Motion Builder Tutorial

2008.07.07

RoboBuilder

<CONTENT>

1. Before beginning

This document explains how to use MotionBuilder the motion building tool for RoboBuilder.

1.1. This document is recommended for people who

1) agree that RoboBuilder is not just an expensive toy
2) want to download and use the motion files that are created by and uploaded on the Internet by others
3) want to modify others' motions and make their own motions
4) want to create a motion file for themselves
5) want to share their own motion files with many others
6) want to learn advanced features of MotionBuilder

1.2. Things to prepare

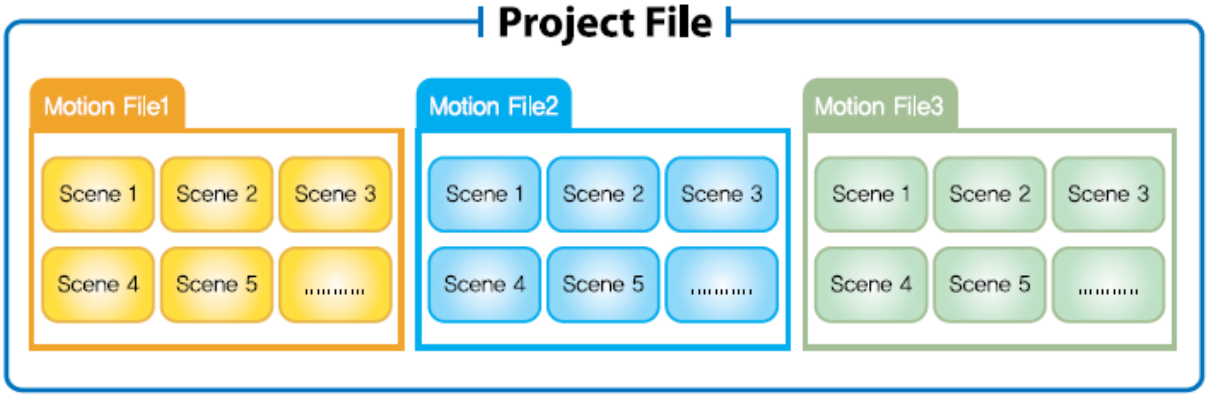
1) **Creator HUNO**
: All examples in this tutorial were done with a standard HUNO robot. All that is explained in this tutorial can be applied for other types of robots too.

2) **PC(desktop or laptop)**
: Windows XP is recommended (Windows Vista is also applicable).
In case the PC doesn't have a COM port, a USB-RS232 converter is required.

3) **PC cable, Power supply**
: the ones included in the RoboBuilder package.

4) **MotionBuilder**
: the software is available in the Homepage or the installation CD.

1.3. What are project file, 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.

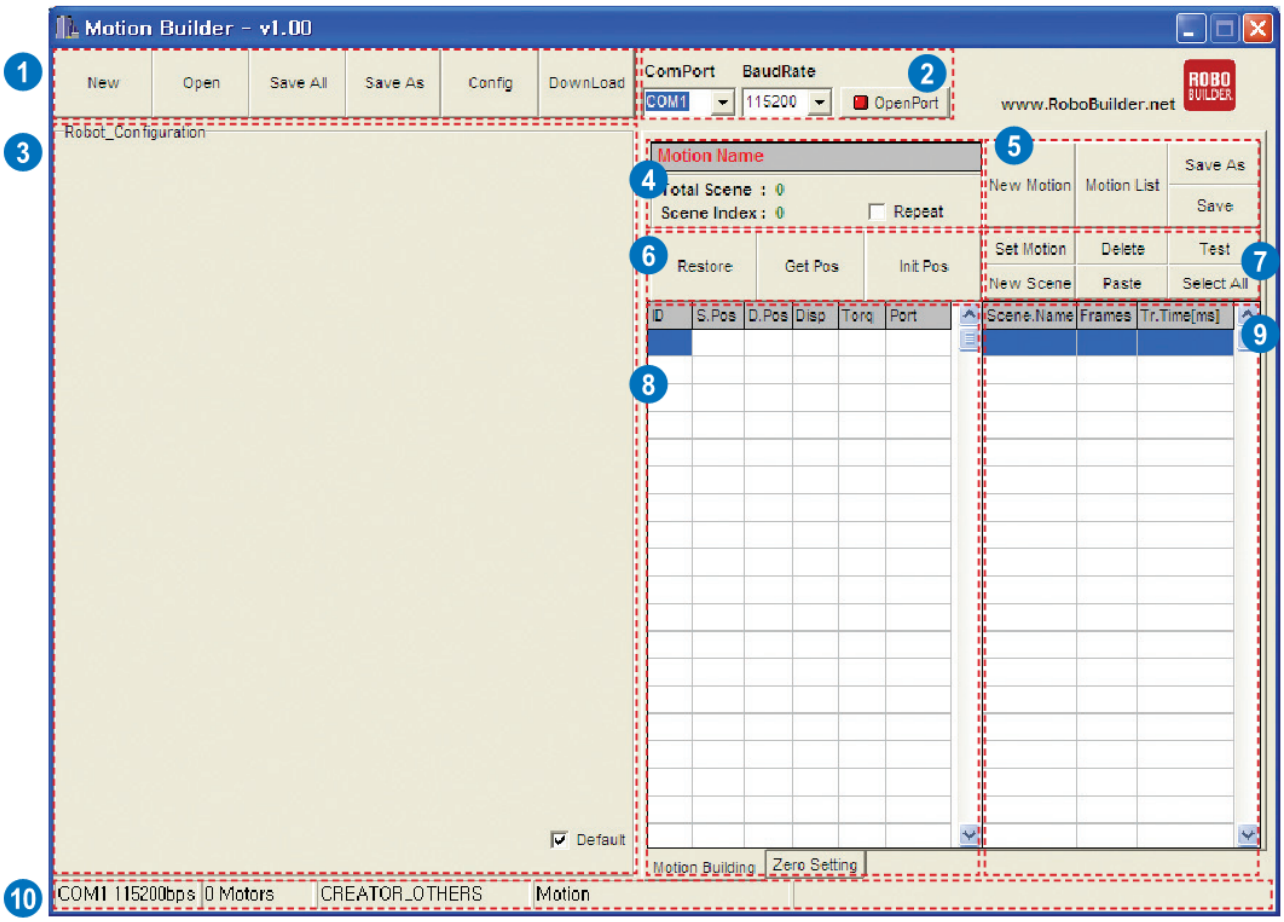
3) **Scene**
: A scene is a smaller motion unit that constitutes a complete motion file. A scene consists of start position and destination position. Except the first scene, the start position of a scene is the destination position of its previous scene. When a scene is executed, the frame data is generated automatically according to the predefiend 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 defien, 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 60000.

1.4. Screen Layout

This is the screen layout of the MotionBuilder.



No.	Area Name	Functions & Descriptions
①	Menu Bar	· New: creates a new project by defining project name, file path, robot type etc. · Open: opens an existing project file. (*.prj) · 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. · BaudRate: 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: 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. · Disp: 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. · Tr.Time[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.

1.5. Caution for input language

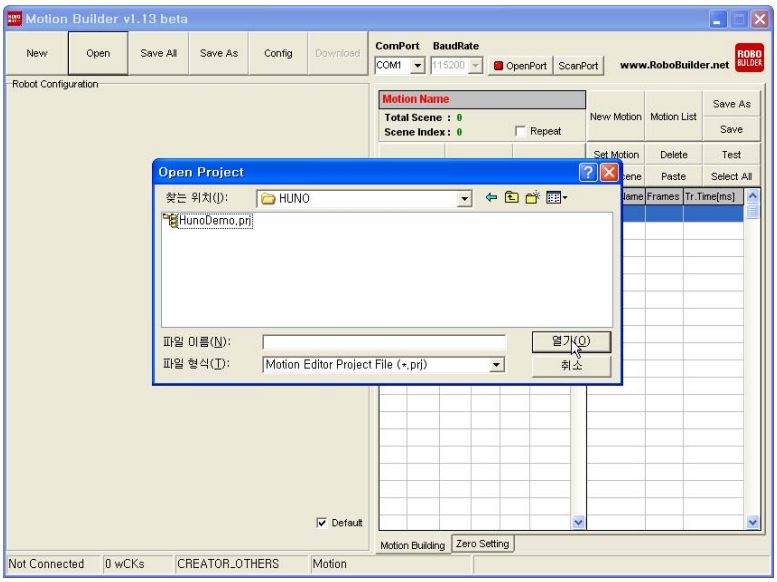
1) It is recommended that users type in English for all text input options.

2. Handling example motion files

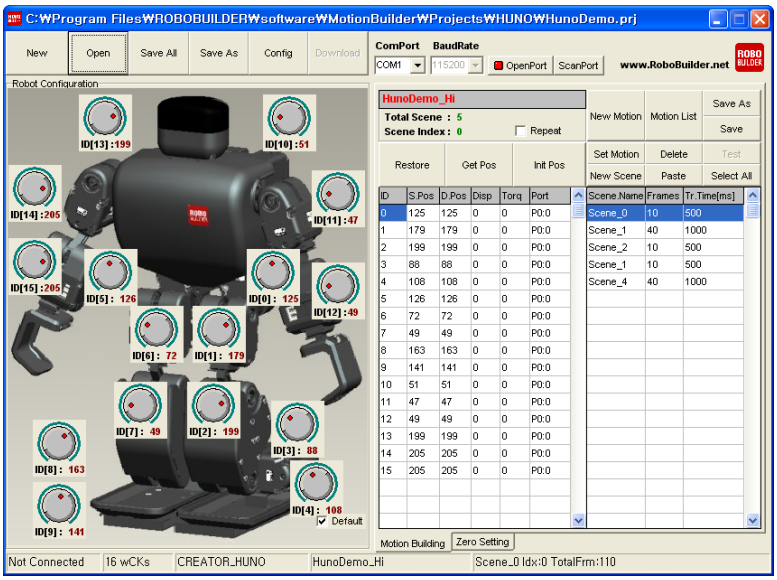
2.1. Open an example project file

1) Run MotionBuilder.

2) Click Open and selec ‘HunoDemo.Prj’ to open the project file.



3) This is the screen when a project file is opened.



