RoboBuilder User Tutorial

RoboBuilder Co., Ltd.
1. ROBOBUILDER PRODUCT LIST ........................................................................................................... 4
   1.1. Kit Comparison ................................................................................................................................. 4
   1.2. Kit and Part List .............................................................................................................................. 5

2. SOFTWARE LIST ..................................................................................................................................... 13
   2.1. ROBOBUILDER DOWNLOAD TOOL ............................................................................................... 13
       2.1.1. Main Features .......................................................................................................................... 13
       2.1.2. Display Structure ...................................................................................................................... 13
       2.1.3. How To Download ..................................................................................................................... 15
     2.2. MOTION BUILDER ....................................................................................................................... 20
          2.2.1. Project, Motion File, Scene, Frame, Transition Time ............................................................... 20
          2.2.2. Screen Layout ......................................................................................................................... 21
          2.2.3. Easy Programming style ......................................................................................................... 22
     2.3. ACTION BUILDER ......................................................................................................................... 23
          2.3.1. Screen Layout ........................................................................................................................ 23
          2.3.2. Downloading Action File (Example : test.rba) .................................................................... 24
     2.4. ROBOBUILDER DIAGNOSTIC TOOL ........................................................................................... 26
          2.4.1. Diagnostic Tool main feature ................................................................................................. 26
          2.4.2. Diagnostic Tool Layout ......................................................................................................... 26
     2.5. RBC FIRMWARE UPGRADE TOOL ............................................................................................... 27
          2.5.1. RBC Firmware Upgrade Tool Layout ...................................................................................... 27
          2.5.2. How to upgrade Firmware ...................................................................................................... 28

3. ROBOBUILDER EDUCATION COURSE ................................................................................................ 29
   3.1. BEGINNER COURSE ..................................................................................................................... 29
        3.1.1. Crossing Gate .......................................................................................................................... 29
        3.1.2. Automatic Tongs ..................................................................................................................... 34
        3.1.3. Tricycle ..................................................................................................................................... 38
        3.1.4. Robotic Arm ............................................................................................................................ 42
        3.1.5. Tri-pedal Robot ......................................................................................................................... 47
        3.1.6. Four-wheeled Vehicle ............................................................................................................. 52
        3.1.7. Tongs Car .................................................................................................................................. 57
        3.1.8. Excavator .................................................................................................................................. 62
        3.1.9. Doggy Robot ............................................................................................................................ 66
        3.1.10. Four-legged Wheel bot .......................................................................................................... 72
        3.1.11. Snake Robot ............................................................................................................................ 76
        3.1.12. Biped Robot ............................................................................................................................. 81
   3.2. INTERMEDIATE COURSE ................................................................................................................. 87
        3.2.1. Creating New Motion (method a, method b) ......................................................................... 87
        3.2.2. Creating New Motion (method c) ........................................................................................... 97
        3.2.3. Creating New Motion (Non-Standard Platform) .................................................................. 101
        3.2.4. Change the Motion Speed ...................................................................................................... 106
        3.2.5. Merging motion files .............................................................................................................. 109
        3.2.6. Using the wheel mode of wCK module .................................................................................. 113
        3.2.7. Using LED Light Function (5720T kit model only) ............................................................... 120
        3.2.8. MotionBuilder Advanced Function (1) : Frame & Transition Time .................................... 123
        3.2.9. MotionBuilder Advanced Function (2) : Torque .................................................................. 123
        3.2.10. MotionBuilder Advanced Function (3) : Control Gain ......................................................... 124
        3.2.11. Creating Action File (1) ......................................................................................................... 125
        3.2.12. Creating Action File (2) ......................................................................................................... 129
        3.2.13. Using 3 axis Acceleration Sensor (Option) (Example : test2.rba) .................................... 132
   3.3. ADVANCED COURSE ...................................................................................................................... 135
        3.3.1. What is MSRDS 2008? ........................................................................................................... 136
        3.3.2. MSRDS main feature .............................................................................................................. 136
        3.3.3. MSRDS 2008 Installation ....................................................................................................... 137
        3.3.4. MSRDS 2008 – RoboBuilder service module installation .................................................. 139
        3.3.5. MSRDS 2008 – VPL Programming ....................................................................................... 140
        3.3.6. VPL Programming – Variable declaration and usage ........................................................... 146
        3.3.7. VPL Programming – Process Variable Value ......................................................................... 151
        3.3.8. VPL Programming – Process Input Value (1) ...................................................................... 154
3.3.9. VPL Programming – Process Input Value (2) ................................................................. 157
3.3.10. VPL Programming – IF condition ............................................................................. 159
3.3.11. VPL Programming – Switch condition ....................................................................... 161
3.3.12. VPL Programming – Repetition statement ................................................................ 162

4. APPENDIX ............................................................................................................................................. 163

4.1. APPENDIX 1 - FAQ ......................................................................................................................... 163
4.2. APPENDIX 2 – HOW TO REPLACE THE DAMAGED WCK MODULE GEAR .................. 168
4.3. APPENDIX 3 - WCK MODULE COMMUNICATION PROTOCOL ........................................... 169
4.4. APPENDIX 4 – HOW TO CHANGE WCK ID ............................................................................. 170
4.5. APPENDIX 5 – 3 HUNO ARM STRUCTURES ......................................................................... 173
4.6. APPENDIX 6 – C PROGRAMMING WITH MOTION FILE ..................................................... 176
4.6.1. Material included .................................................................................................................... 176
4.6.2. Caution ..................................................................................................................................... 176
4.6.3. Work Flow ............................................................................................................................... 176
4.7. APPENDIX 7 – ROBOBUILDER REMOTE CONTROL COMMAND .................................... 180
4.8. APPENDIX 8 – IR REMOTE CONTROLLER COMMAND CODE LIST ................................ 182
4.9. APPENDIX 9 – RBC BOX EXTENSION PORT, IO MAP ......................................................... 183
4.10. APPENDIX 10 – MOTION & SOUND LIST ............................................................................. 185
1. RoboBuilder Product List

1.1. Kit Comparison

<table>
<thead>
<tr>
<th>MODEL Name</th>
<th>5710K-S03</th>
<th>5720T-S03</th>
<th>5710K-E02</th>
<th>5730K</th>
<th>5740K/T</th>
<th>DANCER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Features</strong></td>
<td>Black 16 Actuator, Huno,Dino,Dog</td>
<td>Transparent 16 Actuator Huno,Dino,Dog</td>
<td>Black 8 Actuator Small robots</td>
<td>Many robots (Huno) and Many Accessory</td>
<td>Metal Huno, Strong Motions, All Metal Gear</td>
<td>5 Dancing robot system</td>
</tr>
<tr>
<td>Pre-Assembled</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>&quot;wCK Actuators&quot;</td>
<td>12 x 1108K1</td>
<td>12 x 1108T1</td>
<td>8 x 1108K</td>
<td>16 x 1108K</td>
<td>12 x 1108K2/T2</td>
<td>5x(12 x 1108T2)</td>
</tr>
<tr>
<td></td>
<td>4 x 1111K1</td>
<td>4 x 1111T1</td>
<td>4 x 1111K1</td>
<td>4 x 1111K2/T2</td>
<td>5x(4 x 1111T2)</td>
<td></td>
</tr>
<tr>
<td>Robot Projects</td>
<td>3</td>
<td>3</td>
<td>12</td>
<td>16</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Metal Braket</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>BlueTooth Ready</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>BlueTooth Included</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Distance Sensor</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Sound Sensor</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Voice Output</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Connector Pieces</td>
<td>39</td>
<td>71</td>
<td>71</td>
<td>142</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td>Black</td>
<td>Transparent</td>
<td>Black</td>
<td>Black</td>
<td>Black / Transparent</td>
<td>Transparent</td>
</tr>
<tr>
<td>Internal LEDs</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO/YES</td>
<td>YES</td>
</tr>
<tr>
<td>Remocon Type</td>
<td>IR</td>
<td>IR</td>
<td>IR</td>
<td>IR</td>
<td>BlueTooth</td>
<td>BlueTooth</td>
</tr>
<tr>
<td>Case</td>
<td>Cardboard</td>
<td>Cardboard</td>
<td>Cardboard</td>
<td>Metal &amp; Plastic</td>
<td>Metal &amp; Plastic</td>
<td>Metal &amp; Plastic</td>
</tr>
<tr>
<td>Price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*"wCK Actuators

- All Plastic(4) Gear 1108K/T
- Metal(1)+Plastic(3) Gear 1108K1/T1
- All Metal(4) Gear 1108K2/T2

- Metal(2)+Plastic(2) Gear 1111K/T
- Metal(3)+Plastic(1) Gear 1111K1/T1
- All Metal(4) Gear 1111K2/T2
## 1.2. Kit and Part List

<table>
<thead>
<tr>
<th>Picture</th>
<th>Model No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Picture" /></td>
<td><strong>RBK-EXPERT 5730K</strong>&lt;br&gt;RoboBuilder EXPERT 5730K (BLACK)</td>
<td>- 16 Robot Projects&lt;br&gt;- 20 wCK module&lt;br&gt;- Voice Output&lt;br&gt;- BlueTooth Included&lt;br&gt;- Metal &amp; Plastic Case</td>
</tr>
<tr>
<td><img src="image2" alt="Picture" /></td>
<td><strong>RBK-CREATOR5740</strong>&lt;br&gt;Pre-assembled RoboBuilder for ROBOT Competition</td>
<td>- Pre-Assembled Version&lt;br&gt;- Metal Bracket&lt;br&gt;- No. 4 Metal Gear&lt;br&gt;- Joy Pad Remote Control Kit&lt;br&gt;- Speedy Motion&lt;br&gt;- Metal &amp; Plastic Case</td>
</tr>
<tr>
<td><img src="image3" alt="Picture" /></td>
<td><strong>RBK-CREATOR5710K</strong>&lt;br&gt;RoboBuilder Creator KIT 5710K (BLACK)</td>
<td>- 3 Robot Projects (HUNO, DOGY, DINO)&lt;br&gt;- 16 wCK module&lt;br&gt;- Sound Sensor&lt;br&gt;- Expansible Kit to 5730K&lt;br&gt;- Cardboard Case</td>
</tr>
<tr>
<td><img src="image4" alt="Picture" /></td>
<td><strong>RBK-CREATOR5720T-S02</strong>&lt;br&gt;RoboBuilder Creator KIT 5720T (TRANSPARENT)</td>
<td>- 3 Robot Projects (HUNO, DOGY, DINO)&lt;br&gt;- 16 wCK module&lt;br&gt;- Sound Sensor&lt;br&gt;- Distance Sensor&lt;br&gt;- Voice Output&lt;br&gt;- Internal LEDs&lt;br&gt;- Cardboard Case</td>
</tr>
<tr>
<td>Picture</td>
<td>Model No.</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| ![RoboBuilder Creator KIT](image) | **RBK-CREATOR5710K-E02**<br>RoboBuilder Creator KIT 5710K-E02 (BLACK) | - 12 Robot Projects  
- 8 wCK module  
- Distance Sensor  
- Sound Sensor  
- Cardboard Case |
| ![RoboBuilder Model KIT](image) | **RBK-MODEL5710K**<br>RoboBuilder Model KIT (dummy black) | - 5710K MODEL  
- Only for Displaying in Shop  
- NO movement |
| ![wCK Robot Module](image) | **wCK-1108K**<br>wCK Robot Module | - Voltage: 7.4VDC~8.4VDC  
- Torque: 8kg.cm in max  
- Output: 1.1W  
- Operating Angle  
  0°~269° (Standard Mode)  
  0°~333° (High Resolution Mode)  
- Case: Black Engineering Plastic |
| ![wCK Robot Module](image) | **wCK-1108T**<br>wCK Robot Module | - Voltage: 7.4VDC~8.4VDC  
- Torque: 8kg.cm in max  
- Output: 1.1W  
- Operating Angle  
  0°~269° (Standard Mode)  
  0°~333° (High Resolution Mode)  
- Internal LED (RED, BLUE)  
- Case: White Transparent Engineering Plastic |
<table>
<thead>
<tr>
<th>Picture</th>
<th>Model No.</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![wCK-1111K](image1) | **wCK-1111K**  
* wCK Robot Module | - Voltage: 7.4VDC~8.4VDC  
- Torque: 11kg.cm in max  
- Output: 1.1W  
- Operating Angle  
  0°~269° (Standard Mode)  
  0°~333° (High Resolution Mode)  
- Case: Black Engineering Plastic |
| ![wCK-1111T](image2) | **wCK-1111T**  
* wCK Robot Module | - Voltage: 7.4VDC~8.4VDC  
- Torque: 11kg.cm in max  
- Output: 1.1W  
- Operating Angle  
  0°~269° (Standard Mode)  
  0°~333° (High Resolution Mode)  
- Internal LED (RED, BLUE)  
- Case: White Transparent Engineering Plastic |
| ![RBC-08128NNN](image3) | **RBC-08128NNN**  
* Main Controller Box for 5710K | - wCK Module Connection Port  
- Distance/Remote Sensor Port  
- Sound Chip  
- PC port (RS-232 Serial Port)  
- Standard or Others Platform Select Button  
- Rechargeable Battery Pack  
- Power Supply Level Indication |
| ![RBC-08128YNN](image4) | **RBC-08128YNN**  
* Main Controller Box for 5720T | - wCK Module Connection Port  
- Distance/Remote Sensor Port  
- Sound Chip/ Voice Output Speaker  
- PC port (RS-232 Serial Port)  
- Standard or Others Platform Select Button  
- Rechargeable Battery Pack  
- Power Supply Level Indication |
<table>
<thead>
<tr>
<th>Picture</th>
<th>Model No.</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![RBX-IRCONT](image1) | **RBX-IRCONT**<br>IR Remote Control | - 23 Buttons Type  
  * Default 11 Button  
  * User Assign 12 Button  
  - AAA Battery 2 EA included  
  - Included in RoboBuilder Kit |
| ![RBX-PCCBL](image2) | **RBX-PCCBL**<br>PC Communication Cable | - RS-232 Serial Cable  
  - PC <= RBC Controller  
  - It is for  
    * Firmware Upgrade  
    * Robot File download |
| ![RBX-BLTOOTHCOM](image3) | **RBX-BLTOOTHCOM**<br>Bluetooth Communication Kit | - BlueTooth Module 2EA  
  - BlueTooth Interface  
    B/D 1EA  
  - S/W CD (Support Joystick) |
| ![RBX-BTCONT-01](image4) | **RBX-BTCONT-01**<br>Joypad Remote Controller Kit | - Joypad Remote  
  - Control B/D 1EA  
  - BlueTooth Module 2EA  
  - Battery Holder 1EA  
  - Screws, Bolts, Nuts |
| ![RBX-HEAD01BLK](image5) | **RBX-HEAD01BLK**<br>Head Part  
  [sensor module, black] | - Distance sensor  
  - Remote control receiver  
  - Cable connector |
| ![RBX-HEAD01TPT](image6) | **RBX-HEAD01TPT**<br>Head Part  
  [sensor module, transparent] | - Distance sensor  
  - Remote control receiver  
  - Cable connector |
<table>
<thead>
<tr>
<th>Picture</th>
<th>Model No.</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Image](image1.png) | **RBX-ACL3A01** Tri-axial acceleration sensor module | - Detect Acceleration  
- Detect Slope  
- Equip into RBC Controller |
| ![Image](image2.png) | **RBP-BATN17C** Battery Pack (Rechargeable) | - 8.4VDC, Ni-MH  
- 1.2VDC AAA type 7EA Pack |
| ![Image](image3.png) | **RBP-SMPS12V** Power Supply | - SMPS 12V 5A  
- Input : 100~240VAC, 50/60Hz  
- Output : 12VDC, 5.0A  
- CE,UL Certification |
| ![Image](image4.png) | **RBO-WCKCBL01** wCK module Cable Set [10pc] | - 4 line twist cable  
- Length : 15cm  
- Special connector |
| ![Image](image5.png) | **RBO-SCRDVR** Screw driver [+type] | - EDM Accurate screw driver (‘+’ type)  
- Thickness : 3mm  
- Length : 100mm |
| ![Image](image6.png) | **RBO-CHEST01BLK** Chest Part for HUNO (black) | - Chest cover for HUNO (5710K)  
- Simple assembling (3 screw only)  
- RoboBuilder logo sticker |
<table>
<thead>
<tr>
<th>Picture</th>
<th>Model No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="RBO-CHEST01TPT" /></td>
<td><strong>RBO-CHEST01TPT</strong>&lt;br&gt;Chest Part for HUNO (blue)</td>
<td>- Chest cover for HUNO (5720T)&lt;br&gt;- Simple assembling (3 screw only)&lt;br&gt;- RoboBuilder logo sticker</td>
</tr>
<tr>
<td><img src="image2" alt="RBO-CHEST02TPT" /></td>
<td><strong>RBO-CHEST02TPT</strong>&lt;br&gt;Chest Part for HUNO [transparent for tuning]</td>
<td>- Chest cover for HUNO (Transparent for tuning)&lt;br&gt;- Simple assembling (3 screw only)&lt;br&gt;- RoboBuilder logo sticker</td>
</tr>
<tr>
<td><img src="image3" alt="RBO-MTBKT01SLR" /></td>
<td><strong>RBO-MTBKT01SLR</strong>&lt;br&gt;Metal Bracket Silver</td>
<td>- Knee Bracket 2EA&lt;br&gt;- Leg Bracket 2EA&lt;br&gt;- Joint02 Bracket 4EA&lt;br&gt;- Joint12 6EA&lt;br&gt;- Screws</td>
</tr>
<tr>
<td><img src="image4" alt="RBO-WCKGEAR03" /></td>
<td><strong>RBO-WCKGEAR03</strong>&lt;br&gt;Gear 1 for wCK-1111</td>
<td>- wCK-1111 Series Gear&lt;br&gt;- No. 1 Gear 1EA&lt;br&gt;- Black Plastic</td>
</tr>
<tr>
<td><img src="image5" alt="RBO-MFRAME01BLK" /></td>
<td><strong>RBO-MFRAME01BLK</strong>&lt;br&gt;Body Frame (black)</td>
<td>- Body Frame (5710K)</td>
</tr>
<tr>
<td><img src="image6" alt="RBO-MFRAME01TPT" /></td>
<td><strong>RBO-MFRAME01TPT</strong>&lt;br&gt;Body Frame (transparent)</td>
<td>- Body Frame (5720T)</td>
</tr>
<tr>
<td>Picture</td>
<td>Model No.</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------</td>
<td>--------------------------------------</td>
</tr>
</tbody>
</table>
| ![Image](image1) | **RBO-BPARTS01BLK**  
Body Parts (black) | - Hand 2EA  
- Knee 2EA  
- Leg 2EA |
| ![Image](image2) | **RBO-BPARTS01TPT**  
Body Parts (transparent) | - Hand 2EA  
- Knee 2EA  
- Leg 2EA |
| ![Image](image3) | **RBO-JOINT01BLK**  
Joint Set 01 (11 types) | - wCK module  
Joint Set Type 1 |
| ![Image](image4) | **RBO-JOINT02BLK**  
Joint Set 02  
(8 types for tuning) | - wCK module  
Joint Set Type 2 |
| ![Image](image5) | **RBO-WCKGEAR01**  
Gear 1,2,3 for wCK-1108 | - wCK-1108 Series Gear  
- No. 1,2,3 Gear each 1EA  
- Black Plastic |
| ![Image](image6) | **RBO-WCKGEAR02**  
Gear 4 for wCK-1108 | - wCK-1108 Series Gear  
- No. 4 Gear 1EA  
(Included metal bearing)  
- Black Plastic |
<table>
<thead>
<tr>
<th>Picture</th>
<th>Model No.</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Image](image1.png) | **RBO-WCKGEAR04**  
Gear 2,3 for wCK-1111 | - wCK-1111 Series Gear  
- No. 2, 3 Gear each 1EA  
- Metal Gear |
| ![Image](image2.png) | **RBO-WCKGEAR05**  
Gear 4 for wCK-1111 | - wCK-1111 Series Gear  
- No. 4 Gear 1EA  
(Included metal bearing)  
- Black Plastic |
| ![Image](image3.png) | **RBO-WCKGEAR-06**  
1108 type Metal Gear  
(No. 4 Gear) | - wCK-1108 Series Gear  
- No. 4 Metal Gear 1EA |
| ![Image](image4.png) | **RBO-WCKGEAR-07**  
1111 type Metal Gear  
(No. 4 Gear) | - wCK-1111 Series Gear  
- No. 4 Metal Gear 1EA |
2. Software List

2.1. RoboBuilder Download Tool

Download Tool is used for downloading a number of Motion files (*.rbm) or Action files (*.rba) at one time. Users can even designate the download positions in remote controller.

