt_ctlmodule.o
```

RT-Linux is started.　If RT-Linux has started normally, the following will be displayed.

(+)rt_ctlmodule
:

9.It is input as follows, and a communication module with the robot is started.

```
# ../bin/rt_ctlapp
0.0.00:$    (←reply prompt)
```

A return key is stamped some times.

5.25.xx:$　　(xx isn't fixed.)

Repeat it until it is indicated.

10.Start another terminal emulator for GNOME' .
This window is nemed ”The window 2 “.
And the window that some time ago is “the window2”.

11.Login at the window2 like the window1 at the order of 7..

12.In the window2, input as follows,

```
# cd /usr/local/hoap2/data
# ../bin/sendseq < HomeReset2a.csv
```

13.Input the following in the window 1.

```
5.25.xx:$ v (Return)
…
cSeq:YY
nSeq:ZZ
```

Confirm that it is same value csep value = nseq value.

14.Estrange it from the joint initialization jig , and hang a robot at the lifting jig Refer to S 1.1.

15.By Window 2, input as

 # ../bin/setpos

 # ../bin/sendseq < AllServoOn.csv (Servo on of all the joints)

 # ../bin/interpol < m01.csv （Making of the Complement data of start position）

 # ../bin/sendseq < interpol.csv （Move the start position）

 # ../bin/sendseq < m01.csv （Start the sample program）

How to shutdown

1.Hung the robot to lifting jig.

2.Input following by Window 2

 # ../bin/sendseq < AllServoOff.csv

3.Finish the monitor program by Window1.

5.25.xx**:$ q (Return)**

4.stop the rt-linux by Window1.

 # rtlinux stop rt_ctlmodule.o

5.Shutdown by Window1.

 # /sbin/shutdown -h now

6.Close both windows.

exit

7.Turn off the robot power supply , after 30 seconds or more have passed since LED of wireless LAN put out.

Caution: If power supply is turned off earlier , the program in CF card may crash.

8.Turn off the remote access PC.

S3.2.2 About the user program for radio operation

The LinuxOS version in the radio option attachment CompactFlash is using RedHatLinux7.1. In order to perform on radio the application program which the user created, the compile by RedHatLinux7.1 is needed.

The setup of Red Hat software7.1 is indicated below.

Moreover, about installation of Red Hat software7.1, it becomes a user's responsibility.

(1)Linux installation

Please carry out PC installation using RedHatLinux7.1.

note：Please make the following reference about a file required for installation, and the installation method.

RedHatLinux http://www.redhat.com/

(2) RTLinux installation

① Installation preparation

1. Unpack the RTLinux source files into a "/usr/src" and a link is created.

ln –s /usr/src/rtlinux-3.1-pre3 /usr/src/rtlinux ↓

note：Please make the following reference about a source file.

FMSLabs http://www.rtlinux.org/

FileName rtlinux-3.1-pre3.tar.gz

2. Unpack the RTLinux source files into a "/usr/src/rtlinux-3.1-pre3/" and a link is created

ln –s /usr/src/rtlinux/linux /usr/src/linux ↓

note：Please make the following reference about a source file.

Kernel org http://www.kernel.org/

FileName linux2.4.4.tar.gz

② RT patch

cd /usr/src/rtlinux-3.1-pre3/linux ↓

patch –p1 < .. /kernel_patch-2.4.4↓

③ The copy of a Configuration

cp /mnt/cdrom/fja244.config /usr/src/rtlinux-3.1pre3/linux/.config ↓

note：It is in the HOAP attachment CD.

FileName fja244.config

④ Compile and install the RTLinux kernel

cd /usr/src/rtlinux-3.1-pre3/linux ↓

make xconfig ↓

⇒ [Save and Exit]

⇒ [ok]

make clean ↓

make dep ↓

make bzImage ↓

make modules ↓

make modules_install ↓

make install ↓

⑤ Compile and install the RTLinux modules

reboot ↓(Please select RTLinux.)