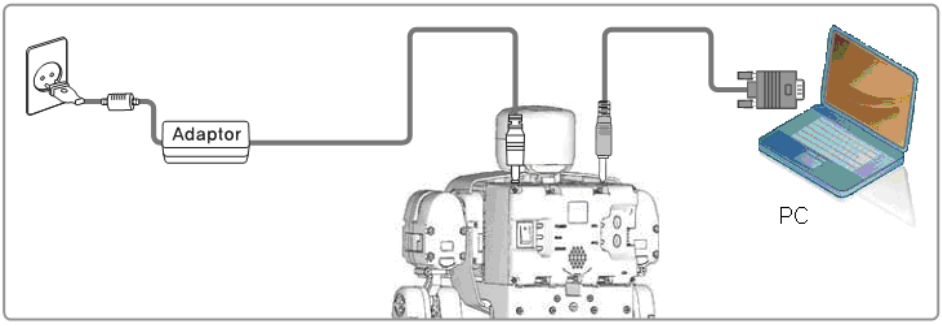
2.2. Running an example motion file

This chapter explains how to run a motion file directly from PC using MotionBuilder without transferring the file to RBC controller. In this case, all commend signals are directly sent from PC to each wCK module that comprises the robot.

1) First, open a project file as explained in 2.1.

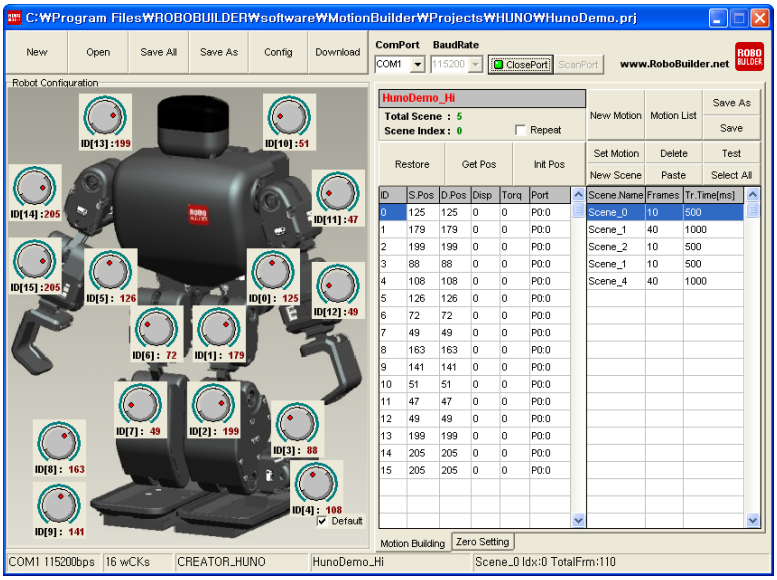
2) Connect the power supply, and use the PC cable to connect the HUNO to the PC.

※ In case the PC is not equipped with a COM port, use a USB-to-RS232 converter cable so that the PC can open a COM port. (The USB-to-RS232 converter cable is not provided.)



3) Click ScanPort and it will automatically search for COM port available and connect to RoboBuilder.

(※ caution : A PC that frequently uses virtual port such as Bluetooth Dongle may fail or take longer time to search for COM port. In this case, choose the ComPort manually and click OpenPort.)



If successfully connected, the OpenPort button turns ClosePort .

If the connection fails, the ClosePort return to OpenPort.

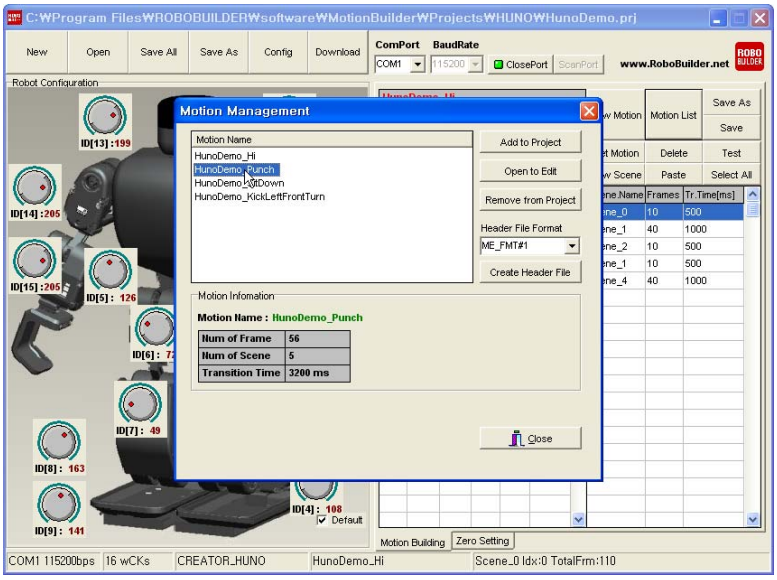
Usual causes of the failure are as follow.

a. When the PC cable is unplugged

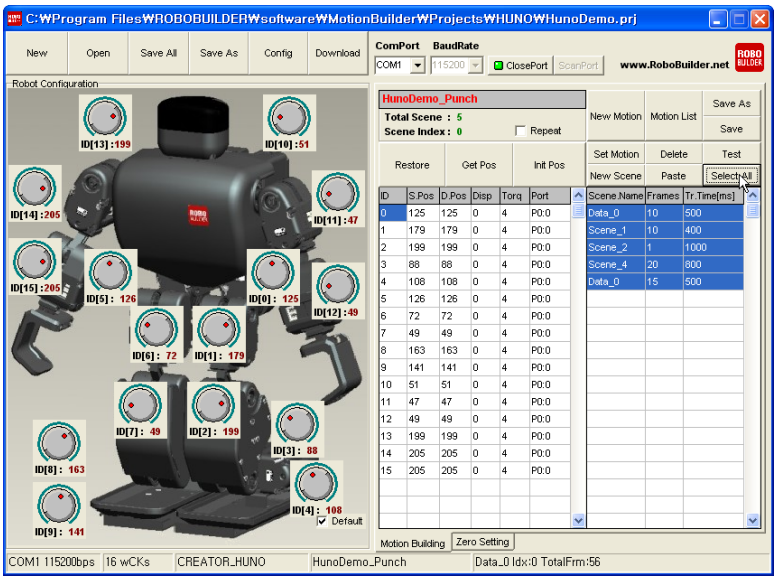
b. When the driver for USB-to-RS232 converter is not installed correctly

4) Click Motion List button and select a motion file in the Motion Management window.

For example, double click 'HunoDemo_Punch' to open the file.



5) Click Select All button to select all scenes of the motion file.



6) Click Test button. The button turns to Stop and the motion is executed.

7) As soon as the motion finished, the Stop button turns to Return.

Click Return button. The button returns to Test and the HUNO returns to its initial position.

※ If you selected only one scene, the HUNO returns to the start position of the scene.

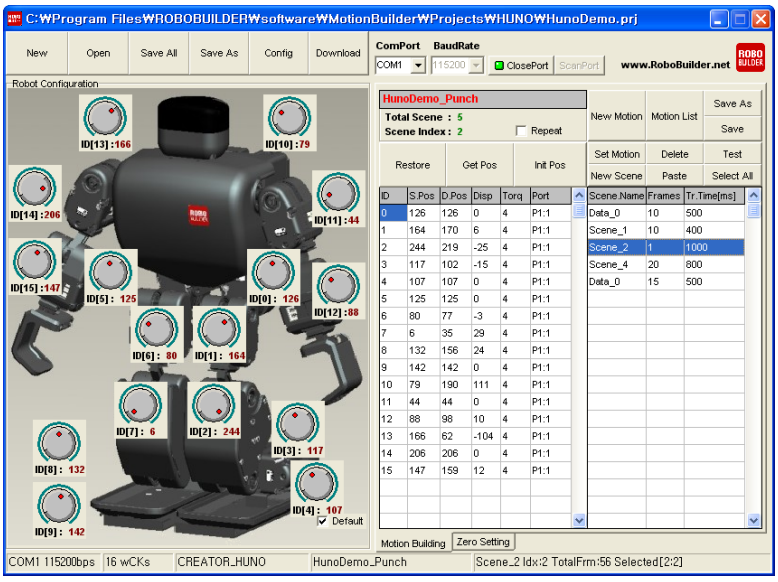
2.3. Modifying an example motion file

This chapter explains how to modify an example motion in the CD and make a new motion file.

1) Open a motion file as shown in 2.2 1)~4).

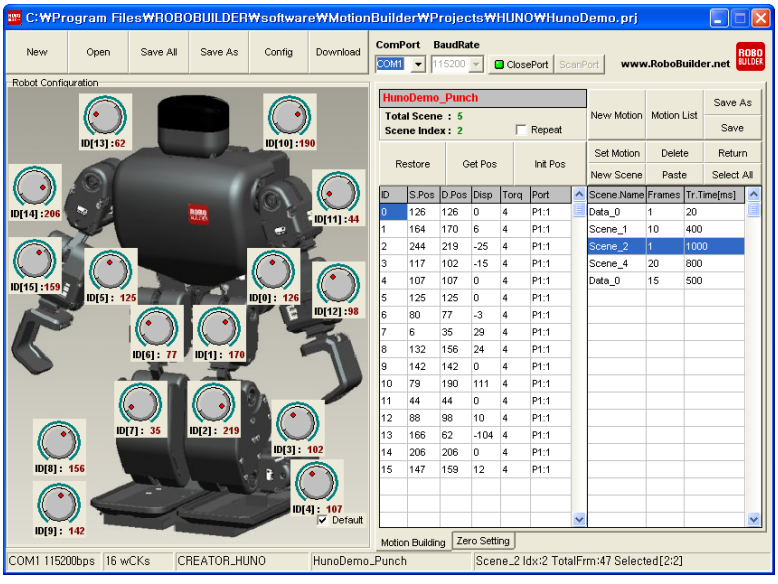
2) Select a first scene and use [↓], [↑] key or the mouse to scroll up down to select the scene to modify.

3) Select Scene_2. The robot takes the start position of Scene_2.



4) Click [Test] button.

The robot moves to the destination position of the scene.



5) Let's modify the destination position of the Scene_2. There are three different methods to do this as below.