2.1.1. Main Features

① Download by Bulk Function
② Download List Edit Function
  1) Directory Search, File Register
  2) File Drag and Drop
  3) File Downloading Sequence Changeable
  4) File Delete in Downloading List
  5) Keyboard Support
③ RoboBuilder Website Link
④ COM Port Access
⑤ File Information Display
⑥ Available Memory Space Display

2.1.2. Display Structure

① PC Port Connection Part
  . COM Port : Designate available COM port for connection of RoboBuilder.
  . BaudRate : It shows data transferring speed.
  . OpenPort : Open PC COM port in order to connect a RoboBuilder.
  . ScanPort : Find available COM port and opens it automatically.

② Logo and website address Part
  . If you click RoboBuilder Logo, it shows RoboBuilder website.
  . Data : It shows Robot File Sharing section webpage.
3 File Type Part
   - Motion File (*.rbm) : If selected, motion file download job is possible.
   - Action File (*.rba) : If selected, action file download job is possible.

4 Directory Part
   - It shows present directory path. You can change the directory path by mouse.

5 File Part
   - It shows present data files. By clicking the left button of mouse, the file is selected.
   - '>' Button : Register a motion/action file into the "Download File List".
   - '>>' Button : Register motion/action files all file (20 files in max.) into "Download File List".

6 RBC Information Part
   - RBC Firmware Version : It shows present RoboBuilder Control Box (RBC) firmware version.

7 Download File List Part
   - It shows download file list. The background color will be changed according to RoboBuilder platform type. And this color is similar with RBC platform LED.

Blue : Creator HUNO, Pink : Creator DINO, Red : Creator DOGY, Orange : Creator Others

8 Memory Space Part
   - It shows available memory space in RBC memory after download files are registered. “E” means, available space and, the number is registered file number. The background of chart color is similar with RoboBuilder platform color indication.

9 File List Editing and Downloading Part
   - Up Button : It moves the file to previous number position.
   - Down Button : It move the file to next number position.
   - Delete Button : It deletes selected file in the download file list.
   - Delete All Button : It deletes all files in the download file list.

10 File Information Display Part
   - It shows selected file information in detail.
   - If motion file is selected, it shows file name, file size, robot platform, scene number, performance time information. If action file is selected, it shows file name, file size, robot platform, statements information.
2.1.3. How To Download

In this example, it shows how to use this Download program in detail.

※ For action file downloading, you select action file (*.rba), then the rest procedures are same as motion files downloading.

① Run RoboBuilder Download Tool. If you downloaded motion file previously, it shows motion files in the Download File List.

② If you press ‘Data’ Button, Internet Explorer runs, and its shows Robot File Sharing section page. Then please click “LOGIN” Button.
③ Input your ID and Password.

④ Click “Download”.

⑤ Click “Save” button when it shows file download window.
6. Then, save the file for downloading.

7. Directory path is changed automatically in RoboBuilder Download Tool, and it shows downloaded file from the website. Click “Delete All” in order to delete the previous downloaded files in the list.

8. In order to register onto Download File List, select the file then, click ‘>’ button.

* Users can use Drag & Drop function.
If you click a file in the Download File List, it shows file name, file size, robot platform, scene number, performance time.

※ If registered files are more than one in the ‘Download File List’, users can change the downloading sequence by clicking “Up” or “Down” button.
※ Users can do this function by using Keyboard.
  ’Up’ button = ‘+’ Key
  ’Down’ button = ‘−’ Key
  ’Delete’ button = ’Delete’ Key

Plug the adapter into RoboBuilder, then connect RoboBuilder with PC through PC download cable. Power on RBC Box.

Click ‘ScanPort’ button in order to find available COM Port.
⑫ RBC serial code and Firmware Version will be shown if connected properly. And "ClosePort" button will be shown as well.

⑬ Click 'Download" button, in order to all files in the "Download File List". Then it starts downloading into RBC Box. Following message box will shown after downloaded completely.
2.2. Motion Builder

Motion Builder is the software for creating the RoboBuilder’s robot motions.

2.2.1. Project, Motion File, Scene, Frame, Transition Time

1) **Project file (*.prj)**
   - A project file contains the information such as the robot’s type and more. It is used to manage multiple motion files of a robot efficiently. Therefore, one project file includes one or more motion files.

2) **Motion file (*.rbm)**
   - A motion file contains the complete data to execute its movement. One motion file consists of multiple scenes.

3) **Scene**
   - A scene is a smaller motion unit that constitutes a complete motion file. A scene consists of start position and destination position. Except the first scene, the start position of a scene is the destination position of its previous scene. When a scene is executed, the frame data is generated automatically according to the predefined number of frames and delivered to each actuator modules.

4) **Frame**
   - A frame is the smallest motion unit that constitutes a scene. Each frame can be considered as the still image that is actually sent to robot actuators. The more frames you define, the smoother the motion becomes. One scene can have from 1 up to 100 frames.

5) **Transition Time**
   - Transition time is the time duration that is taken to execute a scene. Transition time is closely related with the number of frames. It can have value from 20msec up to 6000msec. The minimum transition time that can be allocated for a frame is 20msec.
     
     e.g) If scene A has 10 frames, the transition time can be selected from 200 up to 6000.
2.2.2. Screen Layout

This is the screen layout of the MotionBuilder.
<table>
<thead>
<tr>
<th>No.</th>
<th>Area Name</th>
<th>Functions &amp; Descriptions</th>
</tr>
</thead>
</table>
| ①  | Menu Bar                     | • New: creates a new project by defining project name, file path, robot type etc.  
• Open: opens an existing project file. (*pp)  
• Save All: stores the running project file and all data related to the project.  
• Save As: saves the running project file as a different name.  
• Config: configures and sets the wCK module.  
• Download: transfers robot files to control box. |
| ②  | PC Port connection           | • ComPort: sets the port on PC to connect RoboBuilder with  
• BaudRels: sets the data communication speed. (default: 115,200kbps)  
• OpenPort: opens the set PC port to connect RoboBuilder with. |
| ③  | Robot Configuration          | • This area illustrates the mechanical construction of the wCK modules. Using the jog dial pad, you can control the movement of each wCK module.  
* If the [Default] button is not checked, you can freely relocate the jog dial pads of the wCK modules by dragging them with your mouse/right-click.  
When [Default] is selected, they return to their original default position. |
| ④  | Motion File Information      | • Motion Name: displays the name of the motion file running.  
• Total Scene: displays the total number of scenes that constitutes the motion file running.  
• Scene Index: displays the number of the selected scene in the running motion file.  
• Repeat: is used to repeat and test the selected one or more scenes. |
| ⑤  | Motion File Management       | • New Motion: creates a new motion file.  
• Motion List: add, open, modify, or remove motion files.  
• Save As: saves the running motion file as a different name.  
• Save All: saves the running motion file. |
| ⑥  | Position Control             | • Restore: sets all modules' displacement angles of the selected scene to "0".  
• Get Pos: captures the desired posture of a robot after adjusting the posture manually with user's hands. Captured posture is saved as in a scene.  
• Init Pos: sets the initial torque and angle of the selected wCK module. |
| ⑦  | Scene Management             | • Set Motion: sets the name and saved path of the motion file, configures PID gains of wCK modules.  
• Delete: deletes the selected scene.  
• Test: run the selected scene. (multiple scene selection available)  
• New Scene: adds a new scene.  
• Paste: pastes the copied scene in the selected position.  
• Select All: selects all scenes in a motion file. |
| ⑧  | wCK module Control Detail    | • ID: displays the ID number of the wCK module.  
• S.Pos: stands for Start Position and it displays the start position of the wCK module in unit of control angle.  
• D.Pos: stands for Destination Position and it displays the destination position of the wCK module in unit of control angle.  
• Displ: stands for Displacement and it displays the control angle difference between S.Pos and D.Pos.  
• Torq: it displays the speed of the wCK module. (0: Very fast, ~4: Very slow)  
• Port: displays the status of the LED installed on the I/O port of the wCK module. |
| ⑨  | Scene Editing                | • Scene Name: displays the scene name.  
• Frames: displays the number of frames, into which a scene is divided.  
• TransTime[ms]: displays the transition time that is used for operating the corresponding scene. |
| ⑩  | Task Info                    | • displays the task related information such as the PC port connected, communication speed, number of wCK modules connected, robot type, etc. |

### 2.2.3. Easy Programming style

Motion can be programmed by "[Get Pos]" menu in the position control part of MotionBuilder.

This programming is the way that user makes the robot’s posture himself, then capture this posture. Captured posture is saved as a last posture of next scene.

* Please refer to the detailed motion programming in chapter 3.2
2.3. Action Builder

“Action” is the robot’s action that has certain purpose, and Action Builder is the software that can create, edit, save and download the Action files.

2.3.1. Screen Layout

1. **Menu Bar**
   - New: creates a new action file.
   - Open: opens an existing action file (*.rba).
   - Save: saves the running action file.
   - Save As: saves the running action file as a different name.
   - Config: configures the file properties.
   - Download: transfers an action file to the control box.

2. **PC Port Connection**
   - ComPort: sets the port on PC to connect the robot with.
   - BaudRate: sets the data communication speed (default: 115,200kbps).
   - OpenPort: opens the set PC port to connect the robot with.
   - ScanPort: scans, finds, and opens the PC port connected with the robot automatically.

3. **Action File Information**
   - Action Name: displays the name of the action file currently running.
   - Robot Platform: displays the robot platform type on which the action file will be played.
   - Total Statements: displays the number of statements in the action file.

4. **Statement List**
   - Index: displays the index number of the statement.
   - St. Name: displays the name of the statement.
   - Condition: displays the condition part of the statement.
   - Execution: displays the execution part of the statement.
   - Description: displays the description of the statement.

5. **Statement Editing**
   - Statement Name: displays the statement name. It is also used to type in the name for the statement.
   - Description: displays the description of the statement. It is also used to type in the description for the statement.
   - Delete: delete the selected statement from the action file.
   - Update: update the change to the statement.
6 Conditions [if]
   - None: no condition
   - Distance: condition of visual distance detection
   - Sound In: condition of sound detection
   - Button: condition of the buttons on the control box
   - Remocon: condition of inputs from remote controller
     [IR remote control, compatible joystick etc]
   - Accel.: X, Y, Z axis acceleration

7 Executions [then]
   - None: no execution
   - Motion Out: play the selected motion file.
   - Sound Out: play the selected sound source.
   - Wait Time: wait for the specified amount of time [in millisecond].
   - Jump Index: jumps to the statement of specified index number.

※ Please refer to the Action file programming by using Action Builder in chapter.

2.3.2. Downloading Action File (Example: test.rba)

1) Connect the power supply to the control box, and use the PC cable to connect the robot to the PC. Turn on the power switch on the control box.

2) Open the ActionBuilder software. Click [ScanPort] and then it automatically scan and open the com port connected with the robot.
   ※ Caution: On a PC using a device such as Bluetooth Dongle that involves many virtual ports, the [ScanPort] button may not work properly. In this case, please choose the [ComPort] manually and then click [OpenPort].)
3) Click [Download] in the menu bar, and select the action file to transfer to the control box. Then Click [Open].

4) The selected action file is transferred to the control box, and the [Download Successful] window pops up. Then click [OK] to finish the transfer.

5) In order to play the transferred action file, first click [ClosePort] to disconnect the robot from comport. Use the remote controller to press and hold the button # and then press the numeric button (1~0) together. For example, the first action file is played when you press # button and 1 button together. If you want to stop the action file, turn off the power.

*NOTE*
1. Once file transfer begins, all action files already existing in the control box are deleted and the new files overwrite from the beginning of the ROM memory. The first file transferred is assigned to the button 1 of the remote controller, the second file transferred is assigned to button 2, and the third file to button 3 and so on. Therefore, you have to plan and decide which action file to assign to which button before you actually start transferring the files.
2. When you transfer action files, the motion files are not affected.
3. If the size of a particular action file is too large, the RBC is NOT able to save up to 10 action files.
4. The control box firmware has to be upgraded up ver. 2.0 or above in order to use the Action Builder.
2.4. RoboBuilder Diagnostic Tool

RoboBuilder Diagnostic Tool is to examine and diagnostic whether all the standard platform robots (HUNO, DINO, DOGY) works properly. This software tool advises (wCK) module connected not properly, or damaged wCK module, if there is any.

2.4.1. Diagnostic Tool main feature

① wCK communication verification function
② wCK assembly position verification function

2.4.2. Diagnostic Tool Layout

① PC Port connection
   - COM Port: Designate available PC port.
   - Baud rate: Display Data transmission speed.
   - Open/Close: Open or Close PC Port.
   - Scan Port: Search available PC port automatically, and Open PC Port.

② Test Part
   - STEP 1 Button: Communicate with all 16 wCKs, and display the test result in the text box.
   - STEP 2 Button: Test all 16 wCKs’ position control, LED control so that user check status.
   - STOP: STOP STEP 2 operation.

③ Message text box Part
   - Display job progress and test results.

④ Image display Part
   - Display COM Port status and Robot platform information.
2.5. RBC Firmware Upgrade Tool

This software is to upgrade or change RBC firmware version.

※ This software does not compatible in Windows Vista.
   Please use this software in Windows XP.

2.5.1. RBC Firmware Upgrade Tool Layout

1) Connection : Select proper COM port and Baud rate.
2) File Selection : Choose the firmware file that should be upgraded.
   Click “Click here and Push Reset Button”, then it starts firmware upgrade.
   For Reset Button, please press the button between PF1 and PF2 in RBC.
3) Status : Display upgrade status.
4) End : End upgrade program.
2.5.2. How to upgrade Firmware

1) Run upgrade software (RBC Upgrade Tool), turn on the power of RBC.

2) Connect RBC with PC with RS232 cable and set Com Port and set Baud Rate of 115200bps.

3) Click folder icon to select firmware file (*.hex) to upgrade and click open.

4) Press "Click here and Push Button" and it turns to stand-by mode.

5) Use a sharp pen to push the reset button of RBC (the hole between PF1 button and PF2 button).

6) Proceed the upgrade.

7) When finished RBC, it reset automatically and complete message appears. then, Click OK.

8) If fails, try again from step1.
3. RoboBuilder Education Course

3.1. Beginner course

In this course, it describes 12 examples of robots using 1 to 8 wCK robot modules, various frames and joints. In this course, users can learn robot’s principle and knowledge as they can make various robot type, motions and actions. This helps to increase robot beginners’ creativity as well.

3.1.1. Crossing Gate

1) Goal of the Chapter

In this chapter, users will build a simple one-DOF robotic system and learn basic principles of robotics control. Use one wCK module and the distance sensor to build an automatic crossing gate. If an object is detected approaching, it lifts up the crossbar. It lowers down the crossbar if the object passes by.

2) Robot System

What is a ‘robot system”? Although the robotics terminology is difficult, let’s continue to learn about it by building different robots one by one.

First, what is a ‘robot”? A robot is a mechanical structure that has similar appearances and functions of a human, or a mechanical system that carries out a task autonomously. Robots are not just simple machines but they are capable of doing work autonomously. This “autonomy” of robots helps to replace human labor and carry out a lot of tasks for humans. They can replace human beings by doing arduous and repeated work, by working in hazardous environments, as well as by providing entertainment for humans.

A ‘system’ means a set of elements that work together by generate an output signal in response to an input signal. In other words, a system has a set of required elements, and all elements involved have clearly
defined interrelation. A robot is one example of the system.
A robot system is a mechanical system that incorporates a set of elements such as actuators and sensors that generate an output (e.g. motion) in response to external environment (input). It has similar appearances of a human, or it has the capability to carry out tasks for human autonomously.

3) Sensor
A device that does the function of human organs such as eyes, nose, mouth, ears, and skin is called a sensor. There are a distance sensor and a sound sensor that we will learn in this study book. A distance sensor simulates one of the functions of the eyes. The distance sensor sends out invisible infrared light and a little of the infrared light is reflected by an object. The distance sensor measures the angle of the returning light to decide the distance from the sensor to the object. The distance sensor RBX-HEAD01BLK can detect an object within the distance ranging from 10cm to 50cm. The sound sensor simulates one of the functions of the ears and detects the height of a sound.

4) Robot Building
Follow the building instructions shown below step by step and assemble the automatic crossing gate. Be cautious about the position of the wCK actuator module and the distance sensor. Please pay good attention to STEP02 where the connecting angle of Joint9 with the pentagon-shaped junction is very important.
Part List

- Body Frame x 1
- Control Box x 1
- Sensor Module x 1
- wCK module x 1
- wCK Cable x 1
- Foot Part x 2
- Joint 9 x 1
- Nut x 7
- B8 x 7
- B40 x 4

STEP 01

Sensor

Bolt M2x8L : 2EA
**STEP 02**

**Actuator**

- Bolt M2x8L : 1EA
- Nut M2 : 4EA
- Bolt M2x40L : 4EA

*Caution*

The adjustment of the connecting angle is critically important.

**STEP 03**

**Crossbar**

- Bolt M2x8L : 4EA
- Nut M2 : 3EA
5) Running example robot files [motion file, action file]

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_Opening.rbm</td>
<td>Open the crossing gate</td>
</tr>
<tr>
<td></td>
<td>02_Closing.rbm</td>
<td>Close the crossing gate</td>
</tr>
<tr>
<td>Action File</td>
<td>01_Manual_OC.rba</td>
<td>Use remote control to open and close the gate</td>
</tr>
<tr>
<td></td>
<td>02_Auto_OC.rba</td>
<td>Autonomous control of the gate</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

6) Application

1) Create a few different motions for various opening angles of the crossbar. Incorporate these motions to create an action file that opens the crossbar with different opening angles according to the distance from an object approaching.
2) Create a motion that lifts up the crossbar and close it down. Incorporate this motion to create an action file that runs this motion in response to a handclap. Set the sound level as “15<Sound Level”.
3.1.2. Automatic Tongs

1) Goal of the Chapter
In this chapter, users will build a simple two-DOF robotic system and learn basic principles of robotics control. Use two wCK modules and the distance sensor to build a pair of automatic tongs. If an object is detected approaching, the tongs will hold the object automatically. They open the arms if the object moves away.

2) Controller
Just like the brain of a human controls his body, the CONTROLLER of a robot controls the motion and action of a robotic system. The robot controllers are usually computers of various types. There are three types of controllers that can be used for RoboBuilder.
First, the RBC (RoboBuilder Controller) control box is dedicated to controlling robot systems made of RoboBuilder’s products such as wCK modules. The RBC control box has a micro processor ATMEGA128 built in that provides the function of running robot files users created.
Second, PC (Personal Computer) is widely used to control robot systems. Connect the control box to user’s PC and set the buttons so that both PF1 LED (blue) and PF2 LED (orange) turn on together. Then the control box transfers all commands coming from PC directly to the robot.
Lastly, other types of controllers can also be used to control RoboBuilder robots if they support UART standard and RoboBuilder’s communication protocol. For example, various electronic devices such as a PDA, mobile phone, even the hand-held game Nintendo DS, can be used to control RoboBuilder robots as long as they have compatible program in them.

3) Robot Building
Follow the building instructions shown below step by step and assemble the automatic tongs. Be cautious about the position of the wCK actuator module and the distance sensor. Please pay good attention to STEP01 about the connecting angles of the Joint6 with the axis-junction of the wCK actuator modules.
ID MAP

Top

Part List

- Body Frame x 1
- Control Box x 1
- Sensor Module x 1
- wCK module x 2
- wCK Cable x 2
- Hand Part x 2
- Joint 6 x 2
- Nut x 8
- B6 x 4
- B12 x 4
- B16 x 4
- B40 x 4
STEP 01

**Tongs**
- Bolt M2x12L : 2EA
- Nut M2 : 8EA
- Bolt M2x16L : 4EA
- Bolt M2x40L : 4EA

*Caution*
The adjustment of the connecting angle is critically important.

STEP 02

**Sensor**
- Bolt M2x6L : 4EA
- Nut M2 : 4EA
- Bolt M2x12L : 2EA
4) Running example robot files [motion file, action file]

< List of example files >

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_open.rbm</td>
<td>Open the arms</td>
</tr>
<tr>
<td></td>
<td>02_close.rbm</td>
<td>Close the arms</td>
</tr>
<tr>
<td></td>
<td>03_handclap.rbm</td>
<td>Hand clap</td>
</tr>
<tr>
<td>Action File</td>
<td>01_control_by_button.rba</td>
<td>Use control box buttons to open and close the arms</td>
</tr>
<tr>
<td></td>
<td>02_copy_clap.rba</td>
<td>Follow handclap</td>
</tr>
<tr>
<td></td>
<td>03_autograb.rba</td>
<td>Autonomous control of the tongs</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

5) Application

1) Create a few different motions for various opening angles of the tongs. Incorporate these motions to create an action file that opens the tongs with different opening angles according to the distance from an object approaching.
3.1.3. Tricycle

1) Goal of the Chapter
In this chapter, users will build a three-DOF wheel robot system and learn how to program and control it. Use three wCK modules and the distance sensor to build the tricycle robot. The tricycle robot can move ahead avoiding objects in front by turning to the right or to the left.