cd /usr/src/rtlinux-3.1-pre3 ↓

make xconfig ↓

⇒Do you agree to the terms of this license[y/n]?

y ↓

⇒ [Save and Exit]

⇒[yes]

make clean ↓

make dep ↓

make all ↓

make dirs_install ↓

make install ↓

⑥ The check of RTLinux of operation

rtlinux start ↓

⇒ Please check that all become (+).

rtlinux stop ↓

(3) Installation of HOAP program

① Unpack the HOAP program source files into a “/usr/src“ .

cp /mnt/cdrom/hoap2ohci.tar.gz /usr/local/. ↓

note：It is in the HOAP attachment CD.

FileName hoap2ohci.tar.gz

cd /usr/local ↓

tar zxvf hoap2ohci.tar.gz ↓

② A check of operation

cd /usr/local/hoap2/modules ↓

rtlinux start rt_ctlmodule.o ↓

⇒Please check a display.

(+) rt_ctlmodule

/usr/local/hoap2/bin/rt_ctlapp ↓

⇒Please check a display.

0,0,00:$

$q

S3.3 Compact Flash set up

S.3.3.1. Necessary parts

(1) PC which is attached PC card slot

(2) Compact Flash (128M)

(3) CD-ROM of HOAP

S.3.3.2. Process outline

(1) Provide PC to write in CF.

(2) Cut partition of CF

(3) Copy OS in CF

(4) Make possible CF booted .

(5) Copy the HOAP software in CF.

(6) Boot from CF, and check if it is possible to login via network.

S.3.3.3 Provide PC to write in CF.

Provide PC which is attached PC card slot.

Install RedHat Linux on PC. (refer to S.3.1)

S.3.3.4 Cut partition of CF

Login PC that finished set up at "S3.3.3" by root authority.

Click on the icon of "display and footprint" type at lower of side of screen , and a CF is set on the CF adapter, and that is inserted into the PC card slot, terminal emulator for GNOME is started.

At terminal emulator, input as

tail /var/log/messages

It is checked the CF which IDE device was recognized as.

kernel: idc_cs: hdc: Vcc=3.3, Vpp = 0.0

cardmgr[*]: executing: './ide start hdc'

If write above, it is recognized as hdc.

less /proc/partition

Like to see above, find out which IDE device is recognized.

Input as **/usr/sbin/cdfidk /dev/hdc**

Then Start cdfdisk.

Chosen deletion with an arrow (→) key and if it has an existent territory, chosen with a ↓ key and deleted.

Chosen "A new preparation" → "basic area".

Size : 111.85MB (Return)

「From beginning」 (Return)

「Boot possible」 (Return)

Basic areas is formed as above

Chosen the free area left with a ↓ key , and chosen a new preparation → basic area.

Form basic area by Size:16.00 MB (Return)

And chosen 「FS type」by →key.

「Chosen file system type」: 82 (Return)

Change to linux swap type

hdc1 boot basic area Linux 111.85

hdc2 basic area Linux swap 16.00

confirm above and chosen[write] by allow→

May I write partition information on disc？:yes(return)

Write as above.

[Write area table in disc]

Chosen finish by →key if above word is indicated, then finish cfdisk

CompactFlash is removed for a while, and inserted again.

Input as /sbin/mke2fs /dev/hdc1

make ext2 file system

input as /sbin/mkswap /dev/hdc2

make swap space

S3.3.5 Copy OS on CF

```
mkdir /mnt/flash
mount /dev/hdc1 /mnt/flash
```

Input as above, and mount compact flash on /mnt/flash

Insert CD of HOAP into cd-rom drive.

Double-click on the icon of the CD form at left side of the desk-top computer (screen), and indicate the contents of the CD.

Chosen below from indicated window's menu,

linux-flash.tar.gz

then, click right side. Chosen copy, purposed place, and write

/tmp

Push OK button.

```
cd /tmp
tar zxv -C /mnt/flash -f linux_flash.tar.gz
```

Input above and extract OS to CF.