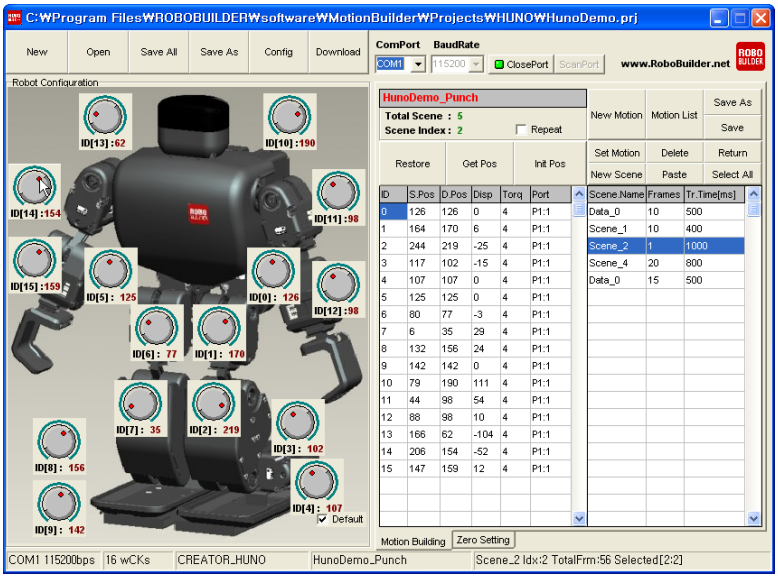
method a. Use the jog dial pad in the Robot Configuration area

method b. Change the value in the wCK Module Control Detail area

method c. Use Teaching Method (posture capture)

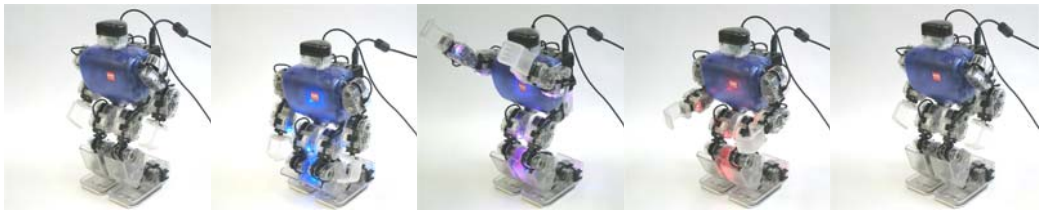
The following is how to use the method a .

6) Adjust the dial pad of **ID[11], ID[14]** in the Robot Configuration area. Let's try to make the robot open its arms wide.

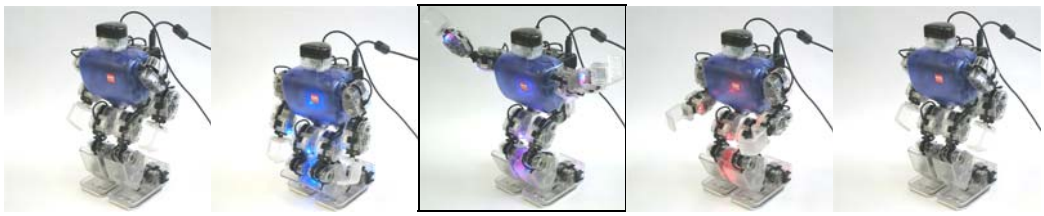


7) Click Return button to have the robot move back to start position. Click Test button to see how the modified scene works. Do you see the motion changed?

8) Click Return button to have the robot move back to start position. Click Select All button to select all scenes and click Test button. The robot show all scenes so that you can see how the small modification affect the whole motion file.



before



after

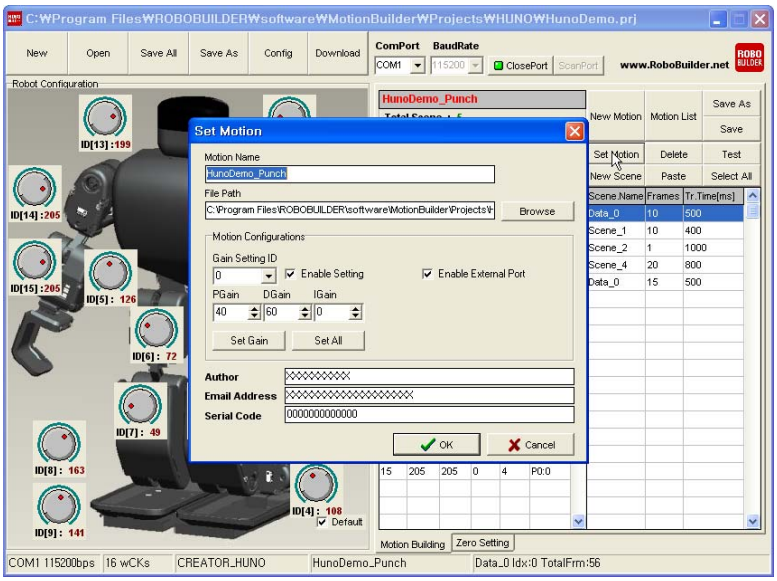
2.4. Saving an example motion file as another name

This chapter explains how to save an example motion file as another name.

For example, let's save 'HunoDemo_Punch.rbm' as 'my_punch.rbm'.

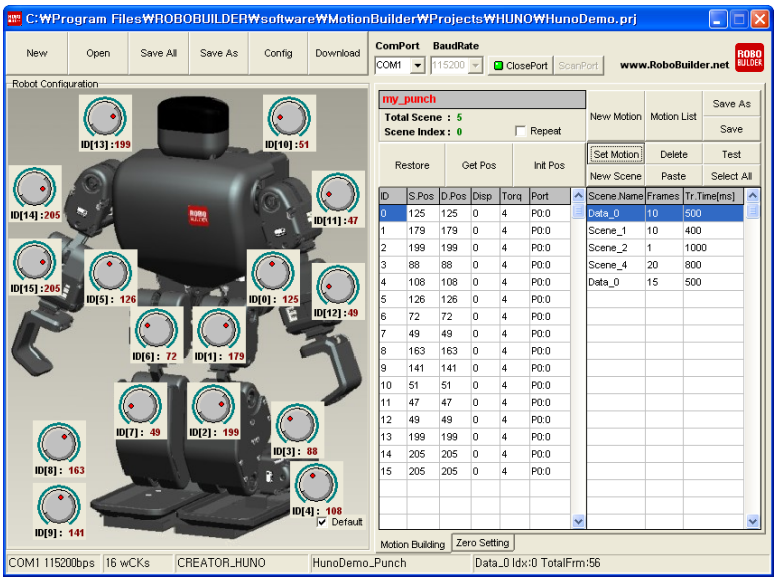
1) Open 'HunoDemo_Punch.rbm' as shown in 2.2 1)~4).

2) Click Set Motion button.



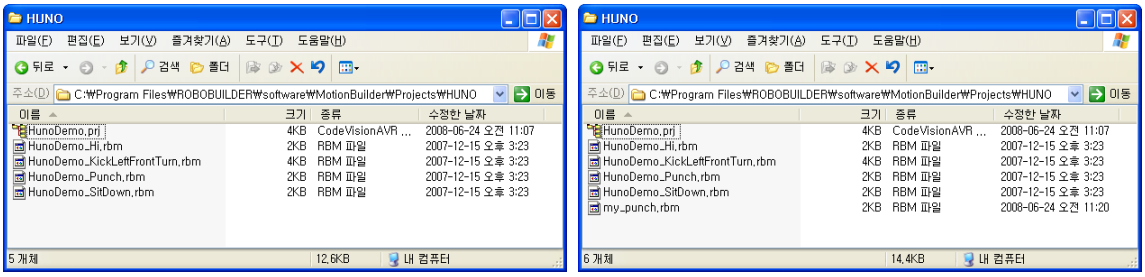
3) Change the Motion Name 'HunoDemo_Punch' as 'my_punch' and click OK button.

You can see the name has been changed to 'my_punch' in the Motion File information area.



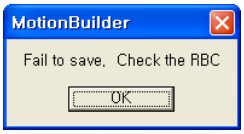
4) Click Save button to save the change.

If successfully saved, you can check the new file 'my_punch.rbm' has been created.

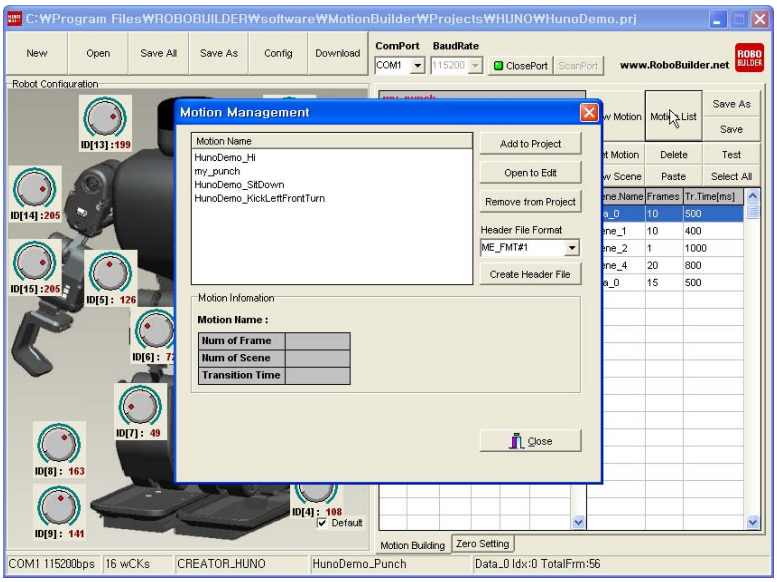


before save after save

In case there is a communication problem between the PC and the RBC control box, MotionBuilder fails to save the new file and an error message pops up as below. In this case, please check the connection status.



5) Click Motion List button. You can see now the file name is changed.