2) Actuator
A robot actuator does the function of human muscle and provides required power so that a robot can move its mechanical body. An actuator usually consists of an electric motor and mechanical gears. In this study book, we use the robotic actuator called wCK-1108. Enclosed inside the wCK module are a sensor that reads the angular position of the rotation axis, an electric DC motor that generates power, a few mechanical gears that transmit power to the rotation axis, as well as the micro controller. The wCK actuator module itself is a good example of a robotic system. A motor system such as wCK module with control of position and speed is widely called a servo motor system.

3) Robot Building
Follow the building instructions shown below step by step and assemble the tricycle robot. Be cautious about the position of the wCK actuator modules and the distance sensor.
ID MAP

Top

Bottom

Part List

Body Frame x 1
Control Box x 1
Sensor Module x 1
wCK module x 3
Joint 5 x 3

wCK Cable x 3

Nut x 12
B6 x 4
B12 x 2

B8 x 3
STEP 01

Sensor, Actuator

<table>
<thead>
<tr>
<th>Bolt M2x6L : 4EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt M2x8L : 3EA</td>
</tr>
<tr>
<td>Bolt M2x12L : 2EA</td>
</tr>
<tr>
<td>Bolt M2x40L : 8EA</td>
</tr>
<tr>
<td>Nut M2 : 12EA</td>
</tr>
</tbody>
</table>

Use glue to attach the tire.

STEP 02

RBC

※ push when disconnect RBC
4) Running example robot files [motion file, action file]

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_con_move_f.rbm</td>
<td>Continuously move forward</td>
</tr>
<tr>
<td></td>
<td>02_step_move_f.rbm</td>
<td>Move forward for a short distance</td>
</tr>
<tr>
<td></td>
<td>03_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>04_step_turn_l.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>05_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>06_step_turn_r.rbm</td>
<td>Turn to the right</td>
</tr>
<tr>
<td></td>
<td>07_con_move_b.rbm</td>
<td>Continuously move backward</td>
</tr>
<tr>
<td></td>
<td>08_step_move_b.rbm</td>
<td>Move backward for a short distance</td>
</tr>
<tr>
<td>Action File</td>
<td>01_const_dist.rba</td>
<td>Keep a short distance from an object ahead</td>
</tr>
<tr>
<td></td>
<td>02_move&amp;avoid.rba</td>
<td>Move forward and avoid the object ahead by turning to the right</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

5) Application
1) Create an action file that makes the tricycle to move forward in response to handclap and stop in detection of an object ahead.
2) Add one more wCK module and build a modified tricycle that can steer the front wheel.

6) Game
1) Hold a straight course running competition.
2) Arrange a floor filled with some obstacles and hold a running competition.
3.1.4. Robotic Arm

1) Goal of the Chapter
In this chapter, users will build a five-DOF robotic arm and learn how to control multi-axis robotic arm. Use five wCK modules and the distance sensor to build the automatic robotic arm. The robotic arm can pick up and place down things flexibly.

2) Robot Building
Follow the building instructions shown below step by step and assemble the robotic arm. Be cautious about the position of the wCK actuator module and the distance sensor. Please pay good attention to STEP02 where the connecting angle of Joint12 with the pentagon-shaped junction is very important.
ID MAP

Part List

Body Frame x 1  Control Box x 1  Sensor Module x 1  wCK module x 5  Leg Frame x 5  Foot Part x 3

Hand Part x 2  wCK Cable x 5  Joint 2 x 1  Joint 5 x 4  Joint 7 x 2  Joint 9 x 4

Joint 12 x 1  Nut x 22  B8 x 10  B12 x 3  B16 x 18  B40 x 8
STEP 01

**Sensor, Legs**

- Bolt M2x8L : 5EA
- Nut M2 : 8EA
- Bolt M2x16L : 2EA
- Bolt M2x40L : 8EA

※ same leg for the other side

STEP 02

**Arm**

- Bolt M2x8L : 1EA
- Nut M2 : 4EA
- Bolt M2x16L : 4EA

Caution: The adjustment of the connecting angle is critically important.
STEP 03

**Tongs**

<table>
<thead>
<tr>
<th>Bolt M2x6L</th>
<th>4EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nut M2</td>
<td>10EA</td>
</tr>
<tr>
<td>Bolt M2x12L</td>
<td>2EA</td>
</tr>
<tr>
<td>Bolt M2x16L</td>
<td>8EA</td>
</tr>
</tbody>
</table>

※ same for the other side
3) **Running example robot files [motion file, action file]**

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_open.rbm</td>
<td>Open the tongs</td>
</tr>
<tr>
<td>Motion File</td>
<td>02_clap.rbm</td>
<td>Handclap</td>
</tr>
<tr>
<td>Motion File</td>
<td>03_close.rbm</td>
<td>Close the tongs</td>
</tr>
<tr>
<td>Motion File</td>
<td>04_left_down.rbm</td>
<td>Move the arm to the left</td>
</tr>
<tr>
<td>Motion File</td>
<td>05_basic_pose.rbm</td>
<td>Basic Posture</td>
</tr>
<tr>
<td>Motion File</td>
<td>06_right_down.rbm</td>
<td>Move the arm to the right</td>
</tr>
<tr>
<td>Action File</td>
<td>01_copy_clap.rba</td>
<td>Follow the handclap</td>
</tr>
</tbody>
</table>

1) Use the MotioBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

4) **Application**

1) Create a few different motions for various opening angles of the tongs. Incorporate these motions to create an action file that opens the tongs with different opening angles according to the distance from an object approaching.
2) Have the tongs hold a fan and create a fanning motion. Incorporate this motion to create an action file that does the fanning motion when it detects an object approaching.

5) **Game**

1) Put a basket to the left filled with candies and put an empty basket to the right. Hold a competition where you win if you move more candies to the empty basket using the remote controller.
3.1.5. Tri-pedal Robot

1) Goal of the Chapter
In this chapter, users will build a six-DOF tri-pedal robot and learn how to control its movement. Use six wCK modules and the distance sensor to build the tri-pedal creature. The tri-pedal robot can move forward and turn in various directions.

2) Robot Building
Follow the building instructions shown below step by step and assemble the tri-pedal robot. Be cautious about the position of the wCK actuator module and the distance sensor. Please pay good attention to STEP02, STEP03, and STEP04 where the connecting angle of joints with the wCK modules is very important.
STEP 01

Sensor
- Bolt M2x6L : 3EA
- Nut M2 : 4EA
- Bolt M2x12L : 2EA

STEP 02

Legs
- Bolt M2x8L : 1EA
- Nut M2 : 4EA
- Bolt M2x16L : 4EA

The adjustment of the connecting angle is critically important.

Use glue to attach the tire.
STEP 03

Legs

- Bolt M2x8L: 1EA
- Nut M2: 10EA
- Bolt M2x12L: 1EA
- Bolt M2x16L: 2EA
- Bolt M2x40L: 6EA

*Caution* The adjustment of the connecting angle is critically important.

STEP 04

Legs

- Bolt M2x8L: 2EA
- Nut M2: 7EA
- Bolt M2x16L: 3EA
- Bolt M2x40L: 4EA

*Caution* The adjustment of the connecting angle is critically important.

Use glue to attach the tire
3) Running example robot files [motion file, action file]

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>02_step_move_12h.rbm</td>
<td>Move in the direction of 12 o’clock</td>
</tr>
<tr>
<td></td>
<td>03_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>04_step_turn_ccw.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>05_basic_pose.rbm</td>
<td>Basic posture</td>
</tr>
<tr>
<td></td>
<td>06_step_turn_cw.rbm</td>
<td>Turn to the right</td>
</tr>
<tr>
<td></td>
<td>07_step_move_04h.rbm</td>
<td>Move in the direction of 4 o’clock</td>
</tr>
<tr>
<td></td>
<td>08_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>09_step_move_08h.rbm</td>
<td>Move in the direction of 8 o’clock</td>
</tr>
<tr>
<td>Action File</td>
<td>01_move&amp;avoid.rba</td>
<td>Move forward and avoid obstacles by turning to the right</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

4) Application
1) Modify the inclinations of the three wheels and see how the motions change. Discuss what affected this result.

5) Game
1) Hold a straight course running competition.
2) Arrange a floor filled with some obstacles and hold a running competition.
3.1.6. Four-wheeled Vehicle

1) Goal of the Chapter
In this chapter, users will build a four-DOF vehicle robot and learn how to control four-wheeled automobile. Use four wCK modules and the distance sensor to build the vehicle robot. The vehicle can freely navigate its way by avoiding obstacles in front.

2) Robot Building
Follow the building instructions shown below step by step and assemble the vehicle robot. Be cautious about the position of the wCK actuator module and the distance sensor.
ID MAP

Part List

Body Frame x 1
Control Box x 1
Sensor Module x 1
wCK module x 4
wCK Cable x 4
Joint 5 x 4
Nut x 18
B6 x 4
B8 x 4
B12 x 2
B40 x 4
STEP 01

**Sensor**

- Bolt M2x6L : 4EA
- Nut M2 : 4EA
- Bolt M2x12L : 2EA

STEP 02

**Leads**

- Nut M2 : 14EA
- Bolt M2x40L :
STEP 03

Wheels

Bolt M2x8L : 4EA

Use glue to attach the tire
3) Running example robot files [motion file, action file]

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_con_move_f.rbm</td>
<td>Continuously move forward</td>
</tr>
<tr>
<td></td>
<td>02_step_move_f.rbm</td>
<td>Move forward for a short distance</td>
</tr>
<tr>
<td></td>
<td>03_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>04_step_turn_l.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>05_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>06_step_turn_r.rbm</td>
<td>Turn to the right</td>
</tr>
<tr>
<td></td>
<td>07_con_move_b.rbm</td>
<td>Continuously move backward</td>
</tr>
<tr>
<td></td>
<td>08_step_move_b.rbm</td>
<td>Move backward for a short distance</td>
</tr>
<tr>
<td>Action File</td>
<td>01_move&amp;avoid.rba</td>
<td>Move forward and avoid obstacles by turning to the left</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

4) Application
   1) Discuss the difference of this four-wheeled vehicle robot from the tri-pedal robot we studied in Chapter 2-5.

5) Game
   1) Hold a straight course running competition.
   2) Arrange a floor filled with some obstacles and hold a running competition.
3.1.7. Tongs Car

1) Goal of the Chapter
In this chapter, users will build a four-wheeled vehicle with automatic tongs installed on it. Use eight wCK modules and the distance sensor to build the tongs car. The car can freely navigate its way by avoiding obstacles in front as well as hold various objects.

2) Robot Building
The four-wheeled vehicle can move its tongs vertically as well as horizontally just like human arms. Follow the building instructions shown below step by step and assemble the vehicle robot. Be cautious about the position of the wCK actuator module and the distance sensor.
ID MAP

Top

Bottom

Part List

Body Frame x 1
Control Box x 1
Sensor Module x 1
Foot Part x 2
wCK module x 8
wCK Cable x 8
Joint 5 x 4
Joint 10 x 2
Nut x 26
B6 x 4
B8 x 4
B12 x 6
B40 x 22
STEP 01

Sensor

<table>
<thead>
<tr>
<th>Component</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt M2x6L</td>
<td>4EA</td>
</tr>
<tr>
<td>Nut M2</td>
<td>4EA</td>
</tr>
<tr>
<td>Bolt M2x12L</td>
<td>2EA</td>
</tr>
</tbody>
</table>

STEP 02

Left Tong

<table>
<thead>
<tr>
<th>Component</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt M2x12L</td>
<td>2EA</td>
</tr>
<tr>
<td>Nut M2</td>
<td>4EA</td>
</tr>
<tr>
<td>Bolt M2x40L</td>
<td>4EA</td>
</tr>
</tbody>
</table>
STEP 03

**Right Tong**
- Bolt M2x12L : 2EA
- Nut M2 : 4EA
- Bolt M2x40L : 4EA

STEP 04

**Wheels**
- Bolt M2x8L : 4EA
- Nut M2 : 14EA
- Bolt M2x40L :

Use glue to attach the tire.
3) Running example robot files [motion file, action file]

< List of example files >

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_put_down.rbm</td>
<td>Put down an object</td>
</tr>
<tr>
<td></td>
<td>02_con_move_f.rbm</td>
<td>Continuously move forward</td>
</tr>
<tr>
<td></td>
<td>03_lift_up.rbm</td>
<td>List up an object</td>
</tr>
<tr>
<td></td>
<td>04_step_turn_l.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>05_stop.rbm</td>
<td>Basic posture</td>
</tr>
<tr>
<td></td>
<td>06_step_turn_r.rbm</td>
<td>Turn to the right</td>
</tr>
<tr>
<td></td>
<td>07_handclap_l.rbm</td>
<td>Handclap to the left</td>
</tr>
<tr>
<td></td>
<td>08_con_move_b.rbm</td>
<td>Continuously move backward</td>
</tr>
<tr>
<td></td>
<td>09_handclap_r.rbm</td>
<td>Handclap to the right</td>
</tr>
<tr>
<td></td>
<td>handclap.rbm</td>
<td>Handclap</td>
</tr>
<tr>
<td></td>
<td>step_move_b.rbm</td>
<td>Move backward for a short distance</td>
</tr>
<tr>
<td></td>
<td>step_move_f.rbm</td>
<td>Move forward for a short distance</td>
</tr>
<tr>
<td></td>
<td>welcome.rbm</td>
<td>Welcome bow</td>
</tr>
<tr>
<td>Action File</td>
<td>01_move&amp;avoid.rba</td>
<td>Move forward and avoid obstacles by turning to the left</td>
</tr>
<tr>
<td></td>
<td>02_copy_clap.rba</td>
<td>Handclap in response to sound</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

4) Application

1) Create a motion that push down a paper cup.

5) Game

1) Hold a competition where you win if you first rush to a paper cup and push it down.
2) Hold a sumo wrestling game where you win if you push your opponent out of the field.
3.1.8. **Excavator**

1) **Goal of the Chapter**
In this chapter, users will build an excavator robot, a four-wheeled vehicle with an articulated automatic arm installed on it. Use eight wCK modules and the distance sensor to build the excavator. The excavator can freely move its arm to show dynamic pick and place motions.

2) **Robot Building**
The four-wheeled excavator robot is structured similar to real excavators. Follow the building instructions shown below step by step and assemble the excavator robot. Be cautious about the position of the wCK actuator module and the distance sensor.
ID MAP

Part List

- Body Frame x 1
- Control Box x 1
- Sensor Module x 1
- Hand Part x 1
- wCK module x 8
- wCK Cable x 8
- Joint 2 x 1
- Joint 4 x 2
- Joint 5 x 4
- Joint 6 x 2
- Joint 7 x 2
- Joint 8 x 1
- Joint 9 x 2
- Nut x 24
- Washer x 2
- B8 x 7
- B12 x 3
- B16 x 10
- B40 x 18
STEP 01

**Wheels**

- Bolt M2x8L: 4EA
- Nut M2: 14EA
- Bolt M2x40L:

*Use glue to attach the tire*

STEP 02

**Arm**

- Bolt M2x8L: 3EA
- Nut M2: 10EA
- Bolt M2x12L: 3EA
- Bolt M2x10L:
- Bolt M2x40L: 4EA
3) Running example robot files [motion file, action file]

< List of example files >

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_fork_l.rbm</td>
<td>Move the arm to the left</td>
</tr>
<tr>
<td></td>
<td>02_step_move_f.rbm</td>
<td>Move forward</td>
</tr>
<tr>
<td></td>
<td>03_fork_r.rbm</td>
<td>Move the arm to the right</td>
</tr>
<tr>
<td></td>
<td>04_step_turn_l.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>05_fork_f.rbm</td>
<td>Move the arm to the front</td>
</tr>
<tr>
<td></td>
<td>06_step_turn_r.rbm</td>
<td>Turn to the right</td>
</tr>
<tr>
<td></td>
<td>07_fork_scoop.rbm</td>
<td>Scoop</td>
</tr>
<tr>
<td></td>
<td>08_step_move_b.rbm</td>
<td>Move backward</td>
</tr>
<tr>
<td></td>
<td>09_fork_putdown.rbm</td>
<td>Lift up and put down the arm</td>
</tr>
<tr>
<td></td>
<td>blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>con_move_b.rbm</td>
<td>Continuously move forward</td>
</tr>
<tr>
<td></td>
<td>con_move_f.rbm</td>
<td>Continuously move backward</td>
</tr>
<tr>
<td></td>
<td>fork_up.rbm</td>
<td>Lift up the arm</td>
</tr>
<tr>
<td>Action File</td>
<td>01_demo1.rba</td>
<td>Stop movement in detection of an object approaching</td>
</tr>
</tbody>
</table>

1) Use the MotioBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

4) Application
1) Discuss how different this excavator robot is from real excavators.
2) Create a motion that uses the arm to turn to the right and the left.
3) Create a motion that the vehicle rises up when it fell over.

5) Game
1) Arrange a floor filled with some obstacles and hold a running competition.
2) Hold a sumo wrestling game where you win if you push your opponent out of the field. Use some dynamic movements of the arm.
1) Goal of the Chapter
In this chapter, users will build a four-legged doggy robot and learn how to program and control the movement of the four-legged mechanical system. Use eight wCK modules and the distance sensor to build the doggy robot.

2) Robot Building
Follow the building instructions shown below step by step and assemble the four-legged doggy robot. Be cautious about the position of the wCK actuator modules and the distance sensor. Please pay good attention to STEP02 and STEP03 where the connecting angle of Joint2 with the pentagon-shaped junction of the wCK modules is very important.
ID MAP

Top

Bottom

Part List

Body Frame x 1
Control Box x 1
Sensor Module x 1
wCK module x 8
wCK Cable x 8
Joint 2 x 4
Joint 11 x 4
Nut x 28
B6 x 4
B6 x 4
B12 x 6
B16 x 8
B40 x 14
STEP 01

**Sensor**
- Bolt M2x6L: 4EA
- Nut M2: 4EA
- Bolt M2x12L: 2EA

STEP 02

**다리 조립**
- Bolt M2x12L: 2EA  
- Bolt M2x8L: 2EA
- Nut M2: 11EA
- Bolt M2x16L: 4EA
- Bolt M2x40L: 7EA

*Caution*  
The adjustment of the connecting angle is critically important.
STEP 03

---

**Legs**

- Bolt M2x12L : 2EA
- Bolt M2x8L : 2EA
- Nut M2 : 11EA
- Bolt M2x16L : 4EA
- Bolt M2x40L : 7EA

---

The adjustment of the connecting angle is critically important.

STEP 04

---

**RBC**

※ Adjust wCK cables first
3) Principles of four-legged Crawling motion

1) The four-legged crawling motions can be created by imitating quadruped animals.
2) There are five patterns of four-legged crawling. They are crawl, walk, trot, pace, and bound.

<table>
<thead>
<tr>
<th>Crawling Pattern</th>
<th>Grounding Rate</th>
<th>No of legs on ground</th>
<th>phase difference (left and right)</th>
<th>phase difference (front and rear)</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Crawl&quot;</td>
<td>0.75+</td>
<td>3 or 4</td>
<td>0.5</td>
<td>1 – (No of legs on ground)</td>
<td>Low</td>
</tr>
<tr>
<td>&quot;Walk&quot;</td>
<td>0.75~0.5</td>
<td>3</td>
<td>0.5</td>
<td>0</td>
<td>Middle</td>
</tr>
<tr>
<td>&quot;Trot&quot;</td>
<td>0.5</td>
<td>2</td>
<td>0</td>
<td>0.5</td>
<td>High</td>
</tr>
<tr>
<td>&quot;Pace&quot;</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;Bound&quot;</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3) The “Crawl” pattern is slowest but most stable because it always uses three or four legs to step on the ground. The example motions are based on the “Crawl” pattern.

4) The principle of crawl forward motion (crawl_f.rbm)
- If the initial posture of the doggy robot has the front right leg ahead of the front left leg, and the rear right leg ahead of rear left leg, the crawling steps are taken as below.