S3.3.6 Make CF to boot possible

```
cd /mnt/flash; /usr/sbin/chroot . sbin/lilo –C etc/lilo.conf.flash
```

Input as above, write in master boot record of CF make boot possible.

Added linux-2.4.18-rtl *

Succeed if above is indicated.

S3.3.7 Copy HOAP software to CF

Insert HOAP CD into cd-rom drive.

Double-click on the icon of the CD form at left side of the desk-top computer (screen), and indicate the contents of the CD.

Chosen below from indicated window's menu.

hoap_ohci_release.tar.gz

data.tar.gz

then, click right side.　Chosen copy, purposed place, and write

/tmp

Push OK button.

tar zxv -C /mnt/win/usr/local/ -f hoap_ohci_release.tar.gz

input above, install HOAP software into /mnt/win/usr/local

Appears hoap.ohci directory in /usr/local/, Symbol click "ln -s hoap.ohci hoap"

Also、input as below, and install walking data.

tar zxv -C /mnt/win/usr/local/hoap -f data.tar.gz

Input as below and mount CF.

umount /dev/hdc1

S3.3.8 Start from CF, and check whether it can be log in via network.

Refer to wireless LAN access point manual, set up wireless LAN as below.

ＥＳＳＩＤ：GeoWave

Channel　：2

(ip_address：10.25.184.64)

(ip_netmask：255.255.255.0)

(may possible in default except ESSID and Channel)

Set up of the CF is:

ip address：10.25.184.151

netmask：255.255.255.0

gateway：10.25.184.26

hostname：kiwi.stars.flab.fujitsu.co.jp

Above setup is written in "/etc/sysconfig/network" file , "/etc/sysconfig/network-scripts/ifcfg-wlan0" file, and "/etc/hosts" file.

Change them if necessary.

Set up wireless LAN of CF is:

SSID:"GeoWave"

This setup is setting APSSID="GeoWave" in /etc/pcmcia/wlan-ng.opts .

Change them if necessary.

Connect The PC which Ethernet card was attached (Both Windows and Linux are good.) for the remote access to with Ethernet hub, and a wireless LAN access point is connected with that hub, too.

Remote log in from remote access PC to robot installed PC.

In case of Windows：　Start command prompt.

telnet 10.25.184.151

Input above and return.

In case of Linux　：　Start terminal emulator

rlogin 10.25.184.151

Input above and return.

Log in by　User:guest、Password:guest.　If log in is completed, input

su

then return, and Log in by root authority to input password as ‘default.’

The rest can be operated with a same command that is operating method from local user PC

Input as exit when remote log in is finished.

After finishing operation, input as

/sbin/shutdown -h -now

and shut down OS at robot side　before shout off power supply,

S3.4 Explanation for sequence data file.

Following data file of sequence motion is including in “/usr/local/hoap2/data” directory.

It can be used as below by sendseq program.

/usr/local/hoap2/bin/sendseq < *.csv

File name	Explanation
AllServoOff.csv	All joints Servo off
AllServoOn.csv	All joints Servo On
HomeReset2a.csv	Transmit under the condition that it is installed in the initial position-setting jig. Count pre-set + servo gain + motion range setup、
m01.csv	ZMP walking
2ms/ Less than a directory is for 2mS at the time of the wireless mode.	
AllServoOff.csv	All joints Servo off
AllServoOn.csv	All joints Servo On
HomeReset2a.csv	Transmit under the condition that it is installed in the initial position-setting jig. Count pre-set + servo gain + motion range setup
m01.csv	ZMP walking

S4 Caution item for RT-Linux

When power supply is turned off compulsively due to the blackout or the hang-up during the use, It has the possibility to damage the file-system of Linux.

When it was started and it stopped by the following message.

(as example here, in case of had of IDE)

```
Give root password for maintenance
 (or type Control-D for normal startup):
```

The system this message judged that there was a wrong point in the ext2 file system, they demanded to have a disk check by the manual.