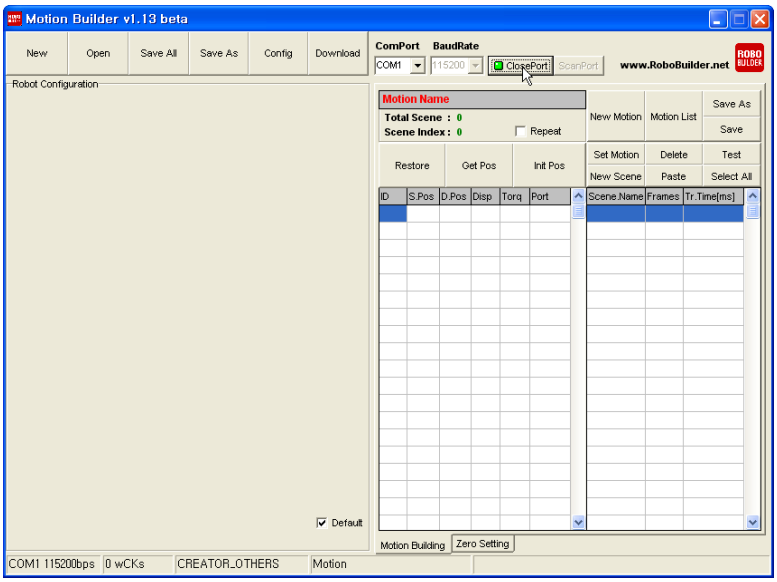
2.5. Transferring a motion file to the robot

This chapter explains how to transfer a motion file from PC to the robot.

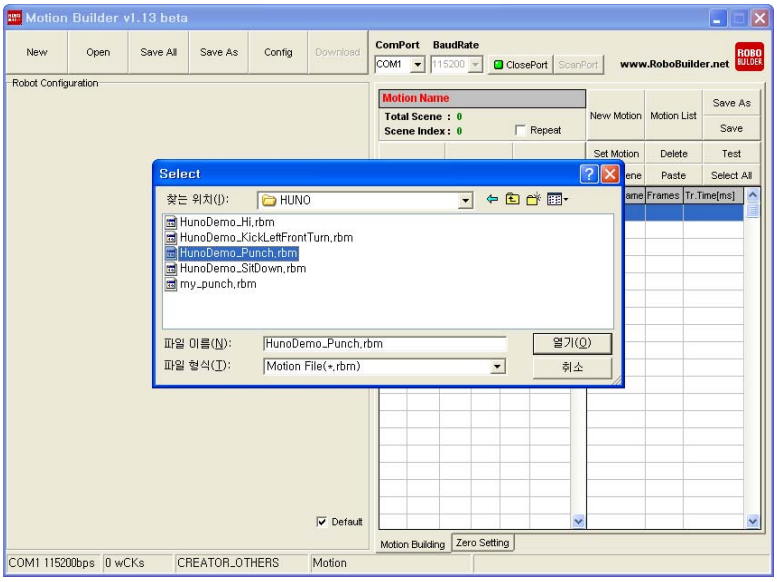
As an example, let's transfer the 'HunoDemo_Punch.rbm' and 'my_punch.rbm'.

1) Connect PC with the RBC control box and open a COM port as explained in 2.2 2)~3).

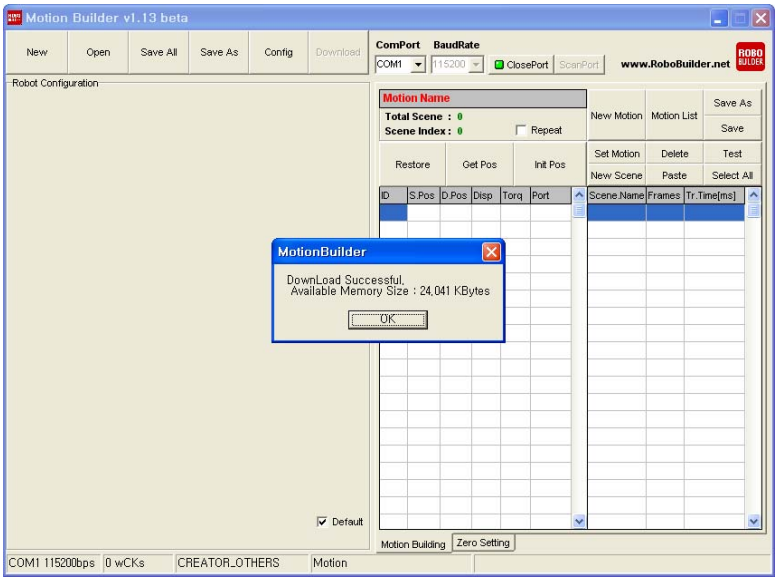
You don't need to open a project file to do this.



2) Click Download button and select the 'HunoDemo_Punch.rbm'.



3) Click Open button then the file transfer begins. When the file transfer is completed, a message appears as below.



In case the transfer fails, a 'Fail' message appears.

The usual causes of the failure are as below.

a. when the PC cable is not correctly connected

: double check the PC cable is firmly plugged in the COM port and the RBC control box.

b. when the motion file is too big or the memory space is not big enough

c. when the file was damaged or it is not a motion file(*.rbm)

4) Repeat 2)~3) to transfer the next file 'my_punch.rbm'.

5) When you finish the transfer of the two files, remove the PC cable from RBC control box. Direct the remote controller to the top of the robot's head and press the red button [basic posture]. Then press button number 1 to see if the robot show the motion of 'HunoDemo_Punch.rbm'..

※ Whenever RoboBuilder is turned on, you need to press the red button of the remote controller first before you press any other button to run any motion.

6) If you press button number 2, the robot will show the motion of 'my_punch.rbm'..

3. Handling downloaded motion files

3.1. Downloading a motion file from the Internet

You can download and use the moiton files from the Internet that other people created and uploaded there.

The following is how to download a motion file from the Internet.

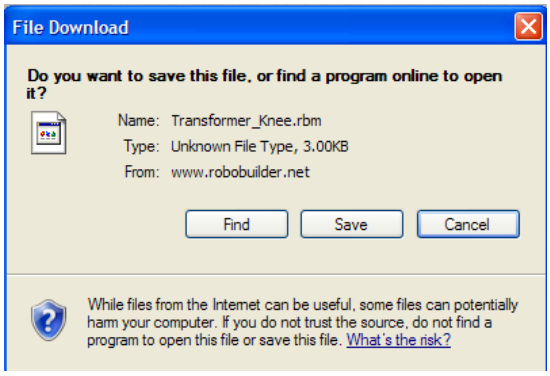
1) Join the membership at RoboBuilder's homepage (http://www.robobuilder.net). Sign in to download a file.

2) Register the serial number of your robot at the homepage (http://www.robobuilder.net/mypage/index.asp?mode=product).

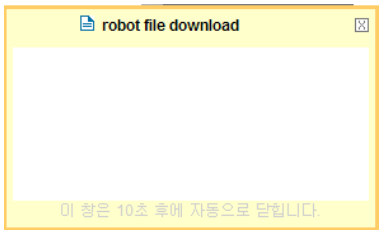
※ The serial number is printed on a bar code that is attached on to the bottom of each RBC control box.

3) Go to 'Robot File Sharing' of the homepage (http://www.robobuilder.net/community/board_index.asp?cmd=list) and select a file you want to download.

4) Click 'Download' button and choose Save button to save the file on your hard disk.



※ If you have more than one serial numbers registered, you can choose the serial number you want to use for this file download.



※ If the serial number you used here doesn't match with the actual serial number fo the RBC control box you will use to run the motion file, the file will not work .

3.2. Transferring a motion file to the robot

Please refer to the chapter 2.5. ‘Transferrring a motion file to the robot’.

One thing different from the example file is that If the serial number you used here doesn’t match with the actual serial number fo the RBC control box you will use to run the motion file, the file will not work.

In other words, you need first to register your seiral number at the homepage before you download the file from the homepage.

3.3. Modifying a motion file

Please refer to the chapter 2.3. Modifying an example motion file.

4. Uploading a motion file to the Internet

You can upload and share with others the motion files you created or modified from another file.

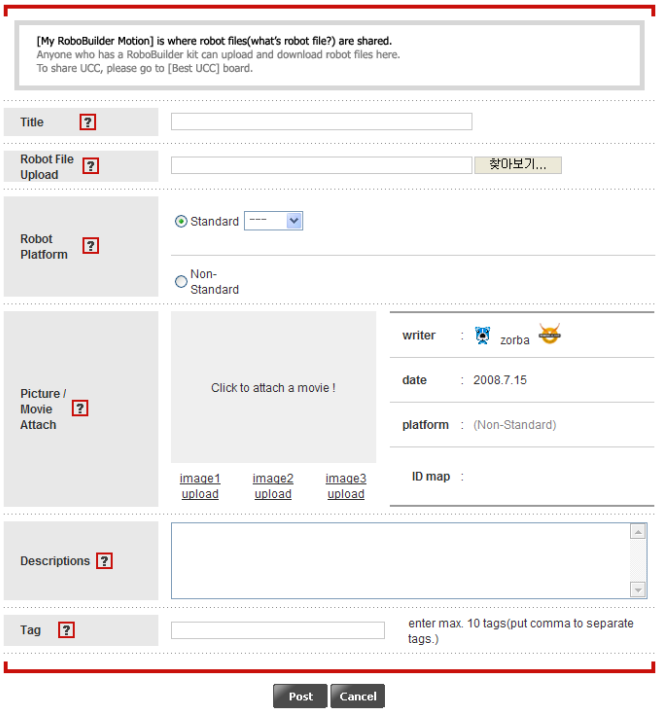
The following is how to upload a motion file to the Internet.

1) Prepare the motion file that you want to upload.

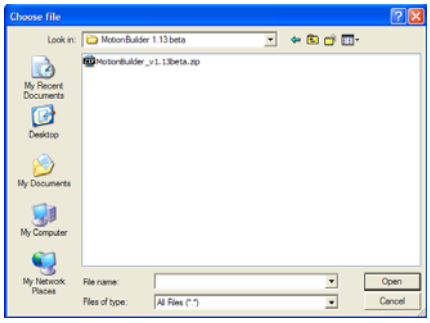
2) Use camcoder or a digital camera and take a movie picture of the moiton you will upload. This movie will be uploaded together with the motion file so that others can watch it before they decide to download the motion file.

3) Sign in the homepage.

4) Click Write button at the homepage 'Robot File Sharing' (http://www.robobuilder.net/community/board_index.asp?cmd=list).



5) Write the Title and click Browse button.



6) Choose the robot file(*.rbm) that you want to upload.
Fill in other option and be sure to upload the movie (*.avi, *.wmv, *.flv, .. etc). Click 'Post' button to finish wrting.