<table>
<thead>
<tr>
<th>STEP</th>
<th>Rear Left</th>
<th>Front Left</th>
<th>Rear Right</th>
<th>Front Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Forward 3</td>
<td>Backward 1</td>
<td>Backward 1</td>
<td>Backward 1</td>
</tr>
<tr>
<td>2</td>
<td>Forward 3</td>
<td>Backward 1</td>
<td>Backward 1</td>
<td>Backward 1</td>
</tr>
<tr>
<td>3</td>
<td>Backward 1</td>
<td>Forward 3</td>
<td>Backward 1</td>
<td>Backward 1</td>
</tr>
<tr>
<td>4</td>
<td>Backward 1</td>
<td>Forward 3</td>
<td>Backward 1</td>
<td>Backward 1</td>
</tr>
<tr>
<td>5</td>
<td>Backward 1</td>
<td>Backward 1</td>
<td>Forward 3</td>
<td>Backward 1</td>
</tr>
<tr>
<td>6</td>
<td>Backward 1</td>
<td>Backward 1</td>
<td>Forward 3</td>
<td>Backward 1</td>
</tr>
<tr>
<td>7</td>
<td>Backward 1</td>
<td>Backward 1</td>
<td>Backward 1</td>
<td>Forward 3</td>
</tr>
<tr>
<td>8</td>
<td>Backward 1</td>
<td>Backward 1</td>
<td>Backward 1</td>
<td>Forward 3</td>
</tr>
</tbody>
</table>

※ The numbers in the table are not absolute values but relative values.

5) The principle of crawl backward motion (crawl_b.rbm)
- If the initial posture of the doggy robot has the front right leg ahead of the front left leg, and the rear right leg ahead of rear left leg, the crawling steps are taken as below.

<table>
<thead>
<tr>
<th>STEP</th>
<th>Rear Left</th>
<th>Front Left</th>
<th>Rear Right</th>
<th>Front Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Backward 3</td>
</tr>
<tr>
<td>2</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Backward 3</td>
</tr>
<tr>
<td>3</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Backward 3</td>
</tr>
<tr>
<td>4</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Backward 3</td>
<td>Forward 1</td>
</tr>
<tr>
<td>5</td>
<td>Forward 1</td>
<td>Backward 3</td>
<td>Forward 1</td>
<td>Forward 1</td>
</tr>
<tr>
<td>6</td>
<td>Forward 1</td>
<td>Backward 3</td>
<td>Forward 1</td>
<td>Forward 1</td>
</tr>
<tr>
<td>7</td>
<td>Backward 3</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Forward 1</td>
</tr>
<tr>
<td>8</td>
<td>Backward 3</td>
<td>Forward 1</td>
<td>Forward 1</td>
<td>Forward 1</td>
</tr>
</tbody>
</table>
4) Running example robot files [motion file, action file]

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>02_crawl_f.rbm</td>
<td>Crawl forward</td>
</tr>
<tr>
<td></td>
<td>03_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>04_turn_l.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>05_basic_pose.rbm</td>
<td>Basic posture</td>
</tr>
<tr>
<td></td>
<td>06_turn_r.rbm</td>
<td>Turn to the right</td>
</tr>
<tr>
<td></td>
<td>07_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>08_crawl_b.rbm</td>
<td>Crawl backward</td>
</tr>
<tr>
<td>Action File</td>
<td>01_move&amp;avoid.rba</td>
<td>Crawl forward and avoid obstacles by turning to the left</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

5) Application
1) Create various motions other than crawling, and modify the leg structure too.
2) Create motions of crawling sideways.

6) Game
1) Hold a straight course running competition.
2) Arrange a floor filled with some obstacles and hold a running competition.
3) Hold a sumo wrestling game where you win if you push your opponent out of the field.
3.1.10. Four-legged Wheel bot

1) Goal of the Chapter
In this chapter, users will build a four-legged wheel bot that can walk as well as turn its wheels. Users can learn how to program more complicated motions of quadruped robot that involves rolling and crawling together. Use eight wCK modules and the distance sensor to build the wheel robot.

2) Robot Building
Follow the building instructions shown below step by step and assemble the four-legged wheel bot. Be cautious about the position of the wCK actuator modules and the distance sensor.
# ID MAP

## Top

- ID 02
- ID 03
- ID 06
- ID 07
- ID 08
- ID 05
- ID 00
- ID 04
- ID 01

## Bottom

- ID 04
- ID 05
- ID 03
- ID 02
- ID 01
- ID 00
- ID 07
- ID 06

## Part List

- **Body Frame x 1**
- **Control Box x 1**
- **Sensor Module x 1**
- **wCK module x 8**
- **wCK Cable x 8**
- **Nut x 26**
- **B6 x 4**
- **B8 x 4**
- **B12 x 8**
- **B16 x 8**
- **B40 x 14**
STEP 01

**Sensor**

- Bolt M2x6L : 4EA
- Nut M2 : 4EA
- Bolt M2x12L : 2EA

*Adjust wCK cables first*

STEP 02

**Legs**

- Bolt M2x12L : 2EA
- Nut M2 : 11EA
- Bolt M2x16L : 4EA
- Bolt M2x40L : 7EA

*Use glue to attach the tire*
3) Running example robot files [motion file, action file]

< List of example files >

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_wh_con_f.rbm</td>
<td>Continuously drive forward</td>
</tr>
<tr>
<td></td>
<td>02_wh_step_f.rbm</td>
<td>Drive forward for a short distance</td>
</tr>
<tr>
<td></td>
<td>03_crawl_f.rbm</td>
<td>Crawl forward</td>
</tr>
<tr>
<td></td>
<td>04_wh_turn_l.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>05_stop.rbm</td>
<td>Basic posture</td>
</tr>
<tr>
<td></td>
<td>06_wh_turn_r.rbm</td>
<td>Turn to the right</td>
</tr>
<tr>
<td></td>
<td>07_wh_con_b.rbm</td>
<td>Continuously drive backward</td>
</tr>
<tr>
<td></td>
<td>08_wh_step_b.rbm</td>
<td>Drive backward for a short distance</td>
</tr>
<tr>
<td></td>
<td>09_crawl_b.rbm</td>
<td>Crawl backward</td>
</tr>
<tr>
<td></td>
<td>crawl_turn_l.rbm</td>
<td>Step to the left</td>
</tr>
<tr>
<td></td>
<td>crawl_turn_r.rbm</td>
<td>Step to the right</td>
</tr>
<tr>
<td>Action File</td>
<td>01_move&amp;avoid.rba</td>
<td>Move forward and avoid obstacles by turning to the left</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.
2) Download the motion files to RBC control box and run them using the remote controller.
3) Download the action files to RBC control box and run them using the remote controller.

4) The Principles of the Motions

1) The example motions that involve the four wheels are based on the same principles of the four-wheeled vehicle of Chapter 2-6.
2) The example motions that involve the crawling of the four legs are based on the same principles of the doggy robot of Chapter 2-9.

5) Application

1) Create a motion that move forward using the wheels with three legs on the ground and the other leg lifted.
2) Create a motion of climbing up stairs.
3) Increase the crawling speed by turning the wheels together.
4) Use other joints to increase the length of the legs. Modify motions and see how it affects the movement.

6) Game

1) Hold various types of running competition such as straight course, curved course, course with obstacles.
3.1.11. **Snake Robot**

1) **Goal of the Chapter**
In this chapter, users will build a snake robot that has the shape and movement of a real snake. Users can learn how to simulate and program the movement of a snake that involves the shifting of the center of gravity, friction, time delay of joints, and inertia. Use eight wCK modules and the distance sensor to build the wheel robot.

2) **Robot Building**
Follow the building instructions shown below step by step and assemble the snake robot. Attach Joint5 to four wCK modules facing ground in order to use friction as the source of moving power. Be cautious that the bolts are fastened tight enough to avoid malfunction of the sophisticated movement.
ID MAP

Left Side

Part List

Body Frame x 1
Control Box x 1
Sensor Module x 1
wCK module x 8
wCK Cable x 8
Joint 11 x 1
Joint 12 x 2
Joint 1 x 6
Joint 4 x 1
Joint 5 x 4
Joint 7 x 2
Joint 8 x 2
Joint 10 x 1
B16 x 20
Nut x 24
Washer x 2
B8 x 4
B12 x 12
B40 x 4
STEP 01

Sensor

- Bolt M2x12L : 4EA
- Nut M2 : 1EA
- Bolt M2x16L : 5EA

STEP 02

Body A

- Bolt M2x8L : 2EA
- Nut M2 : 8EA
- Bolt M2x12L : 2EA
- Bolt M2x16L : 5EA
- Bolt M2x40L : 4EA

Use glue to attach the tire
STEP 03

**Body B**

<table>
<thead>
<tr>
<th>Bolt M2x6L</th>
<th>1 EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt M2x12L</td>
<td>3 EA</td>
</tr>
<tr>
<td>Nut M2</td>
<td>9 EA</td>
</tr>
<tr>
<td>Bolt M2x16L</td>
<td>6 EA</td>
</tr>
</tbody>
</table>

Use glue to attach the tire

STEP 04

**Body C**

<table>
<thead>
<tr>
<th>Bolt M2x8L</th>
<th>1 EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nut M2</td>
<td>7 EA</td>
</tr>
<tr>
<td>Bolt M2x12L</td>
<td>3 EA</td>
</tr>
<tr>
<td>Bolt M2x16L</td>
<td>4 EA</td>
</tr>
</tbody>
</table>

Use glue to attach the tire
3) **The principles of snake motion**

1) Snake robots have a different way of movement from other types of articulated robots such as humanoid or doggy robots. Because snake robots can be mobilized in areas that are inaccessible for humans, they are widely used and studied for the purposes of rescue and military operation. Snake robots can move forward and backward by moving the sine curve shaped joint array. It also uses the inertial force to make turns.

2) There are four patterns of crawling. They are lateral undulation, rectilinear, concertina, side winding.

3) The example motions are based on the pattern of lateral undulation.

4) **Running example robot files [motion file, action file]**

< List of example files>

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>02_move_f.rbm</td>
<td>Move forward</td>
</tr>
<tr>
<td></td>
<td>03_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>04_turn_l.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>05_basic_pose.rbm</td>
<td>Basic posture</td>
</tr>
<tr>
<td></td>
<td>06_turn_r.rbm</td>
<td>Turn to the left</td>
</tr>
<tr>
<td></td>
<td>07_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>08_move_b.rbm</td>
<td>Move backward</td>
</tr>
<tr>
<td>Action File</td>
<td>01_move&amp;stop.rba</td>
<td>Move forward if no object is detected</td>
</tr>
</tbody>
</table>

1) Use the MotionBuilder to open and run the example motion files.

2) Download the motion files to RBC control box and run them using the remote controller.

3) Download the action files to RBC control box and run them using the remote controller.

5) **Application**

1) Create a motion that the snake robot gets up when it flip over to the side.

6) **Game**

1) Arrange a routine for race and hold a crawling competition.
3.1.12. Biped Robot

1) Goal of the Chapter
In this chapter, users will build a biped robot that has the shape and movement of human legs. Users can learn how to simulate and program the movement of human legs that involves the shifting of the center of gravity, friction, and inertia. Use eight wCK modules and the distance sensor to build the biped robot.

2) Robot Building
One leg consists of four wCK modules. Two of them move the center of gravity to the left and the right, while the other two to the front and the rear. Follow the building instructions shown below step by step and assemble the biped robot. Please pay good attention to STEP02 and STEP03 where the connecting angle of Joint2 with the pentagon-shaped junction of the wCK modules is very important. Be cautious that the bolts are fastened tight enough to avoid malfunction of the sophisticated movement.
ID MAP

Part List

- Body Frame x 1
- Control Box x 1
- Sensor Module x 1
- Foot Part x 2
- wCK module x 8
- wCK Cable x 8
- Joint 2 x 2
- Joint 4 x 2
- Joint 10 x 2
- Nut x 32
- B6 x 4
- B8 x 2
- B12 x 6
- B18 x 8
- B40 x 16
STEP 01

Sensor

<table>
<thead>
<tr>
<th>Nut M2 : 4EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt M2x6L : 4EA</td>
</tr>
<tr>
<td>Bolt M2x12L : 2EA</td>
</tr>
</tbody>
</table>

STEP 02

Left Leg

<table>
<thead>
<tr>
<th>Bolt M2x8L : 1EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt M2x12L : 3EA</td>
</tr>
<tr>
<td>Nut M2 : 14EA</td>
</tr>
<tr>
<td>Bolt M2x16L : 4EA</td>
</tr>
<tr>
<td>Bolt M2x40L : 8EA</td>
</tr>
</tbody>
</table>

Caution

The adjustment of the connecting angle is critically important.
STEP 03

Right Leg

- Bolt M2x8L: 1EA
- Bolt M2x12L: 3EA
- Nut M2: 14EA
- Bolt M2x16L: 4EA
- Bolt M2x40L: 8EA

The adjustment of the connecting angle is critically important.

STEP 04

RBC

* push when remove RBC

* Adjust wCK cables first
3) The Principles of Biped Walking
1) There are generally two patterns of biped walking. They are the static walking and the dynamic walking. The static walking always keeps the center of gravity within the sole of the stepping foot, thus the robot won’t fall over even when it stops walking. While the dynamic walking keeps the center of gravity outside the sole of the stepping foot. Therefore, the dynamic walking is a repeated process of making the robot start to fall over to the front when it step forward to continue walking without falling. In the dynamic walking, the robot continuously calculates the ZMP (Zero Moment Point) based on the inputs from sensing devices such as accelerometer and pressure sensor. The robot controls its power and keeps walking by maintaining the ZMP within the sole of the stepping foot. The ZMP is the point where the sum of the moment caused by the force produced at the sole of the foot becomes zero. In other words, it is the point within the stepping sole where the sum of the gravity, inertial force, and the repulsive force against the ground becomes zero.

In order to implement the dynamic walking, a powerful computer is required that can process complicated calculations in real time in addition to the accelerometer and pressure sensor.
2) The example motions are based on the static walking pattern, which is relatively simple for many users. Following are the steps to take along with the rules to use in programming motions of static walking.
3) In order to lift one leg, the center of gravity should be moved to the other foot. Therefore, if you want to lift the left leg, you have to lean to the right in order to move the center of gravity to the right foot.
4) If you stretch the lifted left leg to the front, the center of gravity will naturally move to the front. Therefore you have to lean the body backward in order to keep the center of gravity within the right foot while you stretch the left leg without falling over.
5) After you step down the left leg, lean the body forward so that the center of gravity is moved to the left leg.
6) Move the right leg to front and align with the left leg. You can either lift the right leg or drag it.
7) Just like step 3), lean to the left in order to lift the right leg.
8) Lean backward and stretch the lifted right leg to the front.
9) Step down the right leg and lean forward. Move the left leg to align with the right leg.
10) Repeat step 3) to 9) and continue to program the walking motion.

4) Running example robot files [motion file, action file]
< List of example files >

<table>
<thead>
<tr>
<th>Category</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion File</td>
<td>01_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>02_walk_f.rbm</td>
<td>Walk forward</td>
</tr>
<tr>
<td></td>
<td>03_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>04_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>05_basic_pose.rbm</td>
<td>Basic posture</td>
</tr>
<tr>
<td></td>
<td>06_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>07_blank.rbm</td>
<td>No movement</td>
</tr>
<tr>
<td></td>
<td>08_walk_b.rbm</td>
<td>Walk backward</td>
</tr>
<tr>
<td>Action File</td>
<td>01_move&amp;stop.rba</td>
<td>Move forward if no obstacle is detected</td>
</tr>
</tbody>
</table>
Run and see how the example motions work. Study and understand the principles of biped walking and develop advanced walking motions.

5) Application
1) Develop various motions other than walking movement as well as modify the leg structure to study how it affects the biped walking motion.
2) Create kicking motions so as to hold a soccer game.
3) Program a turning motion by means of having the two feet slip in the opposite directions.
4) Program different motions of getting up when the robot fall over.

6) Game
1) Hold a running competition where you win if you first rush to and kick a paper cup.

7) TIP
In the static walking, the lower the center of gravity is positioned, the more stable the walking motion becomes. The reason that a biped robot falls over is because that the center of gravity was moved out of the sole of the stepping foot. If you lower the body, the center of gravity gets closer to the sole of the foot. Attach a weight to each leg and test how it affects the walking motion.
3.2. Intermediate Course

In this course, user will learn how to create and revise the robot motions by using 16 module humanoid robot, and how to use special function of wCK robot module.

Generally, “Programming Robot Motion” means “Teaching Robot”. Therefore, it is called “MotionBuilder” is a kind of “Teaching Software”.

Here are procedures in order to create motions in RoboBuilder.

Create Project => Create Motion File => Create New Scene => Revise Last Posture of the Scene => Add New Scene => Revise the Last Posture of Added Scene =>…….=> Create Last Scence => Save Motion File.

One motion is consist of several scence. User revise every last posture of scene. Using RoboBuilder, there are three ways of making Robot’s posture.

a) Use Jog Dial in Robot Configuration
b) Change the wCK module display value
c) Rotate Robot module directly and capture the posture

3.2.1. Creating New Motion (method a, method b)

Let’s make simple motion (Spread & Narrow HUNO’s Arm) by using Jog Dial method and Change the value.

1) Run MotionBuilder and Click “New” button, then “New Project” window is shown.
2) Name the Project Name as 'test1'. And, click [Browse] button for saving Project File. Then, Select "Creator Huno" and click "OK" button.

※ If you select standard platform among "HUNO, DINO, DOGY", please go to 6).

3) If you chose a non-standard robot platform (Creator Others) in step 2), click [Config] button. The Config window appears as below.

4) Click [Add wCK] and the 'Add Motor' window appears so that you can input the wCK information for the project. Add as many wCK modules as the actual number of the wCK modules of the robot.
The input options are as below.

- **wCK ID**: Each wCK module has its own ID so that the controller can send commands to the wCK with a specific ID. This ID should match with the actual ID of the robot.
  <wCK ID picture>

- **Min Limit, Max Limit**: The minimum and maximum rotational position of the wCK module. The actuator can not move to a higher position than Max Limit or move to a lower position than Min Limit.

- **Init Position**: The initial position of the wCK module. This decides the start position of the first scene of a motion file. In many cases of creating or modifying a motion, the motion starts with a particular posture and ends with that same posture. Therefore, setting this Init Position well helps to make a scene that returns to the start posture easily.

- **PGain, DGain, IGain**: This option sets the P, D, I gains that robot modules use during operation. The values are used only in operation.

5) If you finish adding the wCK modules, click **Set Config** button and then click **Close** button. As you can see below, wCK information has now been added in the ‘Robot Configuration’ area and the ‘wCK module Control Detail’ area.

6) Click **New Motion** button to create a new motion. Input the options carefully and click **OK** button to finish.
The input options are as below.

- **Motion Name**: The name of the motion, which is also used as the name of the motion file.
- **File Path**: The directory path where the motion is save. The path is by default same with the directory set in step 2). This path doesn’t need to be changed.
- **Motion Configurations**: This option sets the P, D, I gains and check if you will use the external port or not.
  ※ In case of a robot whose wCK modules have LEDs built in such as 5720T kit, users can control the LED lights by checking this option.
- **Author, Email Address, Serial Code**: This option is not to be used with current version v1.3.

7) The window appears so that you can input the scene properties. Input the options and click **OK** button to finish.

![Scene Configuration Window](image1)

The input options are as below.

- **Scene Name**: The name of the scene.
- **Number of Frame**: The number of the frames of the scene, which ranges from 1 to 100.
- **Transition Time [ms]**: The execution time of the scene, which ranges from 20 to 6000.

As you can see in the screen, the motion name is changed and the first scene is added now.

![Scene Configuration](image2)
8) Connect com port as described in chapter 2.2 3). COM port connection can be done in any step of 1)~8).

The Disp. Value in the center of the screen is set zero(0) as shown above. This means that the disposition is zero, which means the start position and destination position is currently same.

9) Let's make the robot hold up the arms by dragging the jog dial of ID10 and ID13 in the ‘Robot Configuration’ area.

You can see the D. Pos and Disp. Of the ID10 and ID13 are changed now.

※ If you click the Restore button while you are modifying the destination position of a scene, the Disp. turns to zero and the robot moves to the start position.
10) Click [New Scene] button and add the second scene as explained in step 7). Use ↓ key or the mouse to choose the added second scene.

You can see the Disp. Values of the second scene are all set to zero.

11) Let's make the robot put the arms to the side by dragging the jog dial of ID11 and ID14 in the ‘Robot Configuration’ area.

12) Follow step 10~11) and add the third scene. Let's type in the Disp. Value directly for ID11 and ID14 to make the robot bend the arms inside to the center.

Double click the Disp. of the ID 11 Disp. and the text key in window opens.
Type in -88 and press Enter. Type in 89 for ID13.

13) The scenes made until now are as below.

Scene_0 : Hole up the arms, 30 frames, 1 second
Scene_1 : Stretch the arms to the sides, 30 frames, 1 second
Scene_2 : Bend the arms inside to the center, 30 frames, 1 second

In summary, the robot will hold up the arms and stretch out the arms to the sides and then bend the arms inside to the center of the chest.

14) Now let’s make the robot stretch and bend the arms one more time.
Since the same motion is repeated, you can use copy and paste the previous scenes.
Drag the mouse and select Scene_2 and Scene_3. Right click the mouse.
Choose Copy here.