Log in by root password (initial setting is by default) to type 「Ctrl」+「D」

If following is /dev/hda2(IDE hard disc of first unit), check disc to input following.

If there is a question a middle of it (like send fax?), reply (Y) to all.

fsck -t ext2 /dev/hda2 (partition of /boot)

fsck -t ext2 /dev/hda5 (partition of /)

S５ Current control

S5.1 How to use of the current control mode

Use the command “D” for setting.

Argument is 16bit data, and following the bit-asign.

At bit15～bit12, specificate to this command. (bit15,bit14 are Don't care)

bit15	bit14	bit13	bit12	
x	x	0	0	setting of the current proportional gain
x	x	0	1	setting of the CCW origin
x	x	1	0	setting of the CW origin
x	x	1	1	setting of the same origin both CW and CCW

At bit11～bit1, set 0～4096

bit11	bit10		bit1	bit0	
0	0	......	0	0	0
0	0	......	0	1	1
0	0	......	1	1	2
		..			
1	1	......	1	1	4096

For example, When argument is 0x1015, CCW origin is 21.

The origin is set the current before begining to move the motor.

These values need to set up a proper setting value with the system according to the place and use situation of a motor.

S5.2 Sample program of thecurrent control

This sample is

For M01 motor

Target current value is 30. (About 1Ampere at Type-3 motor)

When motor moves to limit, it is servo off.

If rotation is blocked, the target current continues flowing.

2	3	1	D	24	← setting of the current proportional gain (１８H)
2	3	1	D	4117	← setting of the CCW origin（１０１５H）
2	3	1	D	8212	← setting of the CW origin （２０１４H）
2	3	1	N	10000 -10000	← setting of the encorder limitter
2	3	1	F	67	← setting of the current control mode（４３H）
2	3	1	k	30	← setting of the target current value
2	3	1	L	0	← setting of the encorder origin
2	3	1	I		← Servo On

S５.３ Sample of the calculation

The following is Sample that the calculation of the current.

the current proportional gain is ”１８H”,