7) You can see now the file is posted at ‘Robot File Sharing’.

5. Creating a new motion file

This chapter explains how to create a new motion file.

Generally speaking when you 'program a motion of a robot', it means 'teaching the robot'. Therefore, MotionBuilder is a kind of teaching software.

The steps to take to create a motion is as follows.

Create a project → Create a motion file → Create the first scene → Modify the destination position of the scene → Create an additional scene → Modify the destination position of the added scene → Create the last scene → Save the motion file

More than one scene comprise a motion. Users build a motion file by modifying the destination position of each scene. There are three(3) different methods of modifying robot's posture. Users can choose one or more methods among them. As explained in chapter 2.3., they are.

9) method a. Use the jog dial pad in the Robot Configuration area

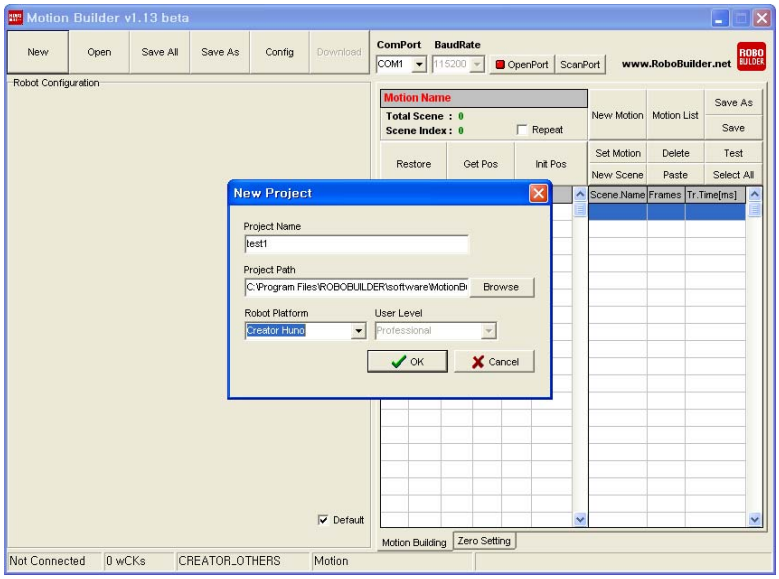
method b. Change the value in the wCK Module Control Detail area

method c. Use Teaching Method (posture capture)

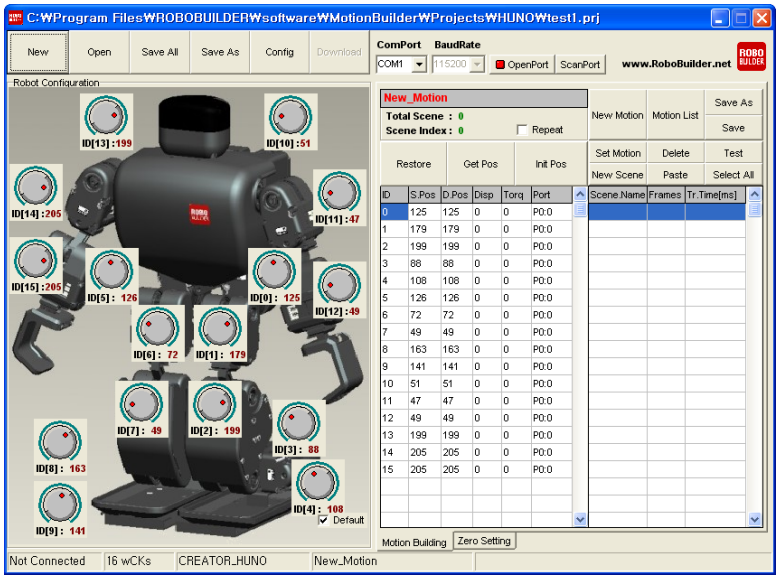
5.1. Example programming A: The robot holds up the arms. And then open the arms wide and close again. (method a, b)

Now let's build a motion that the robot holds up the arms and then open the arms wide and close again using method a and method b.

1) First, run MotionBuilder and click New button to open a new project file.

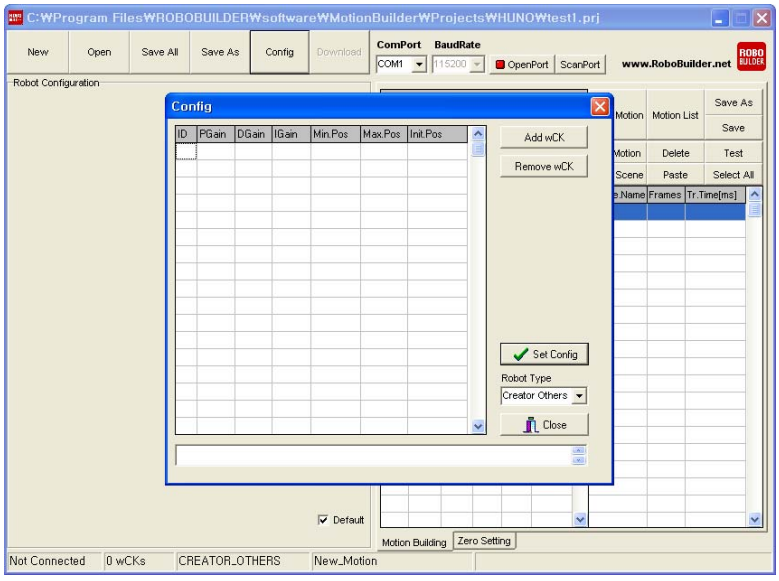


2) Set the Project Name as 'test1' and click Browse button to save the project file. Select 'Creator Huno' for Robot Platform and click OK. The screen will look as below.

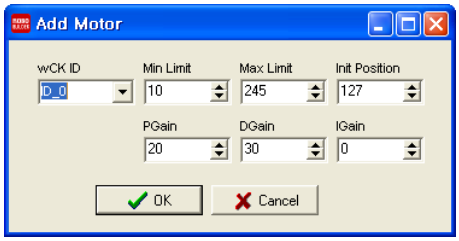


※ Go to step 6) in case you choose a standard platform such as **HUNO, DINO, DOGY**.
It is recommended that users don't change the Config options.

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.

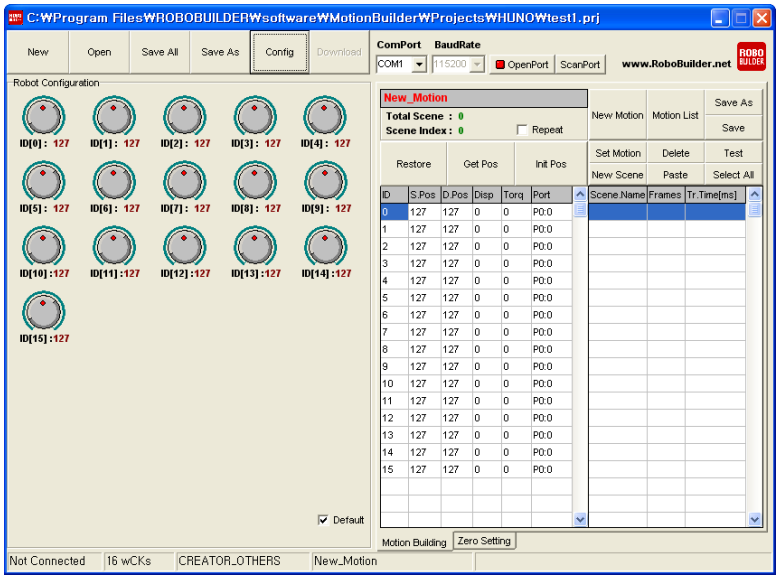
- **wCK ID** : Each wCK module has its own ID so that the controller can send commands to the wCK with a specific ID. This ID should match with the actual ID of the robot.
  <wCK ID picture>

- **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.

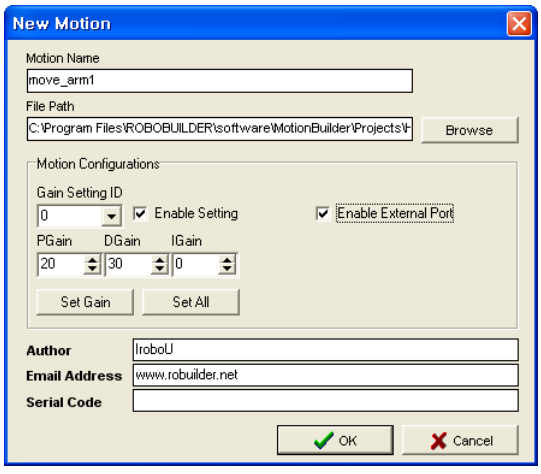
- **Init Position** : The initial position of the wCK module. This decides the start position of the first scene of a motion file. In many cases of creating or modifying a motion, the motion starts with a particular posture and ends with that same posture. Therefore, setting this Init Position well helps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

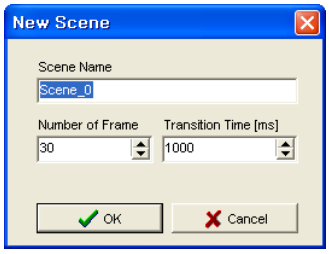
- **File Path** : The directory path where the motion is save. The path is by default same with the directory set in step 2). This path doesn't need to be changed.

- **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.

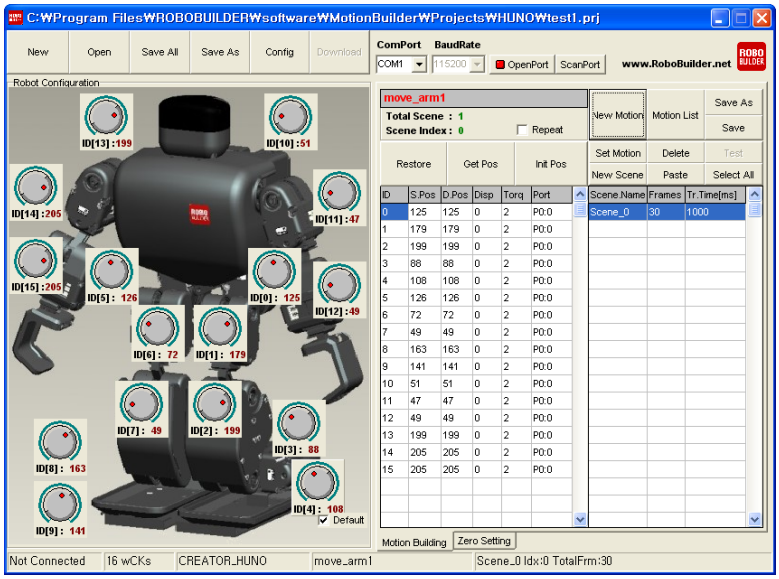
7) The window appears so that you can input the scene properties. Input the options and click OK button to finish.