15) Select the empty line below Scene_3 and right click the mouse to choose Paste/Insert.

As seen below, the Scene_2 and Scene_3 are pasted now.
16) Add the last scene by following the step 11-12). The scene name is 'Scene_6'.

17) It is helpful for the robot to move stably that you set the last scene ends with the initial posture. Click Scene_6 and right click the mouse to choose InitPos.
As seen below, the destination position of the Scene_6 is initial posture.

![Image of initial posture]

18) Click **Save** button to save the motion file 'move_arm1.rbm'.

![Image of save motion file]

※ When you save the file, the RBC control box has to be connected and the com port is open. It fails to save the file if the communication between PC and RBC has a problem.

19) Let's test the created motion. Click **Select All** button and click **Test** button.
3.2.2. Creating New Motion (method c)

Let’s use method c (motion teaching method) and make the robot bend and straighten the knees.

1) Follow chapter 5.1 1~8) and create a motion named ‘bend_knee’. Connect the COM Port.

As seen above, the Disp. Of all wCK modules are set to zero.

2) Click Get Pos button and this window appears.

In this window, you select the wCK modules that you want to adjust. When the robot bend its knees, the robot will fall back if it bend only the knees. Therefore, select the six IDs so that it can flexibly bend its ankles and hip joints together.

3) If you click Close button, you can see the selected six wCK modules are loose so that you can adjust its posture with hands. Touch and adjust the robot so that it bend its knees and stand straight upward. Then click Capture and you will see the data changed as shown below.
Caution: In case communication error occurs during the Capture (unplugged cable, control box power off etc), the MotionBuilder may not work properly. If the program window expands press Alt+F4 to close the program. Check the communication status and try again.

4) Click **New Scene** button and add a scene. Use ↓ key or the mouse to choose the added second scene.

5) Let's capture the robot when it straighten the knees. Click **Get Pos** button and select the same six IDs(1, 2, 3, 6, 7, 8). Touch and adjust the robot to make it straighten the knees and stand straight upward. Click **Capture** button then you will see the data changed as shown in the below.
6) Now two scenes have been completed (bending the knees, straightening the knees). Now let's copy and paste the two scenes so that the robot can repeat the knee bending motion. As explained in chapter 5.1 (14~15) copy and paste the Scene_0 and Scene_1.
7) Let's add the last scene of initial posture. As explained in chapter 5.1 16~17) click New Scene button to add a new scene. Right click the mouse and choose InitPos.

8) Click Save button and save the created motion 'bend_knee.rbm'. If you click Save All button, the 'test1.prj' and 'bend_knee.rbm' are all saved.

9) Let's test the created motion. Click Select All button and click Test button.
3.2.3. **Creating New Motion (Non-Standard Platform)**

In previous motion creatings were available as long as user assembled into HUNO. In this chapter, it describes that user assembled into Non-Standard Platform Robot (4 modules assembled into “Car” robot).

01. Set up COM Port and Baudrate, click “ScanPort”. If PC and RoboBuilder connected properly, “ClosePort” button will be shown.

02. Click “New” in menu bar. In the [New Project] window, assign project name, project path, and select robot platform, then, click [OK].
03. Non-Standard platform is created, Click the “Config” button in order to add “wCK module” in Robot Configuration panel.

04. Click “Add wCK” button to add wCK module.
05. In “Config” window, add 4 wCK module for “four-wheeled vehicle robot ”, then click “Set Config” button, and click “Close”.

06. 4 wCK modules are created. Click “New Motion” in order to add motion. In “New Motion” window, input Motion Name and designate “File Path”, then click “OK” button.
07. Register first scene in the created motion. Setup Scene name, Frame number, Transition Time. Then, click [OK].

08. Now new scene is registered.

<table>
<thead>
<tr>
<th>No</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>It shows wCK module ID.</td>
</tr>
<tr>
<td>2</td>
<td>S.Pos</td>
<td>It shows the start position of the wCK module.</td>
</tr>
<tr>
<td>3</td>
<td>D.Pos</td>
<td>It shows the end position of the wCK module.</td>
</tr>
<tr>
<td>4</td>
<td>Disp</td>
<td>It shows the Displacement between S.Pos and D.Pos.</td>
</tr>
<tr>
<td>5</td>
<td>Torq</td>
<td>It shows wCK module operation speed.</td>
</tr>
<tr>
<td>6</td>
<td>Port</td>
<td>It shows wCK module LED status. (only available for Transparent wCK module.)</td>
</tr>
<tr>
<td>7</td>
<td>Scene.Name</td>
<td>It shows scene name</td>
</tr>
<tr>
<td>8</td>
<td>Frames</td>
<td>It shows the numbers of Frame.</td>
</tr>
<tr>
<td>9</td>
<td>Tr.Time[ms]</td>
<td>It shows Transition Time (ms)</td>
</tr>
</tbody>
</table>
09. If click “Test” button, it moves to end point of scene, and it changes “Return” status.

10. If you adjusted the end posture, click “Return” button. Then, it saves wCK module set value, and the button changes to “Test” status.

Tip: If you select “Scene”, wCK module moves to start position.

If you select “Scene” and click “Test” button, Robot moves as wCK set value. Also, Scene can be selected one or more at one time, therefore, it could be tested continuously. If “Repeat” check box is selected, it repeats “Scene” movement.

11. If all Scene are completed, click “Save” button in order to save Project and Motion file.
3.2.4. Change the Motion Speed
In previous chapters, the frame and transition time were default values (frame 30, transition time 1000ms). In this chapter, let's study how to change frame and transition time.

1) Let's change the speed of the motion ‘bend_knee.rbm’ that we created in the previous chapter. Run MotionBuilder and click [Open] button to open the project ‘test1.prj’.

2) Click [Motion List] button and double click ‘bend_knee’.

The data for ‘bend_knee.rbm’ appear. Connect RoboBuilder to PC and open the com port.
3) You can see the frame 30, Transition time 1000ms in the scene editing area.

4) Let's increase the speed of Scene_0. Double click the scene.

The scene's property window appears. Input 25 for the Number of Frame and 500 for the Transition Time and click [OK]. The frame ranges from 1 up to 100 and the minimum transition time per frame is 20ms. Therefore, the transition time is affected by frame number and ranges from 20 up to 6000ms. (refer to chapter 1.3 for detail)

※ If the Number of Frame is 30, the Transition Time should be more than 600 (30 x 20 = 600). Therefore you cannot change it to 500 without changing the Transition Time first.
5) As shown in the screen below, the frames and transition times for all scenes are changed to 25 and 500 respectively.

6) Let's test the changed motion. Click Select All button and click Test button. You can see the motion is faster now.
3.2.5. Merging motion files

Let's combine 'move_arm1.rbm' and 'bend_knee.rbm' to make a new motion 'merge1.rbm'.

1) Run MotionBuilder and click **Open** button to open the project ‘test1.prj’.

2) Click **New Motion** button.

Set the Motion Name as ‘merge1’ and click **OK** button. If the new scene window appears, leave the default values and click **OK** and then connect com port.

3) Click **Save All** button to save 'test1.prj' and 'merge1.rbm'.
4) Click **Motion List** button and double click ‘move_arm1’.

![Motion Management](image)

5) Click **Select All** button and select all scenes of the ‘move_arm1.rbm’. Right click the mouse and choose **Copy**.

![Motion List](image)

6) Click **Motion List** button and double click ‘merge1’.

Click the second line and right click the mouse to choose **Paste-Insert**.

![Motion List](image)

As seen below, the scenes of ‘move_arm1.rbm’ have been added to ‘merge1.rbm’.

![Motion List](image)
7) Click **Save** button to save the 'merge1.rbm'.

8) Click **Motion List** button and double click 'bend_knee'.

9) Click **Select All** to select all scenes of 'bend_knee.rbm'. Right click and choose **Copy**.

10) Click **Motion List** button and double click 'merge1'.

Click the next line of the last scene of 'merge1.rbm'. Right click the mouse to choose **Paste-Insert**.
As shown below, all scenes of `bend_knee.rbm` have been added to `merge1.rbm`.

<table>
<thead>
<tr>
<th>ID</th>
<th>Pos</th>
<th>Tick</th>
<th>Scene Name</th>
<th>Frames</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>126</td>
<td>0</td>
<td>2</td>
<td>Scene_0</td>
<td>30</td>
<td>1000</td>
</tr>
<tr>
<td>176</td>
<td>0</td>
<td>2</td>
<td>Scene_0</td>
<td>30</td>
<td>1000</td>
</tr>
<tr>
<td>191</td>
<td>0</td>
<td>2</td>
<td>Scene_1</td>
<td>30</td>
<td>1000</td>
</tr>
<tr>
<td>188</td>
<td>0</td>
<td>2</td>
<td>Scene_2</td>
<td>30</td>
<td>1000</td>
</tr>
<tr>
<td>168</td>
<td>0</td>
<td>2</td>
<td>Scene_3</td>
<td>30</td>
<td>1000</td>
</tr>
<tr>
<td>156</td>
<td>0</td>
<td>2</td>
<td>Scene_4</td>
<td>25</td>
<td>500</td>
</tr>
<tr>
<td>141</td>
<td>0</td>
<td>2</td>
<td>Scene_1</td>
<td>25</td>
<td>500</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>2</td>
<td>Scene_0</td>
<td>25</td>
<td>500</td>
</tr>
<tr>
<td>47</td>
<td>0</td>
<td>2</td>
<td>Scene_1</td>
<td>25</td>
<td>500</td>
</tr>
<tr>
<td>85</td>
<td>0</td>
<td>2</td>
<td>Scene_4</td>
<td>25</td>
<td>500</td>
</tr>
<tr>
<td>150</td>
<td>0</td>
<td>2</td>
<td>Scene_4</td>
<td>25</td>
<td>500</td>
</tr>
</tbody>
</table>

11) Click **Save** button to save `merge1.rbm`.

12) Let’s test the merged motion. Click **Select All** button to select all scenes and click **Test** button. You can see the `move_arm1.rbm` and `bend_knee.rbm` are merged now.
### 3.2.6. Using the wheel mode of wCK module

Let's modify the left arm of HUNO so that a 360 degree turning shield can be installed.

1) Disconnect ID12 wCK module from HUNO as below.

![Disconnected wCK module](image1.png)

2) Use J4 and J5 to add the shield as shown in the picture.

![Added shield](image2.png)

3) A new project for non-standard robot need to be created in order to use the wheel mode. Click **New** button.

![New project dialog](image3.png)
4) Set the Project Name as ‘huno_wheel1’ and click Browse button to set the save path. Choose ‘Creator Others’ for Robot Platform and click OK to finish.

※ Standard platform robots(HUNO, DINO, DOGY) don’t support the 360 degree wheel mode.

5) As explained in chapter 5.1 3–5), add 16 IDs in the Config window.

6) Click Save All button to save ‘huno_wheel1.prj’.
7) **Because the robot is not a standard platform, the initial posture need to be defined.** Click **Init Pos** button to set the initial positions of wCK modules.

All initial positions are set to 127 now. But this should be changed because it is not appropriate for HUNO.

**There are two methods to change the initial position as below.**

- **method a.** Double click the Int_Position of each ID or use jog dial pad
- **method b.** Call up the initial position file (*.mip)

8) **As the robot’s structure is almost same with HUNO, let’s call up the initial position file for HUNO.**

First, Click **Close** to close the ‘Set Initial Postion’ window. Click **Open** button to open ‘HunoDemo.prj’.

Click **Init Pos** button and the ‘Set Initial Postion’ window appears.
Click 'Set Initial Position' and save the file as 'huno.mip'.

Click Close button to close 'Set Initial Position'.

9) Click Open button to open the project file ‘huno_wheel1.prj’.

10) Click Init Pos button and ‘Set Initial Position’ appears.
    Click Load from File button and choose ‘huno.mip’ file that you saved earlier.

Now you can see the initial positions are changed and set with the values of HUNO.

Click Set Initial Position button so that the values are applied for the project. Click Close button.

11) Now let’s begin to program a motion.
    Click New Motion button and set the Motion Name as ‘huno_rotation1’. Click OK button to close the window.
    Click OK button for the New Scene window.
    Connect COM port.
12) You can click the Test button to test the initial posture of the first scene.

![Image of RoboBuilder](image.png)

13) You can see the left arm is not very natural because of the modification. Let’s change the initial posture.

Click Init Pos button to see ‘Set Initial Position’ window. Choose ID 10 and change the Init_Position value to 89 by using the jog dial pad or double clicking the value.

![Image of RoboBuilder](image.png)

Click Set Initial Position button so that the change is applied for the project. Click Close button to finish.

14) Although the initial posture is changed, the start posture of the first scene remains unchanged.
If you delete the first scene and make new scene, the changed initial posture is applied.

Click **Delete** button and the Scene_0 is deleted.

Click **New Scene** button and click **OK** for the New Scene window.

Now you can see the initial posture of the first scene has been changed.

15) Let's try turning the wCK ID12 clockwise in Scene_0. If you double click the 'Torq' of ID12 in the 'wCK module Control Detail' area, a small pop up appears as below.

Choose ‘Wheel-CW’.

In 360 degree wheel mode, the start position and destination position is used for speed. It ranges from 0(stop) up to 15(max speed).

Double click the start position or destination position and input 15.
16) Because there is only one scene now, the ID12 will turn endlessly. Therefore, let's add another scene to make it stop.

Click **New Scene** button and click **OK** for the New Scene window.

You can see the newly added Scene_1 is not in wheel mode but in position control mode with torque level2.

17) In this setting, ID12 will move to position 79 instead of stopping the turning. Let's change like this.

18) Let's try the motion. Click **Select All** button to select all scenes and click **Test** button. You can see the shield turns clockwise for one second and stops.
3.2.7. **Using LED Light Function (5720T kit model only)**

Let's learn how to control the LEDs in ID10 wCK which is located in the check in order to make HUNO look like its heart beats.

1) Open a HUNO project file such as ‘HunoDemo.prj’.

2) Click **New Motion** button.

   ![Motion Configuration](image)

   Set the Motion Name as ‘heartLed’
   Check the **Enable External Port** in order to control the external port where the LEDs are connected. Click **OK** button.

   ![Port Connection](image)

   Click **OK** for the New Scene window.

   Connect COM port.

3) Let’s control the LED of ID10 in Scene_0.

   Double click ‘Port’ of ID10 then a small pop up appears.
Choose ‘P1:0(R)’ to turn on the red LED.

4) Click [New Scene] button to add Scene_1. If you click Scene_1, you can see the ‘Port’ of ID10 is ‘P0:0(OFF)’.

5) In order to repeat the LED lighting two more times, select Scene_0 and Scene_1. Right click the mouse and choose [Copy].

6) Click the next line of Scene_1, and right click the mouse to choose [Paste-Insert].
7) Repeat 5~6).

8) Let’s try the motion. Click **Select All** button to select all scenes and click **Test** button. You can see the heart beats three times.
3.2.8. **MotionBuilder Advanced Function (1) : Frame & Transition Time**

This chapter explains how to use the transition time and frame in motion building procedure.

1) Frame ranges from 1 up to 100. The start posture of a scene is always the destination posture of the previous scene. If the frame is 1, only one data, which is the destination posture, is sent to wCK modules. If the frame is 10, ten more data between the start position and destination position are sent to wCK modules. Therefore, the more frame you set, the more precise the motion will become. On the other hand, the less frame you set, the more rough the motion will be. If you want to make smooth motions, it is better you increase the frame. If you want to make dynamic motions, it is better you decrease the frame.

2) If the frame is one, the transition time is minimum 20ms.

3) The transition time for one scene ranges from 20 up to 6000ms.

4) As frame and transition time are inter related, you can not increase the frame without limit under a certain transition time. Or you can not change transition time without limit under a certain frame number.

3.2.9. **MotionBuilder Advanced Function (2) : Torque**

In this chapter, it describes Robot module (wCK)’s torque level adjustment function.

1) A wCK module has 5 levels of torque under 8 bit control mode. Level 0 is the strongest and level 4 is the weakest.

2) Default value for torque is set to 2.
3) If you increase torque, wCK module use more energy and result in faster response time and higher rotation force. On the other hand, if you decrease torque, wCK module use less energy and result in slower response time and lower rotation force.

4) Generally speaking, higher torque is used for scenes that require fast and strong movement, and lower torque is used for scenes that require slow and smooth movement.

3.2.10. **MotionBuilder Advanced Function (3) : Control Gain**

This chapter explains how to adjust the Runtime Gains of wCK module and use for advanced motion building.

1) P gain, D gain, and I gain are used for the control algorithm of wCK module. Users can adjust these values in order to acquire the response characteristics for a specific purpose. These gains can be changed on runtime operation without writing to the EEPROM.

2) The runtime gains can be differently assigned for a motion file. In other words, users can change the motor output for a whole motion file while adjusting torque is used to affect each scene.

3) The gain values can be checked in the ‘New Motion’ window or the ‘Set Motion’ window.

4) P gain is Proportional Gain coefficient. This gain is applied to the output after being multiplied by the control error. Therefore, the higher the P gain is, the stronger the motor output becomes. But the maximum output of the motor is limited, too much P gain doesn’t necessarily generate higher power. Generally from 20 to 40 is recommended for P gain.

5) D gain is Differential Gain coefficient. This gain is applied to the output after being multiplied by the change rate of the control error. Therefore, it affects the overshooting of the output. Generally two times the P gain is recommended for D gain.

6) I gain is Integral Gain coefficient. This gain is applied to the output after being multiplied by the integral of the control error. Therefore, it has an effect of compensating the integral error. In case that I gain is high, the system regist to a strong external force, which may cause damage to the system. Therefore, generally zero(0) is recommended for I gain.

7) The gains explained here as integer value are the values that are used inside the wCK module and are not to be applied as PID gains for other control systems.
3.2.11. Creating Action File (1)

In this chapter, user creates HUNO actions as the below.

Basic Posture => Move Forward => Turn Left => Move Right => Turn Right => Attack Left => Move Backward

User also can create action files with other robot platforms.

※ Please refer to the <Motion & Sound List> for Motion & Sound No.

1) Click New in menu bar. Input Action Name/File Name, and select Robot Platform, then click OK
(Example) Action Name: Huno motion test, File Name: mtest.rba, Platform: Creator HUNO)

2) Add button is activated after new action file (Creator HUNO type) is created.
3) Select “None” in the CONDITIONS[If], and select “Motion Out – MOTION[7]:BTN_C” in the EXECUTIONS [Then], and change Statement Name and Description as you want, then, click Add.
(Example: Run “Basic Posture” without any conditions.)

4) Select “None” in the CONDITION[If], and select “Motion Out – MOTION[4]:BTN_U” in the EXECUTIONS [Then], and change Statement Name and Description as you want, then, click Add.
(Example: Run “Move Forward” motion)

5) Select “None” in the CONDITION[If], and select “Motion Out – MOTION[3]:BTN_LR” in the EXECUTIONS [Then], and change Statement Name and Description as you want, then, click Add.
(Example: Run “Turn Left” motion)
6) Select “None” in the CONDITION[If], and select “Motion Out – MOTION[8]:BTN_R” in the EXECUTIONS [Then], and change Statement Name and Description as you want, then, click Add. (Example: Run “Move Right” motion)

7) Select “None” in the CONDITION[If], and select “Motion Out – MOTION[5]:BTN_RR” in the EXECUTIONS [Then], and change Statement Name and Description as you want, then, click Add. (Example: Run “Turn Right” motion)

8) Select “None” in the CONDITION[If], and select “Motion Out – MOTION[9]:BTN_LA” in the EXECUTIONS [Then], and change Statement Name and Description as you want, then, click Add. (Example: Run “Left Attack”)
9) Select “None” in the CONDITION[If], and select “Motion Out – MOTION[10]:BTN_D” in the EXECUTIONS [Then], and change Statement Name and Description as you want, then, click Add. (Example: Run “Move Backward” motion).

10) Save Action File in Menubar. (예 : mtest.rba)

11) With same procedure that is described (Chapter 2,3,2) in the above, user can download and run Action file with remote control.
3.2.12. Creating Action File (2)

In this chapter, user creates action file that robot do "Posture => Idle Status => Attack followed by some sound from outside".

※ Please refer to the <Motion & Sound List> for Motion Out No.

1) Select **New** in the menu bar. Input "Action Name, File Name, Robot Platform, ..etc" in new action window. (Ex) Action Name : test, File Name : test.rba, Platform : Creator HUNO)

![Image of New Action Window]

2) **Add** button is activated after Creator HUNO type action file is created.