the CCW origin is “１５H”,

the CW origin is “１４H”

ＴＹＰＥ－２ motor

Target current value： Ｉ＝（1．２５／７０）×ｋ ＋ 0．２８６

ＴＹＰＥ－３ motor

Target current value： Ｉ＝（1．５５／７０）×ｋ ＋ 0．２８６

※ It is necessary to change the above-mentioned constant according to the individual difference of a motor.

In addition, Set the target current value below to 3Ampere.

S６　Camera Unit　(Option)

S６．１ Specification of the camera unit

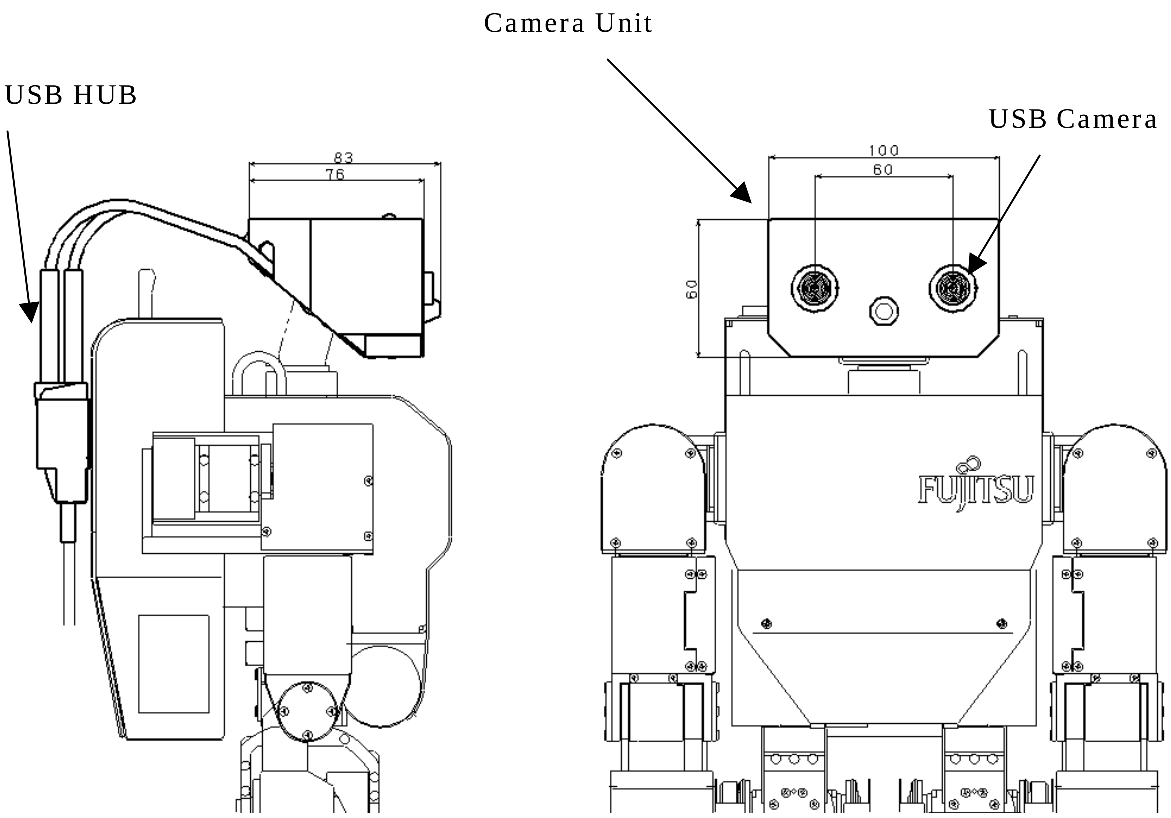
Image receiving component　:　1/4-inch　CMOS sensor×２（asynchronous）

Maximum resolution　:　ＶＧＡ(640×480)

Flame rate　:　15fps(QVGA 320×240)

Number of color　:　about 16.77million color (24bit)

Interface　:　USB1.1



Drw. S6.1-1　Camera Unit

S６．２ Caution of the camera unit

（1） The USB cable of the camera option is connected to the USB port that is different route from the data communication port.

Ex.) At the Attachment PC , the data communication USB cable is connected to the back port of the PC,and the camera unit USB cable is connected to the front port of the PC.

（2） In order to make UHCI usable, use this unit after removing the comment out of USB_UHCI at /etc/modules.conf .

（3） Depending on the PC, real-time control of 1msec may not be able to be performed at the time of camera unit use.

（4） At the time of camera unit use, the movable range of head pan is from –40 ° to 40 ° , and that of head tilt is from –15 ° to 50 ° .

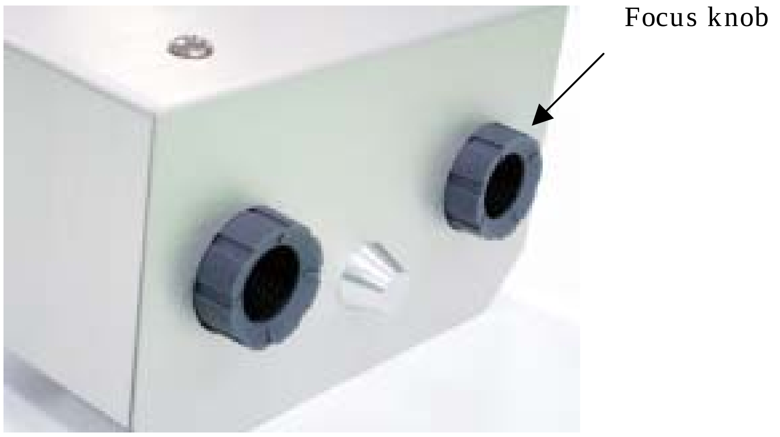
（5） Adjust the camera forcus by focus knob . (refer to Drw. S6.2-1)

Caution : If a focal knob is turned too much enough counterclockwise rotation or pulled too much to the front, a focal knob may separate from a camera.

（6） The application for capture is not appended.

Reference: The check of operation is carried out with the following freeware.

xawtv-3.88 http://bytesex.org/xawtv/



Drw. S6.2-1 Focus adjustment