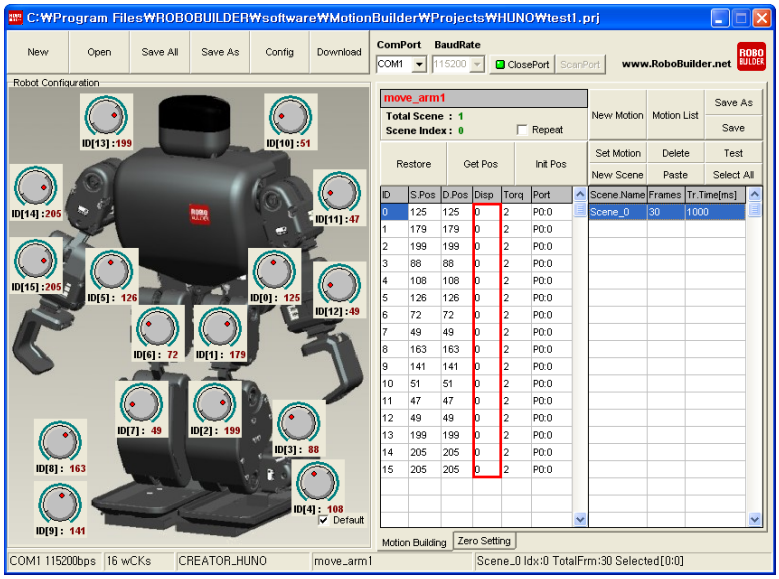
The input options are as below.

- **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 excution 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.

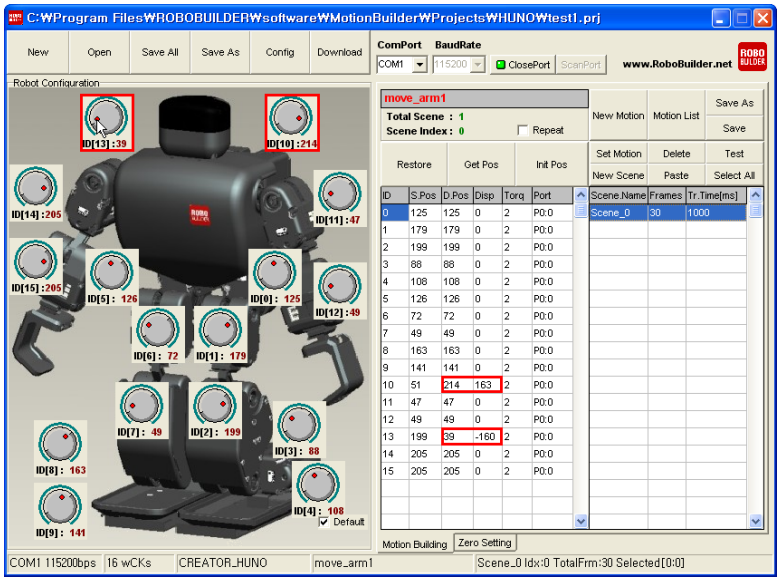


8) Connect com port as described in chapter 2.2 3). COM port connection can be done in any step of 1)~8).



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.

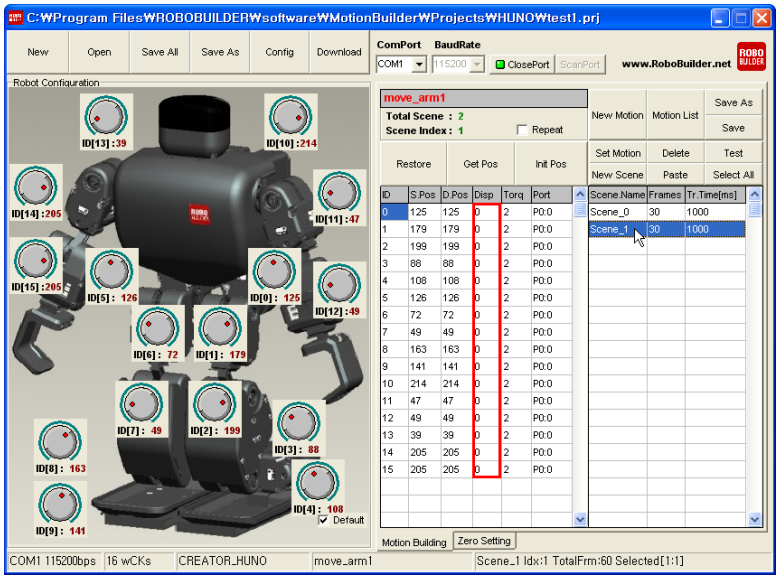
9) Let's make the robot hold up the arms by dragging the jog dial of ID10 and ID13 in the 'Robot Configuration' area.



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

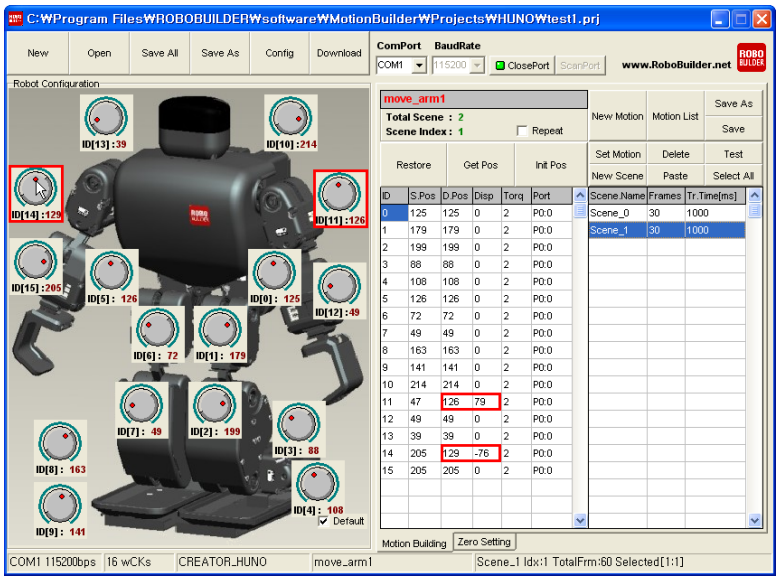
※ If you click the Restore button while you are modifying the destination position of a scene, the Disp. turns to zero and the robot moves to the start position.

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.



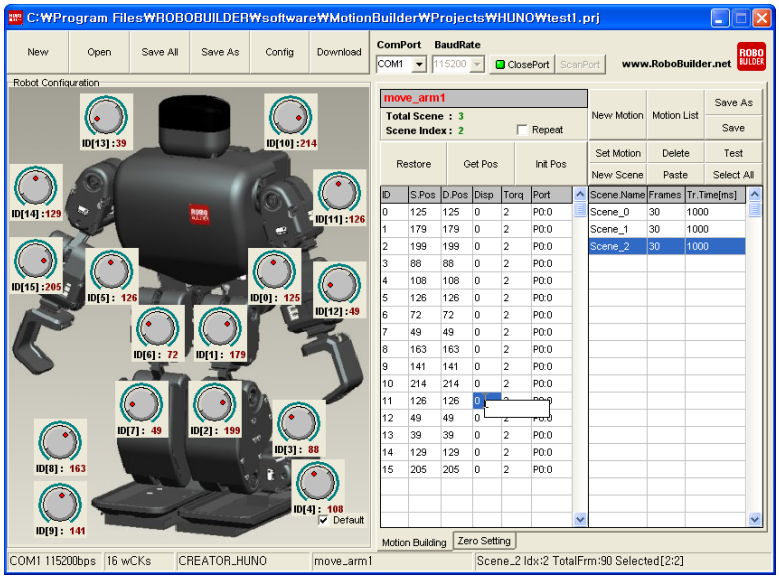
You can see the Disp. Values of the second scene are all set to zero.

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.

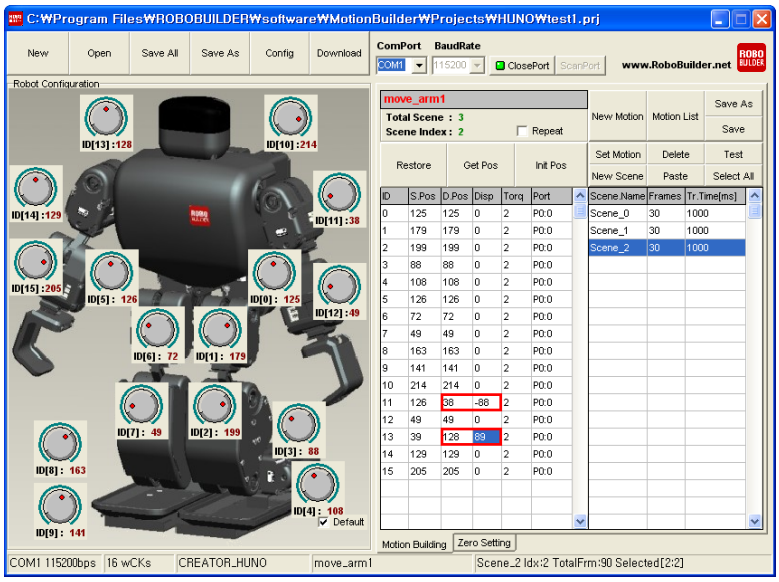


12) Follow step 10~11) and add the third scene. Let's type in the Disp. Value directly for ID11 and ID14 to make the robot bend the arms inside to the center.

Double click the Disp. of the ID 11의 Disp. and the text key in window opens.



Type in -88 and press Enter. Type in 89 for ID13.