![Image of Add Button Activated]

3) Click “None” in CONDITION [If], and Click “Motion Out – MOTION[7]:BTN_C”, then change Statement Name & Description. Click Add button. (Ex: Run “Basic Posture” without any condition)

4) Click None in CONDITION [If], and Click “Wait Time, 2000” in EXECUTIONS [Then], then change the Statement Name & Description as you want. Click Add. (Example: Idle Status without any condition)
5) Click “Sound In, 15 < Sound Level” and Click “Motion Out, MOTION[11]:BTN_RA”, then change the Statement Name & Description as you want. Click Add. (Example: Attack Right, if sound input level is more than 15.)

![Image of RoboBuilder User Tutorial](https://example.com/robobuilder.png)

6) Click “None” in CONDITION [If], and “Jump Index, 2” in EXECUTIONS [Then], then change Statement Name & Description as you want. Click Add. (Example: Move to Index 2)

![Image of RoboBuilder User Tutorial](https://example.com/robobuilder.png)

7) Click “Save” in the menu bar (Example: test.rba)

![Image of RoboBuilder User Tutorial](https://example.com/robobuilder.png)
3.2.13. Using 3 axis Acceleration Sensor (Option) (Example : test2.rba)

Below example is described that Robot automatically stands up if Robot fall down in front and rear side. Robot recognize the front/rear falls down itself, and A button function (remote control) is proceeded automatically for front falls down, and B button function (remote control) is proceeded automatically for lying down.

※ Acceleration Sensor Direction / Data

Measured X, Y, Z axis value is indicated by integral number from −7 to +7. Integral number 1 can be translated as “1/4g”. Note) g : gravity acceleration

For instance, Value would be X=0g, Y=0g, Z=−1g if Robot lies down, because all gravity acceleration direction is minus(−). Therefore, you can judge whether Robot is lies down or not, as long as you check acceleration data would be −1< X < 1, −1 < Y < 1, −5 < Z < −3 (Please refer to Figure 5-1)

※ NOTE

In order to Tri–axial acceleration sensor,

1. Equip the Tri–axial acceleration sensor board (RBX–ACL3A01).
2. Upgrade RBC (Control Box) firmware to Ver. 2.09

1) Click [New] in the menu bar. Input “Action Name, File Name, Robot Platform” (Example : Action Name : test2, File Name : test2.rba, Platform : Creator HUNO)
2) Add button is activated after Creator HUNO type action file is created.

3) Click “None” in CONDITIONS [If], and click “Motion Out – MOTION[7]:BTN_C” in EXECUTIONS [Then], then, change Statement & Description as you want. Click Add button. (Example : Run “Basic Posture" without any condition.)

4) Click “Accel., $-1 < X < 1, -1 < Y < 1, 2 < Z < 6$ in CONDITIONS [If], and click “Motion Out, MOTION[1]:BTN_A” in EXECUTIONS [Then], then change Statement Name & Description as you want. Click Add button. (Example : Run “Stand Up, if robot falls down” => Remote control “A" function)
5) Click “Accel., -1 < X < 1,  -1 < Y < 1,  -6 < Z < -2” in CONDITION [If], and click “Motion Out, MOTION[2]:BTN_B” in EXECUTIONS [Then], then change Statement Name & Description. Click Add button. (Example: Run “Stand Up, if robot falls back side” => Remote control “B” function)

6) Click “None” in CONDITION [If], and click “Jump Index, 1” in EXECUTIONS [Then], then change Statement Name & Description. Click Add button. (Example: “Go to Index 1”)

7) Click Save button to save Action File (Example: test2.rba)

8) Please refer to the chapter 2.3.2 for Action file downloading and testing.
3.3. Advanced Course

In this course, users learn MSRDS (Microsoft Robotics Studio 2008) setup/configuration and VPL programming tool in order to control RoboBuilder.

Figure 1. Robot System Structure by using MSRDS based on Computer
3.3.1. What is MSRDS 2008?

MSRDS stands for Microsoft Robotics Developer Studio, and is robot development software solution. MSRDS is widely used for various robot platforms, and easy to develop various robot application by using VPL programming tool. User can build and study from simple and beginner program to complicate algorithm.

3.3.2. MSRDS main feature

① Provided GUI programming environment : Visual Programming Language (VPL)

② Provided integrated solution with Visual Studio : C#, VB.Net, C++.Net, Python

③ Provide simulation tool.

④ Provided various sample and documents in MSRDS website and communities.
3.3.3. MSRDS 2008 Installation

In this chapter, it describes how to install MSRDS 2008 Express in **Windows XP**.

① **Download Setup file**

Please download Microsoft Robotics Developer Studio 2008 Express Edition (Offline installation).exe in the below link.


③ **Click “install” button**, and later it shows re–boot message box, then, click “Yes” for re–booting.

④ **After re–boot**, click “install” again for next steps.
⑤ Click “Next”, and select “I accept the terms in the license agreement”, then click “Next” again.

⑥ Click “Next” and click “Install” in next step.

⑦ Click “Finish” then, it completes MSRDS 2008 Express installation.
3.3.4. MSRDS 2008 – RoboBuilder service module installation

In this chapter, it describes how to install RoboBuilder service module for MSRDS 2008 in Window XP.

① Click MSRDS2008_RoboBuilder_service–install.exe

② Click “Next”.

③ Select Icon creates, then, click “Next”, and click “Install” again.

Click “Finish” for MSRDS 2008 RoboBuilder service module setup.
3.3.5. MSRDS 2008 - VPL Programming

In this chapter, it describes how to control HUNO RoboBuilder by using MSRDS 2008-VPL Programming tool.

① Plug power-adaptor into RoboBuilder and connect with PC by using PC cable, then power-on a RoboBuilder power switch.

※ RBC Box firmware version should be ver 2.15 or higher in order to support MSRDS 2008. (Regarding firmware upgrade, please refer to the Kit user guide or website FAQ.)
② Run Microsoft Robotics Developer Studio 2008 Express–Visual Programming Language 2008 Express

③ Double-click “RoboBuilderBrick” at Services section in the following screen, then RoboBuilderBrick would be registered.

④ Click RoboBuilderBrick activity in the diagram, then, select “Set Initial configuration” in the Configuration section in the right bottom.
⑤ Input proper COM PORT No. in SerialPort. For instance, if RoboBuilder use COM 1, then input “1”.

⑥ Double-click or Drag a “Direction Dialog” in Services section. If Direction Dialog is overlapped then, move this activity module.

⑦ In similar way, add “If” 1 module, “Data” 6 modules, “Merge” 1 module in Basic Activities section.
⑥ Drag the brown circle of DirectionDialog to “If” module as shown in the below.

⑨ Select “ButtonPress” in Connections as shown in the below, then click “OK” button for connection.

⑩ Input Name == “Forwards” in “If” module, then, click + button in order to add the rest of Names as show in the below.

⑪ Drag the brown triangle of “Merge” module to “RoboBuilderBrick”, then select “MotionControl” in the Connections, and select “value” in the Data Connections as shown in the below.
⑫ In similar way, input the following Nos. in “Data” module, then connect between activities as the followings.

⑬ Click “▶” button or press “F5” in the keyboards.

If this program is never saved before, then input file name, and click “Save”.
It shows RoboBuilder control program as the shown in the below after RoboBuilder pose basic posture.

You can control RoboBuilder by clicking direction button.
- ▲ button : Move Forward
- ◄ Button : Move Left
- ► Button : Move Right
- ▼ Button : Move Backward
- Stop : Basic Posture

※ RoboBuilderBrick Data Value reference
1) MotionControl : Refer to Motion No. in the appendix.
2) SoundPlay : Refer to Sound No. in the appendix.
3) The rest input commands are NOT supported at the present.
3.3.6. VPL Programming – Variable declaration and usage

In VPL tool, variable declaration is usable just like other programming language. In this chapter, user declare various style variable, and show how to use the variable properly.

① Let’s declare variable first. Double-click “Variable” activity in Basic Activities, or Drag & drop to diagram window. You can see “...” icon in that activity, click this icon, then designate variable type and name.

![Figure 1](image)

② Click “Add” button and designate String type “myName”.

![Figure 2](image)

![Figure 3](image)
As the above figures, variable “myName” is declared. Then, we will assign the value into this variable. In order to assign a value into variable, use Data activity. Double-click “Data” activity in the Basic Activities window, or Drag & drop at left side of myName activity as shown in the figure 4.

![Figure 4](image)

Make the variable type to “string” and input your desirable string into textbox. We have input “David” in this example.

![Figure 5](image)

Connect the right orange point of Data activity with the left orange point of Variable activity. Then, new window is shown in Figure 6.

![Figure 6](image)

In the above window, select SetValue in the “To” items. Basically, Variable reads and configure the value. In this example, we show how to save the value into Variable, therefore, select “SetValue”. Then, OK.
Let's check out “myName” variable whether it has been assigned value. In order to check certain variable value, select Simple Dialog activity in the Services list.

To find in a short time, input “Simple” in the Service list search ("Find service…") text box. In this case, all service that included “Simple” word is shown.

Simple Dialog conducts as a dialog function in general existed program. Simple Dialog provides variable value showing, value receiving, OK or Cancel value from user.

Locate Simple Dialog activity the right side of Variable activity, then connect the right orange point of variable with the left point of Simple Dialog.

Connections window is shown as Figure 8. In this window, AlertDialog shows incoming value. PromptDialog receives keyboard input value, ConfirmDialog receives OK & Cancel value. In this example, we use Simple Dialog function, therefore, select AlertDialog, and click OK.

Then, Data Connections window is shown as Figure 9.
Select `myName` in combo box. To do so, `myName` value is delivered to `AlertText` of Simple Dialog.

After click OK button, connected diagrams are shown as figure 10.

If user wants to re-connect between activities or revise values, point a mouse in line between connections, then click the right button of mouse. Pop-up menu is shown for Delete, Connections, Data Connections. User can check or revise mapping relationship by selecting Data Connections.

Save the program, then click the Run icon.
Program project status is shown in the Run window.

![Run window showing project status](image)

**Figure 13**

Alert Dialog window is shown, and incoming value is displayed.

![Alert Dialog window](image)

**Figure 14**

To exit, click “Stop” button in Run window.
3.3.7. VPL Programming – Process Variable Value

In this chapter, user learn how to changes the value of variable.

① Let’s show “Your name is David” in the display. First of all, remove the connection between Variable activity and Simple Dialog activity.

![Figure 1](image1.png)

② Select Calculate activity in Basic Activities, and add it between Variable activity and Simple Dialog activity.

![Figure 2](image2.png)

User can use various four arithmetical calculations by using Calculate activity.

If user click on the Calculate activity, it shows available variables automatically.

![Figure 3](image3.png)

③ Next job is connection between Variable activity and Calculate activity. At this time, no other Connection window is now shown. After connection, “myName” is added if Calculate activity is activated.

![Figure 4](image4.png)

④ Select “myName”.

![Figure 5](image5.png)

⑤ Now, connect Simple Dialog activity. At this time, Connection window is displayed automatically.

Select AlertDialog and click OK.
⑥ Click Value-value item, and click OK in Data Connections window.

⑦ Let's run the program. After click Run, “David” value is saved into myName, then, this value is delivered to Simple Dialog through by Calculate activity.
8. Above example is just showing delivered value. Now, user add strings.

Activate Calculate activity, and input strings as belows.

![Diagram of RoboBuilder User Tutorial](image9.png)

Figure 9

9. Let's Run. As expected, Data activity value is added with Calculate activity string.

![Diagram of RoboBuilder User Tutorial](image10.png)

Figure 10
3.3.8. VPL Programming – Process Input Value (1)

In this chapter, let’s learn how to display input value.

① In order to make this program, user use two Simple Dialog activity and one Calculate activity. Calculate activity can be used for calculating input value, and Simple Dialog activity read strings.

First of all, add Simple Dialog activity, and Calculate activity. Connect output point of Simple Dialog with input point of Calculate activity. At this time, select PromptDialog – Success item in From:

![Figure 1](image1.png)

② After connection, activate Calculate activity input box, then it shows selectable values. Select TextData variable.

![Figure 2](image2.png)

③ Add Simple Dialog activity again. Since the first Simple Dialog is already added, user is asked whether this activity to use as a common purpose, or adding separately with new Simple Dialog activity name.

![Figure 3](image3.png)
In this example, user can use the Simple Dialog as a common purpose, select the second item “SimpleDialog” then, click OK. In this case, we selected to re-use activity, however, another SimpleDialog activity is added in the display.

5) Connect output point of Calculate activity with input point of added SimpleDialog. You will see the Connections window.

6) In order to display the received value onto the display, select AlertDialog item. Data Connections window is shown. Select value item.

Completed diagram is as belows.

7) Input strings when program runs.
User can see the string value as received the value from keyboard.

Let's add more strings at this time.

Input strings “My Name is” in the front of TextData, and “.” the after TextData varaible as belows.

After configuration as Figure 9, user gets following result as belows.
### 3.3.9. VPL Programming – Process Input Value (2)

In this chapter, it shows how to program the plus calculation. Join activity is used for this calculation. This activity receives two input and process it, then forward the value to the activity.

1. First of all, add two Data activity, then input int type “10” and “20” in text box.

2. Add Join activity in Basic Activities items.

3. Connect these two output point with input point of Join activity. No other window is shown.

4. Add Calculate activity. Connect output point of Join activity with input point of Calculate activity. No other window is shown. After connection, click Calculate activity text box. Then it shows available value or variable list.

5. Revise Calculate activity text as belows.
6. After adding SimpleDialog activity, connect the output point of Calculate activity with input point of Simple Dialog activity. At this time, Connections window is shown. Select AlertDialog.

![Figure 5](image1)

7. Select “Value-value” in Data Connections window.

![Figure 6](image2)

Completed diagram is as follows.

![Figure 7](image3)

8. Run program, the result is shown as Figure 8.

![Figure 8](image4)
3.3.10. **VPL Programming** – IF condition

In this chapter, it shows how to use IF activity. In order to test “IF” scenario, receive the two values, then compare the value. If two values are same, it shows “Same”, if not, it shows “Different” message.

1. First of all, add two SimpleDialog and two Calculate activity in order to receive two strings. Then, connect each activity. To receive the values, select “PromptDialog-Success” and connect.

![Figure 1](image1.png)

2. Select TextData in Calculate activity text box.

![Figure 2](image2.png)

3. Then, add Join and two Data activity, then connect these as follows.

![Figure 3](image3.png)

4. Add two SimpleDialog activity, then connect two Data activity. At this time, set to AlertDialog.
5 Completed diagrams are as follows.

6 Run program, and check the result.
3.3.11. VPL Programming – Switch condition

In this chapter, it shows how to use Switch activity.

To test this example, add Switch activity.

Figure 1

In the above diagram, Merge activity was used for merging the various data flow.

Merge activity can have various input data. Any input data forward to connected activity.

Merge activity deliver the value even if just one data arrives while Join activity deliver the value if all input value arrives.
3.3.12. VPL Programming – Repetition statement

In VPL, user can use repetition function by using If and Merge activity. The below diagram is repetition example that is 10 times increase the value by plus and repeats.

User can apply this kind of repetition program in other way. (Example : Repeat this program pattern until the initial value approaches certain value by repetition.)

Below program repeats until user input “Stop”.

![Figure 1](image1.png)

![Figure 2](image2.png)
4. Appendix

4.1. Appendix1 - FAQ

<table>
<thead>
<tr>
<th>Title</th>
<th>What is wCK module?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>RoboBuilder’s wCK module is an intelligent robotic module with which users can build creative robots of various shapes and easily operate and control robots. It is the first block type robotic module in the world that has a joint insertion assembly structure. This quick and simple joint assembly scheme enables users to simply plug a plastic joint part into wCK modules to mechanically link two different wCK modules, which dramatically helps reduce building time. Internally a control board and a servo actuator mechanism are integrated together within the small plastic enclosure. The wCK module itself can operate as a small independent robot system because it is equipped with external I/O ports and can run a self-running motion program. The wCK module adopt a PID motion control technology and realized motion control characteristics as precise and accurate as industrial servo motors. Users can design and build robotic systems with multi-axis articulated mechanism much more easily and efficiently by adopting wCK robotic modules. This is because the wCK module is designed for users to easily extend functionality and to fast track troubleshooting and maintenance.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>How can I build Robot program?</th>
</tr>
</thead>
</table>
| Description | At present, RoboBuilder provides 3 ways for robot programming.  
1. MotionBuilder, ActionBuilder programming (Refer to download section in website)  
2. C programming (Refer to tips for specialist)  
3. MSRDS programming (Using VPL tool– Refer to download section in website)  

No. 1 is RoboBuilder exclusive programming method. This is most wisely used way. Please refer to the download section in RoboBuilder website for recent software version.  
No. 2 is for computer programmer or expert who has good knowledge of computer hardware.  
No. 3 is GUI programming based that using MSRDS (Microsoft Robotics Developer Studio) Visual Programming Language. Please refer to the download section in RoboBuilder website for tutorial. |

<table>
<thead>
<tr>
<th>Title</th>
<th>How can I adjust joint angle when assembling?</th>
</tr>
</thead>
</table>
| Description | If you are not clear the RoboBuilder User Manual,  
First, connect RBC Box with Head module and all wCK module. It is parallel connection. Therefore, connection order is not a problem.  
Second, set the control box platform as you want to assemble to HUNO, DINO, DOGY  
Third, press remote control red button (basic posture), then wCK angle is adjusted automatically  
Fourth, now, disconnect all cable, then assemble according to User Manual.  
This way makes to assemble very easily as all the wCK module is adjusted in advance  
※ Please refer to the assembling movie–clip at download section in RoboBuilder website. |
<table>
<thead>
<tr>
<th>Title</th>
<th>What is the check point when Error LED is blinking during basic posture?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>After assembling with standard platform (HUNO, DINO, DOGY) completely, robot takes basic posture when you press remote control red button (■). At this time, there would be two symptoms if there were errors.</td>
</tr>
<tr>
<td></td>
<td>1. No movement, just error LED blinking.</td>
</tr>
<tr>
<td></td>
<td>2. Move and error LED blinking.</td>
</tr>
<tr>
<td>Description</td>
<td>No. 1 error is wCK communication error. In this case, user should check wCK cable again.</td>
</tr>
<tr>
<td></td>
<td>1) Check wCK cable connection with RBC box and try again.</td>
</tr>
<tr>
<td></td>
<td>2) Use &lt;RoboBuilder Diagnostic Tool&gt; in order to check fault cable.</td>
</tr>
<tr>
<td></td>
<td>No. 2 error is usually joint angle is not adjusted well. In this case, wCK cable is taken out during movement.</td>
</tr>
<tr>
<td></td>
<td>1) Disassemble the wCK and press the red button (■). Then assemble it.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Can I buy Robot part separately?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Yes, you can buy it from your local distributor.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>HUNO keep falling down. What is wrong?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. Wrong assembling</td>
</tr>
<tr>
<td></td>
<td>1) Check the shoulder angle (wCK ID No. 10, 13)</td>
</tr>
<tr>
<td></td>
<td>2) Check wCK ID No. 0, 5, 4, 9 with B8 screws.</td>
</tr>
<tr>
<td></td>
<td>3) B8 with wCK ID No. 0, 5, 4, 9 is not screwed well.</td>
</tr>
<tr>
<td></td>
<td>4) Shoe Plate is assembled in opposite way.</td>
</tr>
<tr>
<td></td>
<td>5) Knee joint (J6, ID No. 1<del>2, 6</del>7) is assembled in opposite way.</td>
</tr>
<tr>
<td></td>
<td>6) wCK cable got tangled.</td>
</tr>
<tr>
<td></td>
<td>7) Head module is assembled in opposite way.</td>
</tr>
<tr>
<td></td>
<td>2. Other reason</td>
</tr>
<tr>
<td></td>
<td>1) Floor is not flat.</td>
</tr>
<tr>
<td></td>
<td>2) Battery is not charged enough.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>How can I charge the battery?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>1. Connect adapter with RBC Box.</td>
</tr>
<tr>
<td></td>
<td>2. Press “#” button and “A” button in remote controller at the same time, or Press “PF1” and “PF2” button in RBC at the same time.</td>
</tr>
<tr>
<td></td>
<td>3. Green POWER LED is blinking when charging started.</td>
</tr>
<tr>
<td></td>
<td>4. When Green POWER LED stops blinking, charging stops.</td>
</tr>
<tr>
<td></td>
<td>Operating time is depends on what kind of movement is performed. Usually, 20~30 minutes moves continuously according to specification.</td>
</tr>
</tbody>
</table>
### How can I upgrade RBC firmware?

1. Run upgrade software (RBC Upgrade Tool), turn on the power of RBC.
2. Connect RBC with PC with RS232 cable and set Com Port and set Baud Rate of 115200 bps.
3. Click folder icon to select firmware file (*.hex) to upgrade and click open.
4. Press "Click here and Push Button" and it turns to standby mode.
5. Use a sharp pen to push the reset button of RBC (hole between PF1 button and PF2 button).
6. Proceed the upgrade.
7. When finished RBC, it reset automatically and complete message appears, then, Click OK.
8. If fails, try again from step1.