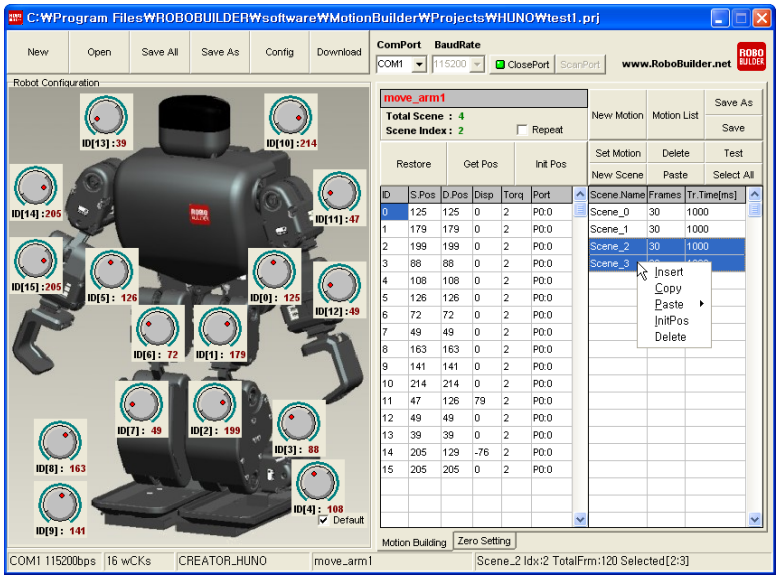


13) The scenes made until now are as below.

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

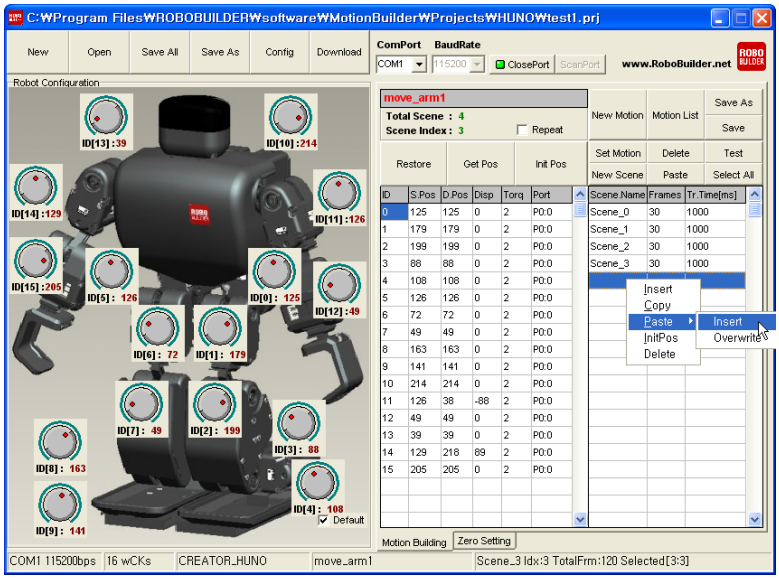
In summary, the robot will hold up the arms and strech out the arms to the side and then bend the arms inside to the center of the chest.

14) Now let's make the robot strech 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.

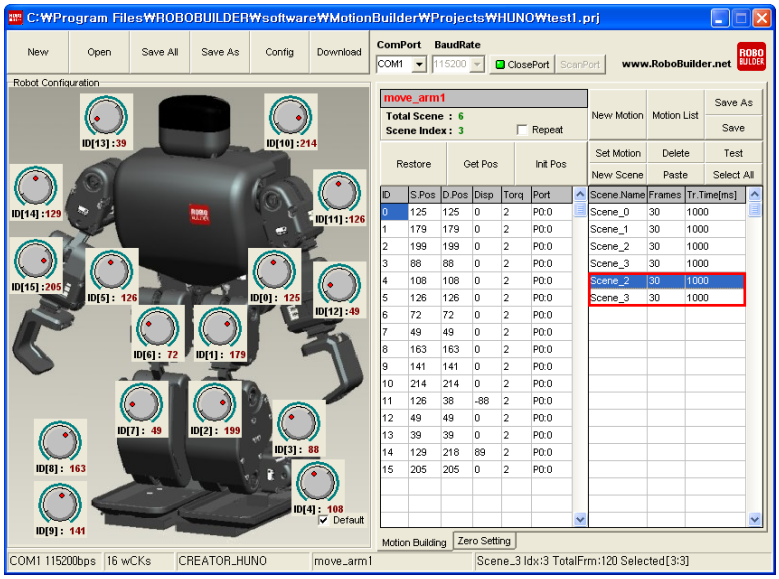


Choose **Copy** here.

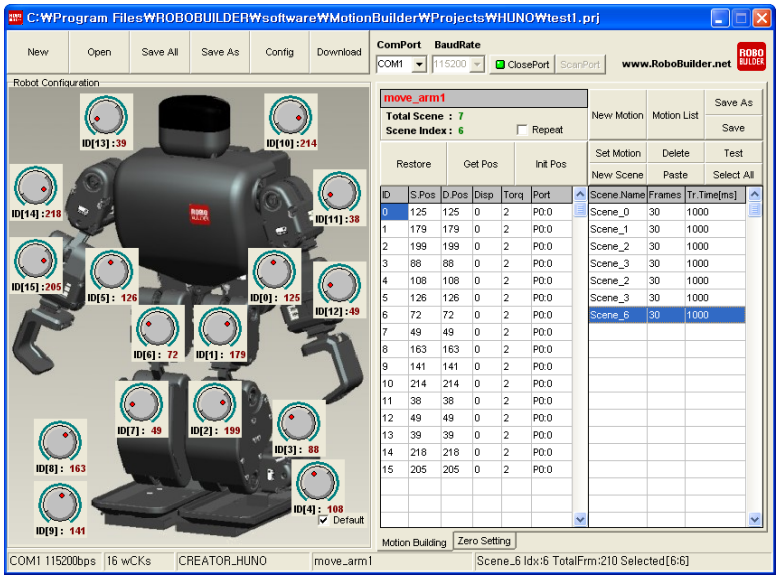
15) Select the empty line below **Scene_3** and right click the mouse to choose **Paste/Insert**.



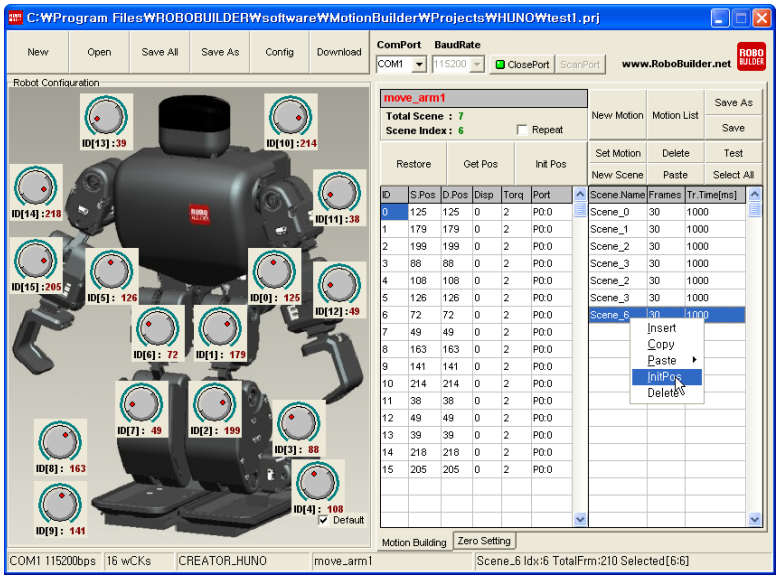
As seen below, the Scene_2 and Scene_3 are pasted now.



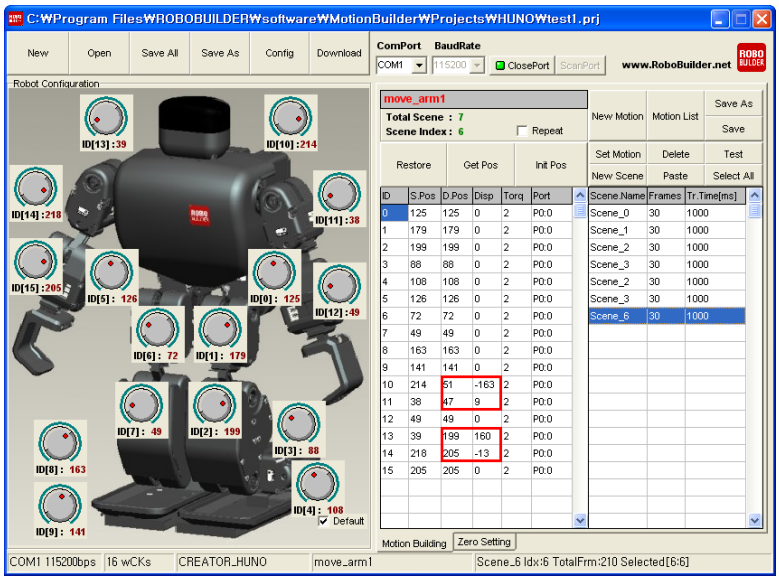
16) Add the last scene by following the step 11~12). The scene name is 'Scene_6'.



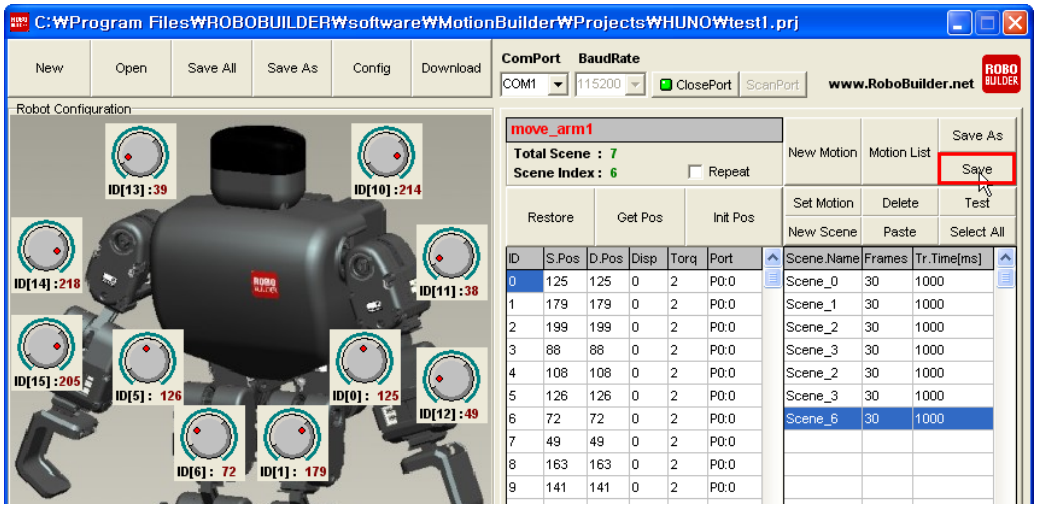
17) It is helpful for the robot to move stably that you set the last scene ends with the initial posture. Click **Scene_6** and right click the mouse to choose **InitPos**.