---

### Is it possible to control RoboBuilder without RBC Box?

**Description**

By using PC serial COM port, it is possible to control communication (RS-232=>TTL Level). RoboBuilder protocol reference can be found in RoboBuilder website.

### What is Firmware?

**Description**

FIRMWARE is for controlling RoboBuilder Kit hardware-software, this is saved in RBC Box internal ROM. In order to add the function or improve/fix the bug, user upgrades the firmware.

### How many Robot files can be saved?

**Description**

In RBC firmware version 2.15, User can download & save 20 motion files (*.rbm) in max. If file size are big, total downloadable files can be decreased.

**Note**

Once RoboBuilder power off and power on, existed Motion files are deleted if user starts download again. Therefore, user should not off the power in order to add the motion files.
### What is difference between Motion and Action?

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
</table>
| Motion is robot’s motion or movement. This can be made by MotionBuilder software, and it is “.rbm” extension file. In case of DOGY robot, various motions can be made as below. | Motion 1] Shake the right front foot.  
Motion 2] Nod the head  
Motion 3] Hand Stand  
Motion 4] Move forward and go left  
Motion 5] Turn right …… |
| Action is the combination of various motion factors, motion factors are motion, hearing, speaking, detecting object… etc. ActionBuilder is using simple if… Then… logics and makes motions continuously. In case of DOGY robot, various actions can be made as below. | Action 1] Bark when hear clap sounds, and left turn, and go forward.  
Action 2] Go forward, if detects object, move backward.  
Action 3] Do dance when hear clap sound, and stops when hear clap sound again. |

### Is it possible to build different robots except stand robot platform?

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes. Of course, any kind of robot can be made by user. Using various joint and other parts, try to build various robots. If two users made the robot with same ID map, they can share the robot file through RoboBuilder website.</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>What is zero position posture adjustment?</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>1. What is zero position posture adjustment?</td>
<td>In order to know each robot posture, this is standard posture information.</td>
</tr>
<tr>
<td>2. Why zero position needs?</td>
<td>By this, motion files can be shared between users. Every servo motors has tolerance with others. Therefore, zero position function helps A robot’s motion works into B robot’s motion.</td>
</tr>
<tr>
<td>3. How can I set zero position of robot?</td>
<td>After assemble HUNO type, put into RoboBuilder Box Tray, then make the zero position. (Please refer to the User Manual page 87)</td>
</tr>
<tr>
<td>4. Is zero position setup necessary?</td>
<td>RoboBuilder wCK module has very small tolerance (<del>0.8</del>+0.8 degree), therefore, it is not need in general situation.</td>
</tr>
<tr>
<td>5. Others</td>
<td>Zero position program tool is available in RoboBuilder website. User can do this function without RoboBuilder Box Tray.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>What is the PC control mode of Control Box?</th>
</tr>
</thead>
<tbody>
<tr>
<td>In order to control the ROBOT movement, you have to send the Command to wCK module.</td>
<td>The easiest way to send the Command to wCK module is using Control Box. There are two ways using Control Box.</td>
</tr>
<tr>
<td>1. Control Box, itself generates Command</td>
<td>Example) Download the Motion File, Action File, then RUN.</td>
</tr>
<tr>
<td>2. Connect the Control Box with External device (PC, PDA, ... etc), then this external device makes the Command, and send it to Control Box. Example) Run Motion in MotionBuilder. Configure wCK parameter in wCK Programmer.</td>
<td>No.2 is PC control mode. PC control mode is run automatically in MotionBuilder &amp; Self Diagnostic Tool Program. For manual way to use the PC control mode, press PF2 Button in Control Box, then PF1 (Blue LED) and PF2 (Orange LED) will be light up.</td>
</tr>
</tbody>
</table>
4.2. Appendix 2 – How to replace the damaged wCK module gear

1. Unscrew 5 bolts by using "V" screw driver.
2. Put '+' screw driver into No. 4 Gear as shown in the figure.
3. Press the '+' screw driver to the down side as shown in the figure by using screw driver, then detach the Top case.
4. Take out the gears to upside as shown in the figure. Then detach the bottom case.
5. Put grease on the No. 4 Gear, then assemble as shown in the figure.
6. Assemble the bottom case as shown in the figure. Insert 3 gear pins into the housing, then, put some grease on the No. 3 Gear’s teeth.
7. After put some grease on the No. 2 Gear’s teeth, assemble it as shown in the figure.
8. After put some grease on the No. 1 Gear’s teeth, assemble it as shown in the figure.
9. Assemble the top case, and screw the 4 bolts as shown in the figure. Lastly, screw the small bolts for top case fixing.

Some wCK modules do not have fixing screw bolt in this position. It is totally OK even if you do not assemble it. And it is better not to screw it so tightly.

Be sure to match the half-round shape between sensor and No. 1 Case, then, put it in accurately. If you put the gear strongly, sensor could be damaged.
4.3. Appendix 3 - wCK module Communication Protocol
4.4. Appendix 4 – How to change wCK ID

wCK module in RoboBuilder Kit is configured own ID from 0 to 15. But you may change the wCK ID as you want to do from 0 to 30. This document describes how to change wCK ID. With this tip, you can change other wCK parameter in similar way.

Requirements
① wCK
② wCK Cable
③ Control Box(RBC)
④ PC Cable
⑤ Power supply adaptor
⑥ Windows XP OS PC
⑦ wCK Programmer S/W

Procedures (Change wCK ID from 2 to 0)

1) Connect wCK, Control Box, Power Supply Adaptor, with PC

※ It would be no problem to connect one wafer in wCK with any one wafer in Control Box.

※ NOTE : Make sure to connect only one wCK when you want to change wCK ID.
However, you may connect more than one wCK if you change other parameters except ID.
2) PF1 LED (Blue), PF2 LED (Orange) lights because the Control Box become PC control mode when you power on after you pressed PF2 button as shown in the below figure.

※ In wCK Programmer ver 1.34 or above version, pass step2.

3) Run wCK Programmer in PC, then select BasicSetting Tap.

4) Match COM Port with RBC Box connected COM Port, then click Scan Baud button, it shows “wCK configured comm. speed” in Scan Baudrate.

※ If comm. speed is not shown, and just show “Try Again” message, check out the connection between wCK cable and PC cable.
5) Configured ID is shown when Scan ID button is clicked.

6) Select to be changed ID, then click “Set ID” button, it shows “Good ID Setting!!” message. Then it shows newly changed ID.

7) Completed wCK ID.
4.5. Appendix 5 – 3 HUNO Arm Structures

Representative 3 types of HUNO Arm Structure

- **Walking Type**: This arm type has the advantage of walking, and basic structure for RoboBuilder HUNO.
- **Dance Type**: This arm type has the advantage of various arm movement like Dancing.
- **Transformer Type**: This arm type makes to transform the HUNO into Wheel-Type Robot.

1) Walking type arm structure

2) Dance type arm structure

3) Transformer type arm structure
How to Change from Walking type to Dance type

1. Unscrew Bolt & Nut, then, disconnect the Joint 1 from wOK 11.

2. The palm of the hand towards front side if you change wOK 12 to Dance type.

3. Connect Joint 1 as shown in the figure.

4. If you want to change the direction of palm of the hand, unscrew the bolts & nuts. Then, change the direction after disconnect the hand. And assemble it.

5. This is the example that the palm of the hand towards body.
How to change from Walking type to Transformer type
4.6. Appendix 6 – C Programming with Motion File

This chapter explains how to create user’s own firmware to control RBC controller by including a motion file (xxx.rbm) in C programming instead of using RoboBuilder’s standard firmware.

4.6.1. Material included

1) Example Motion File
   ① Project file: p_ex1.prj
   ② Motion file: m_ex1.rbm

2) Example Source C code (CodeVisionAVR 1.24.8d)
   ① Project file: cv_ex1.prj
   ② Unit file: main.c, comm.c, dio.c
   ③ Header file: main.h, comm.h, dio.h, macro.h, m_ex1.h

1) This material can be downloaded from the ‘Tips for Specialist’ page on the homepage.

4.6.2. Caution

1) This function is supported only with MotionBuilder version 1.10 beta or higher.

2) C code is based on CodeVisionAVR 1.24.8d.

4.6.3. Work Flow

1) Convert a motion file to a header file (*.h)
   ① Open MotionBuilder (version 1.10 beta or higher)
   ② Click [Open] button to open [p_ex1.prj] file in [motion_exam] folder
   ③ Click [Motion List] button

2) Include the header file into C coding

3) Generate an executable hex file

4) Download to RoboBuilder and run
Set the “Header File Format” as “ME_FMT#1” (default) and push the Create Header File button.

Assign a name for the header file (*.h) and push [Save] button. For example, a name “p_ex1.h” was used in this example.
A pop-up window appears and ask if you want to read the generated header file. Choose as you wish.

Header file generation is completed.

2) Include the header file(*.h) into C coding

1) Move the generated file "p_ex1.h" into the folder "cv_exam/src". A file with identical name will be overwritten.

2) Use CodeVisionAVR and read "cv_ex1.prj".

3) Edit "comm.c" so as to match with the name as below.
   ```c
   #include "p_ex1.h"
   ```

4) Change the array names to match with the motion file in “SampleMotion1” function in “comm.c” (use capital letter only) For example, if the motion name is 'M_EX1'
   ```c
   gpT_Table = M_EX1_Torque;
gpE_Table = M_EX1_Port;
gpPg_Table = M_EX1_RuntimePGain;
gpDg_Table = M_EX1_RuntimeDGain;
gplg_Table = M_EX1_RuntimeIqGain;
gpFN_Table = M_EX1_Frames;
gpRT_Table = M_EX1_TrTime;
gpPos_Table = M_EX1_Position;
Motion.NumOfScene = M_EX1_NUM_OF_SCENES;
Motion.NumOfwCK = M_EX1_NUM_OF_WCKS;
   ```

5) Header file registration is completed

3) Generate an executable hex file(*.hex)

1) Use CodeVisionAVR and run “Project – Make” menu or push Shift+F9.

2) Executable hex file generation is completed

4) Download to RoboBuilder and run

1) Connect RoboBuilder with PC and turn it on. (connect power supply too).

2) Run the [RBCUpgradeTool] and set Com port accordingly.

3) Set the 'Firmware File' to the "main.hex" file which was generated in "cv_exam/src" folder.
④ Push the 'Click here and Push Reset Button' button. Then RoboBuilder wait for you to push the reset button.

⑤ Push the Reset button (the hole between PF1 button and PF2 button to start the firmware upgrade.

⑥ When download is completed, the message of 'Flash File successfully downloaded.' appears.

⑦ Disconnect RoboBuilder from PC and push PF1 button to run the motion

⑧ All procedures are completed.
4.7. Appendix 7 – RoboBuilder Remote Control Command

In this chapter, it shows how to control the RoboBuilder in remote.

1) Communication Standard

Use series asynchronous communication way for RoboBuilder RBC Box “RBC-08128XXX”.

① Baud rate : 115200bps
② Data bits : 8
③ Stop bits : 1
④ Parity : None

2) Communication Channel

There are two channels that can receive the commands.

① Cable connection

<table>
<thead>
<tr>
<th>Pin No</th>
<th>Pin Name</th>
<th>Direction</th>
<th>Description</th>
<th>Signal Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td></td>
<td>Power Ground</td>
<td>Ground</td>
</tr>
<tr>
<td>2</td>
<td>VDD</td>
<td>Input</td>
<td>DC input(3.0~3.3V)</td>
<td>Power</td>
</tr>
<tr>
<td>3</td>
<td>Status</td>
<td>Output</td>
<td>Status</td>
<td>TTL</td>
</tr>
<tr>
<td>4</td>
<td>RST</td>
<td>Input</td>
<td>Reset(Active Low)</td>
<td>TTL</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>TxD</td>
<td>Output</td>
<td>UART data output</td>
<td>TTL</td>
</tr>
<tr>
<td>8</td>
<td>RXD</td>
<td>Input</td>
<td>UART data input</td>
<td>TTL</td>
</tr>
</tbody>
</table>
3) Communication Packet Structure

Likewise RoboBuilder IR remote controller, PC or other controller can send the commands to RBC box.

Remote controller device ▶▶▶ RBC Box

| 0xFF     | 0xFF | 0xAA | 0xAA | 0x55 | 0x55 | 0x37 | 0xBA | 0x14 | 0x00 | 0x00 | 0x00 | 0x01 |

RBC Box ▶▶▶ Remote controller device

| 0xFF     | 0xFF | 0xAA | 0xAA | 0x55 | 0x55 | 0x37 | 0xBA | 0x14 | 0x00 | 0x00 | 0x00 | 0x01 |

※ Command List : Refer to the Appendix 8.
## 4.8. Appendix 8 – IR Remote Controller Command Code List

Following list can be used when user want to use the remote control commands or C programming.

<table>
<thead>
<tr>
<th>Button Code</th>
<th>Button Name</th>
<th>Command Code</th>
<th>Button Code</th>
<th>Button Name</th>
<th>Command Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>BTN_A</td>
<td>0x01</td>
<td>F</td>
<td>BTN_F</td>
<td>0x16</td>
</tr>
<tr>
<td>0</td>
<td>BTN_B</td>
<td>0x02</td>
<td>G</td>
<td>BTN_G</td>
<td>0x17</td>
</tr>
<tr>
<td>1</td>
<td>BTN_C</td>
<td>0x03</td>
<td>H</td>
<td>BTN_H</td>
<td>0x18</td>
</tr>
<tr>
<td>2</td>
<td>BTN_D</td>
<td>0x04</td>
<td>I</td>
<td>BTN_I</td>
<td>0x19</td>
</tr>
<tr>
<td>3</td>
<td>BTN_E</td>
<td>0x05</td>
<td>J</td>
<td>BTN_J</td>
<td>0x1A</td>
</tr>
<tr>
<td>4</td>
<td>BTN_F</td>
<td>0x06</td>
<td>K</td>
<td>BTN_K</td>
<td>0x1B</td>
</tr>
<tr>
<td>5</td>
<td>BTN_G</td>
<td>0x07</td>
<td>L</td>
<td>BTN_L</td>
<td>0x1C</td>
</tr>
<tr>
<td>6</td>
<td>BTN_H</td>
<td>0x08</td>
<td>M</td>
<td>BTN_M</td>
<td>0x1D</td>
</tr>
<tr>
<td>7</td>
<td>BTN_I</td>
<td>0x09</td>
<td>N</td>
<td>BTN_N</td>
<td>0x1E</td>
</tr>
<tr>
<td>8</td>
<td>BTN_J</td>
<td>0x0A</td>
<td>O</td>
<td>BTN_O</td>
<td>0x1F</td>
</tr>
<tr>
<td>9</td>
<td>BTN_K</td>
<td>0x0B</td>
<td>P</td>
<td>BTN_P</td>
<td>0x20</td>
</tr>
<tr>
<td>0</td>
<td>BTN_L</td>
<td>0x0C</td>
<td>Q</td>
<td>BTN_Q</td>
<td>0x21</td>
</tr>
<tr>
<td>A</td>
<td>BTN_A</td>
<td>0x22</td>
<td>F</td>
<td>BTN_F</td>
<td>0x23</td>
</tr>
<tr>
<td>B</td>
<td>BTN_B</td>
<td>0x24</td>
<td>G</td>
<td>BTN_G</td>
<td>0x25</td>
</tr>
<tr>
<td>C</td>
<td>BTN_C</td>
<td>0x26</td>
<td>H</td>
<td>BTN_H</td>
<td>0x27</td>
</tr>
<tr>
<td>D</td>
<td>BTN_D</td>
<td>0x28</td>
<td>I</td>
<td>BTN_I</td>
<td>0x29</td>
</tr>
<tr>
<td>E</td>
<td>BTN_E</td>
<td>0x2A</td>
<td>J</td>
<td>BTN_J</td>
<td>0x2B</td>
</tr>
<tr>
<td>F</td>
<td>BTN_F</td>
<td>0x2C</td>
<td>K</td>
<td>BTN_K</td>
<td>0x2D</td>
</tr>
<tr>
<td>G</td>
<td>BTN_G</td>
<td>0x2E</td>
<td>L</td>
<td>BTN_L</td>
<td>0x2F</td>
</tr>
<tr>
<td>H</td>
<td>BTN_H</td>
<td>0x30</td>
<td>M</td>
<td>BTN_M</td>
<td>0x31</td>
</tr>
<tr>
<td>I</td>
<td>BTN_I</td>
<td>0x32</td>
<td>N</td>
<td>BTN_N</td>
<td>0x33</td>
</tr>
<tr>
<td>J</td>
<td>BTN_J</td>
<td>0x34</td>
<td>O</td>
<td>BTN_O</td>
<td>0x35</td>
</tr>
<tr>
<td>K</td>
<td>BTN_K</td>
<td>0x36</td>
<td>P</td>
<td>BTN_P</td>
<td>0x37</td>
</tr>
<tr>
<td>L</td>
<td>BTN_L</td>
<td>0x38</td>
<td>Q</td>
<td>BTN_Q</td>
<td>0x39</td>
</tr>
</tbody>
</table>

![Remote Control Image](image-url)
4.9. Appendix 9 – RBC Box Extension Port, IO MAP

1) RBC Box Extension Port

2) ATMEGA128 pin placement
### 3) RBC Box (RBC-08128NNN, RBC-08128YNN)  IO MAP

<table>
<thead>
<tr>
<th>PIN No.</th>
<th>PIN Name</th>
<th>In/Output dir</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PEN</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>2</td>
<td>(RXD0/PD0) PE0</td>
<td>I</td>
<td>Communication with wCK, Sound IC (YNN model only)</td>
</tr>
<tr>
<td>3</td>
<td>(TXD0/PD0) PE1</td>
<td>O</td>
<td>Communication with wCK, Sound IC (YNN model only)</td>
</tr>
<tr>
<td>4</td>
<td>(XCK0/AIN0) PE2</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>5</td>
<td>(OC3A/AIN1) PE3</td>
<td>O</td>
<td>Speaker Sound Output (YNN model only)</td>
</tr>
<tr>
<td>6</td>
<td>(OC3B/INT4) PE4</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>7</td>
<td>(OC3C/INT5) PE5</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>8</td>
<td>(T3/INT6) PE6</td>
<td>I</td>
<td>IR remote control signal receiver module (38kHz)</td>
</tr>
<tr>
<td>9</td>
<td>(ICP3/INT7) PE7</td>
<td>I</td>
<td>Bluetooth module signal receiver module</td>
</tr>
<tr>
<td>10</td>
<td>(SS)</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>11</td>
<td>(SCK) PB1</td>
<td>O</td>
<td>ISP</td>
</tr>
<tr>
<td>12</td>
<td>(MOSI) PB2</td>
<td>O</td>
<td>24LC256T-I/SN power supply (High : ON, Low : OFF)</td>
</tr>
<tr>
<td>13</td>
<td>(MISO) PB3</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>14</td>
<td>(OC0) PB4</td>
<td>O</td>
<td>Battery charge (High : charge ON, Low : charge OFF)</td>
</tr>
<tr>
<td>15</td>
<td>(OC1A) PB5</td>
<td>O</td>
<td>PSD distance sensor GP2Y0A21YK0F, Power control (High : ON, Low : OFF)</td>
</tr>
<tr>
<td>16</td>
<td>(OC1B) PB6</td>
<td>O</td>
<td>Sound IC reset (High : Disabled, Low : Enabled)</td>
</tr>
<tr>
<td>17</td>
<td>(OC2/OC1C) PB7</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>18</td>
<td>TOSC2 / PG3</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>19</td>
<td>TOSC1 / PG4</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>20</td>
<td>(SCL/INT0) PD0</td>
<td>O</td>
<td>Serial EEPROM communication (SCL)</td>
</tr>
<tr>
<td>21</td>
<td>(SDA/INT1) PD1</td>
<td>I/O</td>
<td>Serial EEPROM communication (SDA)</td>
</tr>
<tr>
<td>22</td>
<td>(RXD1/INT2) PD2</td>
<td>I</td>
<td>PC communication</td>
</tr>
<tr>
<td>23</td>
<td>(TXD1/INT3) PD3</td>
<td>O</td>
<td>PC communication</td>
</tr>
<tr>
<td>24</td>
<td>(ICP1) PD4</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>25</td>
<td>(XCK1) PD5</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>26</td>
<td>(T1) PD6</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>27</td>
<td>(T2) PD7</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>28</td>
<td>(WR) PG0</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>29</td>
<td>(RD) PG1</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>30</td>
<td>(A8) PC0</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>31</td>
<td>(A9) PC1</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>32</td>
<td>(A10) PC2</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>33</td>
<td>(A11) PC3</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>34</td>
<td>(A12) PC4</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>35</td>
<td>(A13) PC5</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>36</td>
<td>(A14) PC6</td>
<td>X</td>
<td>Not Connected</td>
</tr>
<tr>
<td>37</td>
<td>(A15) PC7</td>
<td>O</td>
<td>Power LED (Red)</td>
</tr>
<tr>
<td>38</td>
<td>(ALE) PG2</td>
<td>O</td>
<td>Power LED (Green)</td>
</tr>
<tr>
<td>39</td>
<td>(AD7) PA7</td>
<td>O</td>
<td>Error LED (Red)</td>
</tr>
<tr>
<td>40</td>
<td>(AD6) PA6</td>
<td>O</td>
<td>Run LED (Green)</td>
</tr>
<tr>
<td>41</td>
<td>(AD5) PA5</td>
<td>O</td>
<td>Run LED (Blue)</td>
</tr>
<tr>
<td>42</td>
<td>(AD4) PA4</td>
<td>O</td>
<td>PF2 LED (Orange)</td>
</tr>
<tr>
<td>43</td>
<td>(AD3) PA3</td>
<td>O</td>
<td>PF1 LED (Red)</td>
</tr>
<tr>
<td>44</td>
<td>(AD2) PA2</td>
<td>O</td>
<td>PF1 LED (Blue)</td>
</tr>
<tr>
<td>45</td>
<td>(AD1) PA1</td>
<td>I</td>
<td>PF2 Switch (High : Not pressed, Low : Pressed)</td>
</tr>
<tr>
<td>46</td>
<td>(AD0) PA0</td>
<td>I</td>
<td>PF1 Switch (High : Not pressed, Low : Pressed)</td>
</tr>
<tr>
<td>47</td>
<td>(ADC7/TDI) PF7</td>
<td>I</td>
<td>ISP</td>
</tr>
<tr>
<td>48</td>
<td>(ADC6/TDO) PF6</td>
<td>O</td>
<td>ISP</td>
</tr>
<tr>
<td>49</td>
<td>(ADC5/TMS) PF5</td>
<td>O</td>
<td>ISP</td>
</tr>
<tr>
<td>50</td>
<td>(ADC4/TCK) PF4</td>
<td>O</td>
<td>ISP</td>
</tr>
<tr>
<td>51</td>
<td>(ADC3) PF3</td>
<td>I</td>
<td>MIC OUT (0~5V)</td>
</tr>
<tr>
<td>52</td>
<td>(ADC2) PF2</td>
<td>I</td>
<td>GND</td>
</tr>
<tr>
<td>53</td>
<td>(ADC1) PF1</td>
<td>I</td>
<td>Input Voltage (=wCK output voltage x 560 / 1560)</td>
</tr>
<tr>
<td>54</td>
<td>(ADC0) PF0</td>
<td>I</td>
<td>PSD distance sensor GP2Y0A21YK0F signal (0~3.2V)</td>
</tr>
</tbody>
</table>
## 4.10. Appendix 10 – Motion & Sound List

<table>
<thead>
<tr>
<th>Platform</th>
<th>Creator HUNO</th>
<th>Creator DINO</th>
<th>Creator DOGY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion No.</td>
<td>Motion Name</td>
<td>Motion Name</td>
<td>Motion Name</td>
</tr>
<tr>
<td>1</td>
<td>Getup A</td>
<td>Getup A</td>
<td>Getup A</td>
</tr>
<tr>
<td>2</td>
<td>Getup B</td>
<td>Getup B</td>
<td>Getup B</td>
</tr>
<tr>
<td>3</td>
<td>Turn Left</td>
<td>Turn Left</td>
<td>Turn Left</td>
</tr>
<tr>
<td>4</td>
<td>Move Forward</td>
<td>Move Forward</td>
<td>Move Forward</td>
</tr>
<tr>
<td>5</td>
<td>Turn Right</td>
<td>Turn Right</td>
<td>Turn Right</td>
</tr>
<tr>
<td>6</td>
<td>Move Left</td>
<td>Move Left</td>
<td>Move Left</td>
</tr>
<tr>
<td>7</td>
<td>Basic Posture</td>
<td>Basic Posture</td>
<td>Basic Posture</td>
</tr>
<tr>
<td>8</td>
<td>Move Right</td>
<td>Move Right</td>
<td>Move Right</td>
</tr>
<tr>
<td>9</td>
<td>Attack Left</td>
<td>Attack Left</td>
<td>Attack Left</td>
</tr>
<tr>
<td>10</td>
<td>Move Backward</td>
<td>Move Backward</td>
<td>Move Backward</td>
</tr>
<tr>
<td>11</td>
<td>Attack Right</td>
<td>Attack Right</td>
<td>Attack Right</td>
</tr>
<tr>
<td>12*</td>
<td>Download Motion 1 (Button 1)</td>
<td>Download Motion 1 (Button 1)</td>
<td>Download Motion 1 (Button 1)</td>
</tr>
<tr>
<td>13*</td>
<td>Download Motion 2 (Button 2)</td>
<td>Download Motion 2 (Button 2)</td>
<td>Download Motion 2 (Button 2)</td>
</tr>
<tr>
<td>14*</td>
<td>Download Motion 3 (Button 3)</td>
<td>Download Motion 3 (Button 3)</td>
<td>Download Motion 3 (Button 3)</td>
</tr>
<tr>
<td>15*</td>
<td>Download Motion 4 (Button 4)</td>
<td>Download Motion 4 (Button 4)</td>
<td>Download Motion 4 (Button 4)</td>
</tr>
<tr>
<td>16*</td>
<td>Download Motion 5 (Button 5)</td>
<td>Download Motion 5 (Button 5)</td>
<td>Download Motion 5 (Button 5)</td>
</tr>
<tr>
<td>17*</td>
<td>Download Motion 6 (Button 6)</td>
<td>Download Motion 6 (Button 6)</td>
<td>Download Motion 6 (Button 6)</td>
</tr>
<tr>
<td>18*</td>
<td>Download Motion 7 (Button 7)</td>
<td>Download Motion 7 (Button 7)</td>
<td>Download Motion 7 (Button 7)</td>
</tr>
<tr>
<td>19*</td>
<td>Download Motion 8 (Button 8)</td>
<td>Download Motion 8 (Button 8)</td>
<td>Download Motion 8 (Button 8)</td>
</tr>
<tr>
<td>20*</td>
<td>Download Motion 9 (Button 9)</td>
<td>Download Motion 9 (Button 9)</td>
<td>Download Motion 9 (Button 9)</td>
</tr>
<tr>
<td>21*</td>
<td>Download Motion 10 (Button 0)</td>
<td>Download Motion 10 (Button 0)</td>
<td>Download Motion 10 (Button 0)</td>
</tr>
<tr>
<td>33*</td>
<td>Download Motion 11 (Button ++1)</td>
<td>Download Motion 11 (Button ++1)</td>
<td>Download Motion 11 (Button ++1)</td>
</tr>
<tr>
<td>34*</td>
<td>Download Motion 12 (Button ++2)</td>
<td>Download Motion 12 (Button ++2)</td>
<td>Download Motion 12 (Button ++2)</td>
</tr>
<tr>
<td>35*</td>
<td>Download Motion 13 (Button ++3)</td>
<td>Download Motion 13 (Button ++3)</td>
<td>Download Motion 13 (Button ++3)</td>
</tr>
<tr>
<td>36*</td>
<td>Download Motion 14 (Button ++4)</td>
<td>Download Motion 14 (Button ++4)</td>
<td>Download Motion 14 (Button ++4)</td>
</tr>
<tr>
<td>37*</td>
<td>Download Motion 15 (Button ++5)</td>
<td>Download Motion 15 (Button ++5)</td>
<td>Download Motion 15 (Button ++5)</td>
</tr>
<tr>
<td>38*</td>
<td>Download Motion 16 (Button ++6)</td>
<td>Download Motion 16 (Button ++6)</td>
<td>Download Motion 16 (Button ++6)</td>
</tr>
<tr>
<td>39*</td>
<td>Download Motion 17 (Button ++7)</td>
<td>Download Motion 17 (Button ++7)</td>
<td>Download Motion 17 (Button ++7)</td>
</tr>
<tr>
<td>40*</td>
<td>Download Motion 18 (Button ++8)</td>
<td>Download Motion 18 (Button ++8)</td>
<td>Download Motion 18 (Button ++8)</td>
</tr>
<tr>
<td>41*</td>
<td>Download Motion 19 (Button ++9)</td>
<td>Download Motion 19 (Button ++9)</td>
<td>Download Motion 19 (Button ++9)</td>
</tr>
<tr>
<td>42*</td>
<td>Download Motion 20 (Button ++0)</td>
<td>Download Motion 20 (Button ++0)</td>
<td>Download Motion 20 (Button ++0)</td>
</tr>
</tbody>
</table>

* NOTE: Marked "*" is user downloaded motion contents saved No..
<table>
<thead>
<tr>
<th>Sound List</th>
<th>Sound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;Hello, my name is RoobBuilder&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;Go Go Go Go~&quot;</td>
</tr>
<tr>
<td>3</td>
<td>(HUNO Move Backward)</td>
</tr>
<tr>
<td>4</td>
<td>(HUNO Turn)</td>
</tr>
<tr>
<td>5</td>
<td>(HUNO Move Side)</td>
</tr>
<tr>
<td>6</td>
<td>(HUNO Attach)</td>
</tr>
<tr>
<td>7</td>
<td>(HUNO Getup A)</td>
</tr>
<tr>
<td>8</td>
<td>(HUNO Getup B)</td>
</tr>
<tr>
<td>9</td>
<td>(DOGY Basic Posture)</td>
</tr>
<tr>
<td>10</td>
<td>(DOGY Attack)</td>
</tr>
<tr>
<td>11</td>
<td>(DOGY Turn)</td>
</tr>
<tr>
<td>12</td>
<td>(DOGY Getup)</td>
</tr>
<tr>
<td>13</td>
<td>(DINO Getup)</td>
</tr>
<tr>
<td>14</td>
<td>(DINO Move Forward)</td>
</tr>
<tr>
<td>15</td>
<td>&quot;Zero&quot;</td>
</tr>
<tr>
<td>16</td>
<td>&quot;One&quot;</td>
</tr>
<tr>
<td>17</td>
<td>&quot;Two&quot;</td>
</tr>
<tr>
<td>18</td>
<td>&quot;Three&quot;</td>
</tr>
<tr>
<td>19</td>
<td>&quot;Four&quot;</td>
</tr>
<tr>
<td>20</td>
<td>&quot;Five&quot;</td>
</tr>
<tr>
<td>21</td>
<td>&quot;Six&quot;</td>
</tr>
<tr>
<td>22</td>
<td>&quot;Seven&quot;</td>
</tr>
<tr>
<td>23</td>
<td>&quot;Eight&quot;</td>
</tr>
<tr>
<td>24</td>
<td>&quot;Nine&quot;</td>
</tr>
<tr>
<td>25</td>
<td>&quot;Ten&quot;</td>
</tr>
</tbody>
</table>
This assembly instruction is to assemble the humanoid more easily for Robot beginner. The following assembly instruction is one of the RoboBuilder humanoid model “HUNO”.

STEP 1. Part List

1-1 : Place all the parts list for HUNO as shown in the below.
STEP 2. wCK module (servo motor) connection and basic posture setting.

To assemble more easily, set each wCK module for the HUNO basic posture. This helps to avoid the coupling confusion between wCK module and joints.

2-1 : Arrange each wCK ID module, Joint, Head, Body Frame, Hand, Leg, Foot, Shoe parts as shown in the below.
2-2 : Connector connection – Check out the wCK cable connector hole with wCK module connector pin.

2-3 : Connect power adapter with RBC Box first, then, connect all the wCK module as shown in the below.
2-4 : Press IR Remote Controller STOP (RED) button for HUNO basic posture after power on RBC Box as shown in the below.

If RBC Box shows ERROR LED (RED), check out the above wCK module connection again whether they were connected properly. If the error problem is still remained, contact your local RoboBuilder seller for technical support.

2-5 : After above step 2-4, wCK module keeps HUNO basic posture position unless user rotate it. Disconnect all the wCK cables from wCK modules, and arrange it as shown in the below.

Basic setting for HUNO assembly is now finished.
STEP 3. Assembly Left Leg

3-1: Shoe assembly – Use B5 bolt for assemble the Foot with Shoe as shown in the below. The longer part is backside. Assemble two Feet and shoes because the left side and right side are the same for these parts.

3-2: Connect wCK ID04 module with Foot part by using four of B40 bolts and nuts as shown in the below.
3-3 : Put J2 (Joint 2) into wCK ID 04 module and fix it by using B8 bolt. User should follow the Joint assembly as shown in the below.

Caution : Hold tightly J2 as shown in the above when you screw the B8 bolt. Be sure NOT to break the bolt head part as user screws it so forcefully. Once the bolt head is worn out, the 4th gear of wCK module should be replaced together.

3-4 : Connect two wCK cables with wCK ID03 module as shown in the below.
3-5: Put wCK ID03 module in side of Knee part, and pull out the wCK cables toward outside as shown in the below.

3-6: Use B40 bolts and nuts to connect wCK ID03 module as shown in the below.

3-7: Put wCK cable into wCK ID02 module as shown in the below.
3-8: Assemble the wCK ID02 module as shown in the below.

3-9: Use nuts and B40 bolts to connect wCK ID02 module with Knee part as shown in the below.

3-10: Arrange the wCK cable as shown in the below.

Tip Scotch Tape could be helpful for fixing the wCK cable.
3-11 : Put J6 (Joint 6) into wCK ID2 module. Use B12 bolt to connect between wCK ID02 module and J6 as shown in the below.

3-12 : Connect wCK ID01 module with J6 for coupling with wCK ID02 module. Use two B16 bolts and nuts to fix it as shown in the below.

3-13 : Put the wCK cable from wCK ID2 module into wCK ID01 module connector. Arrange the wCK cables as shown in the below.
3-14: Connect with wCK ID04 module as shown in the below.

3-15: For coupling wCK ID03 module with wCK ID04 module, use B12 bolt as shown in the below.
3-16 : Arrange the wCK cable as shown in the below, which is used for wCK ID03 module and wCK ID04 module.

3-17 : Left Leg Part is finished for HUNO.
STEP 4. Assembly Right Leg
Right Leg is assembled same way as the Left Leg.

4-1 : Use wCK ID09 module and four B40 bolts and nuts as shown in the below.

4-2 : Put J2 into wCK ID09 module, and screw B8 bolt as shown in the below.
4-3 : Connect 2 wCK cables into wCK ID08 module connector. Put wCK ID08 module into the Knee part and screw B40 bolt as shown in the below.

4-4 : Connect wCK cable with wCK ID07 module as shown in the below. Put wCK ID07 module into the Knee part, and screw B40 bolt as shown in the below. Pull the wCK cables out as shown in the below.
4-5 : Put J6 into wCK ID07 module and screw B12 bolt as shown in the below.

4-6 : Coupling wCK ID07 module with wCK ID6 module by using J6. and screw two B16 bolt and nuts as shown in the below.

4-7 : Connect wCK cables into wCK ID06 module connector and arrange the wCK cable as shown in the below.
4-8 : Twist the wCK cable one time as shown in the below. You could use scotch tape to fix it later.

4-9 : Use B12 Bolt to connect between J2 and wCK ID08 module.
4-10 : Wind up wCK cable one time as shown in the below and put wCK cable into wCK ID09 module connector.

4-11 : Both Leg parts are finished as shown in the below.
STEP 5. Assembly Left Arm and Right Arm
Assembling arm part is easier than Leg part.

5-1 : Left Arm – Place the parts as shown in the below. Insert J4 (Joint 4) into wCK ID12 module, and use B16 bolt to fix J4 with wCK ID12 module.

5-2 : Use B16 Bolts and nuts to fix joint and module as shown in the below.
5-3 : Put J6 into wCK ID12 module and screw the B12 bolt to fix it.

5-4 : Put wCK cable into wCK ID11 module connector. Insert J6 into the wCK ID11 module as shown in the below.

5-5 : Screw two B16 bolts and nuts to fix the J6 with wCK ID11 module. Twist wCK cable one time and put into wCK ID12 connector as shown in the below.
5-6 : Right Arm – Assembly Hand, J4, wCK ID15 module. For J4 fixing, use B16 bolt as shown in the below.

5-7 : Insert J6 into wCK ID15 module and screw the B12 bolt as shown in the below.
5-8 : Put wCK cable into wCK ID14 module connector. Use B16 bolts and nuts to fix J6 with wCK ID14 module as shown in the below.

5-9 : Both Leg Parts and Arm Parts are finished.
STEP 6. Assembly Body Part
Place Body Parts as shown in the below.

6-1 : Let’s start with wCK ID05 module. Put two wCK cables into both wCK ID05 module connector. Insert J2 (Joint 2) into the wCK ID05 module as shown in the below. Be careful with J2 insert direction. Use B8 bolt to fix the J2 (Joint 2).
6-2 : Screw four B40 bolts and nuts for coupling between Body frame and wCK ID05 module as shown in the below.

6-3 : Put two wCK cables into the both wCK ID00 module connectors. Insert J2 into wCK ID00 module, and use B8 bolt to fix it as shown in the below.
6-4 : Use four B40 bolts and nuts for coupling between Body frame and wCK ID00 module as shown in the below.

6-5 : Arrange the wCK cables as shown in the below.
6-6: Right Shoulder – Insert J2 (Joint 2) into wCK ID13 module as shown in the below. Be careful with direction and position (Joint insert angle is very important). Put two wCK cable into wCK ID13 module connectors as shown in the below.

6-7: Hold J2 tightly and screw B8 bolt to fix J2.

6-8: Coupling wCK ID13 module with Body frame. Screw B40 bolts and nuts as shown in the below. Below RED circle holes are the B40 bolts screwing position to fix with wCK ID13 module.
6-9: RED circles are the B40 bolts and nuts screw positions as shown in the below.

6-10: Left Shoulder – Put two wCK cables into wCK ID10 module connectors. Insert J2 (Joint 2) into wCK ID10 module. Be careful with J2 direction and position. Hold J2 tightly NOT to move it when you screw B8 bolt to fix J2 with wCK ID10 module as shown in the below.
6-11 : Use four B40 bolts and nuts for coupling wCK10 module with Body frame as shown in the below.

6-12 : Below is the Body frame with the assembled wCK ID 10 module.
STEP 7. Assemble HUNO Head

7-1 : Head position is between wCK ID10 module and wCK ID13 module. Use B6 bolt to fix HUNO Head as shown in the below.

7-2 : Pull out wCK cable as shown in the below.

STEP 8. Assembly Chest

8-1 : After arrange the wCK cable, next step is assembling Chest part.
8-2 : Use B6 bolt to fix Chest part as shown in the below.

STEP 9. Mount RBC(RoboBuilder Control) Box

**Caution**: Be careful NOT to mount upside down.

9-1. Mount RBC Box as shown in the below. Put both wCK cables from wCK ID00 module and wCK ID05 module into the bottom of RBC Box connectors first. Put the rest wCK cables into the top of RBC Box connectors. Then, put the Head cable into RBC Box connector as shown in the below.
STEP 10. Mount Leg and Arm

10-1 : Connect J2(Joint 2) from wCK ID00 module with wCK ID01 module as shown in the below.

10-2 : Rotate Leg part a little bit and screw B12 bolt into the J2 (Joint 2) as shown in the below.

10-3 : Right Leg part assembling way is the same as the Left Leg part. Connect the J2 (Joint 2) from wCK ID05 module with wCK ID06 module. Screw B12 bolt into J2(Joint 2) as shown in the below.
10-4 : Twist wCK cable one time as shown in the below. Put wCK cable from wCK ID00 module into wCK ID01 module connector as shown in the below.

10-5 : Twist wCK cable one time as shown in the below. Put wCK cable from wCK ID05 module into wCK ID06 module connector as shown in the below.

10-6 : Mount Right Arm – Screw B12 bolt into the J2 (Joint 2) to connect with wCK ID14 module as shown in the below.

Caution – Be careful with assembly angle as shown in the below.
10-7 : Mount Left Arm - Screw B12 bolt into the J2 (Joint 2) to connect with wCK ID11 module as shown in the below.

10-8 : Put the one two wCK cables into wCK ID12 module and put the other wCK cable into wCK ID14 module as shown in the below.
STEP 11. Assembly Finished and Check

11-1: Check whether it was assembled correctly or not.
To check, power on RBC Box and press STOP (RED) button from IR remote
Controller towards HUNO Head IR receiver
(Below RED circle is Head IR receiver position).

HUNO takes Basic Posture as shown in the below.

11-2: Check the Basic Posture

If the HUNO does NOT take the above basic posture, it was NOT assembled
correctly. If so, check the assembly procedures again step by step.