As seen below, the destination position of the Scene_6 is initial posture.

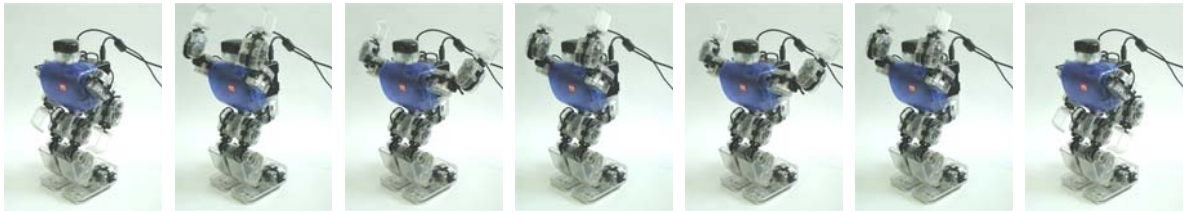


18) Click Save button to save the motion file 'move_arm1.rbm'.



※ 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.

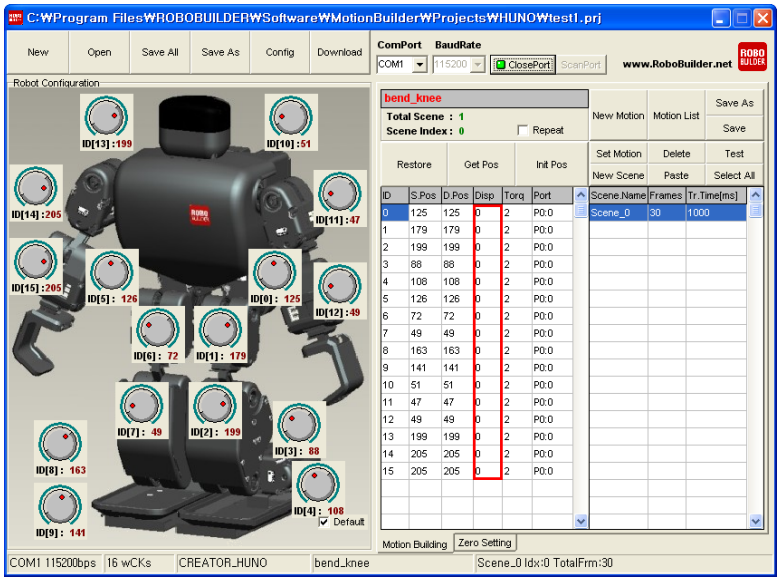
19) Let's test the created motion. Click Select All button and click Test button.



5.2. Example Programming B: Bend and straighten the knees(method c)

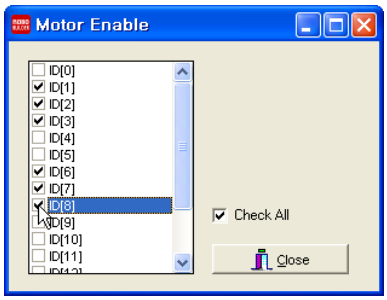
Let's use method c (motion teaching method) and make the robot bend and straighten the knees.

1) Follow chapter **5.1 1~8) and create a motion named** 'bend_knee'. Connect the com port..



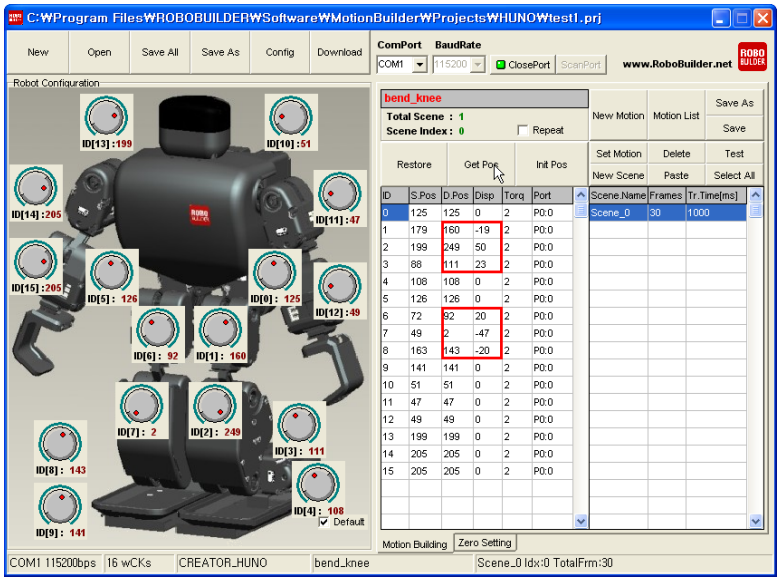
As seen above, the Disp. Of all wCK modules are set to zero.

2) Click Get Pos button and this window appears.



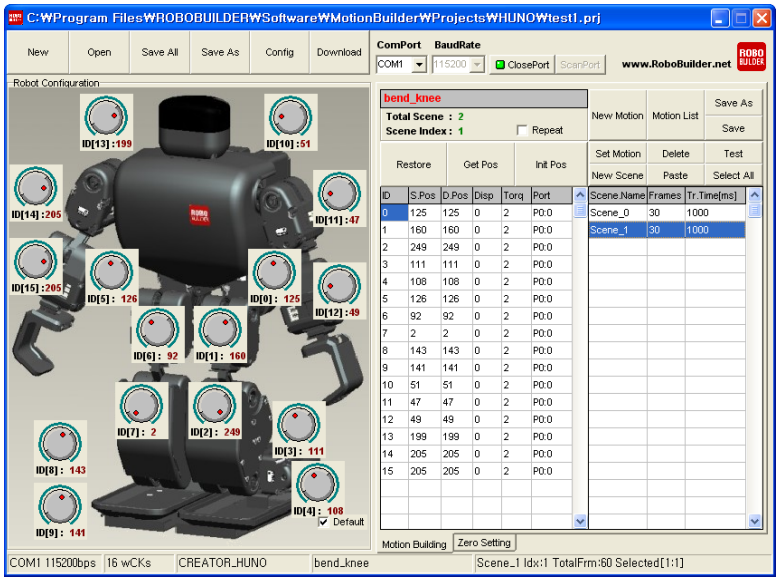
In this window, you select the wCK modules that you want to adjust.
When the robot bend its knees, the robot will fall back if it bend only the knees. Therefore, select the six IDs so that it can flexibly bend its ankles and hip joints together.

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 Catpure and you will see the data changed as shown below.

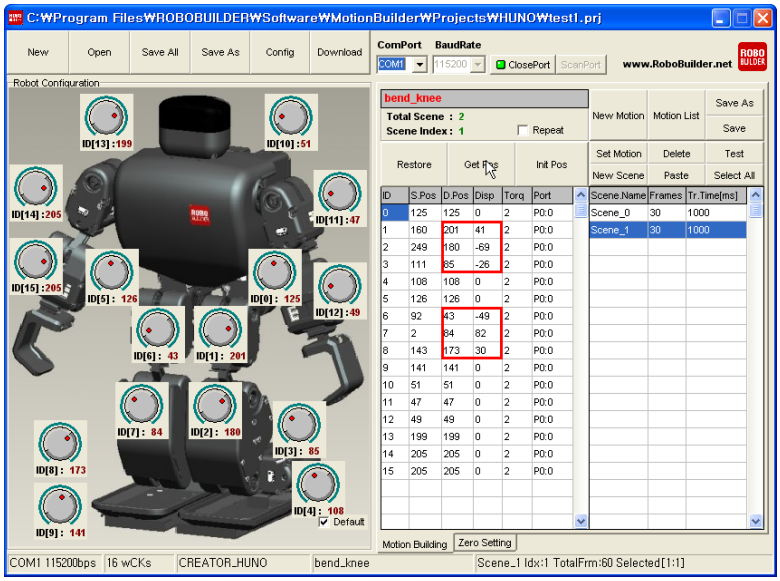


※ Caution : In case communication error occurs during the Capture (unplugged cable, control box power off etc), the MotionBuilder may not work properly. If the program window expands press Alt+F4 to close the program. Check the communication status and try again.

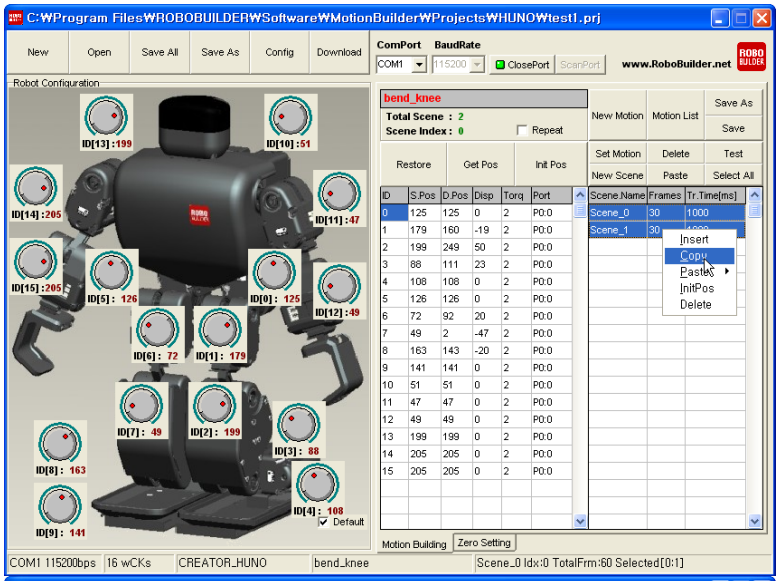
4) Click New Scene button and add a scene. Use ↓ key or the mouse to choose the added second scene.

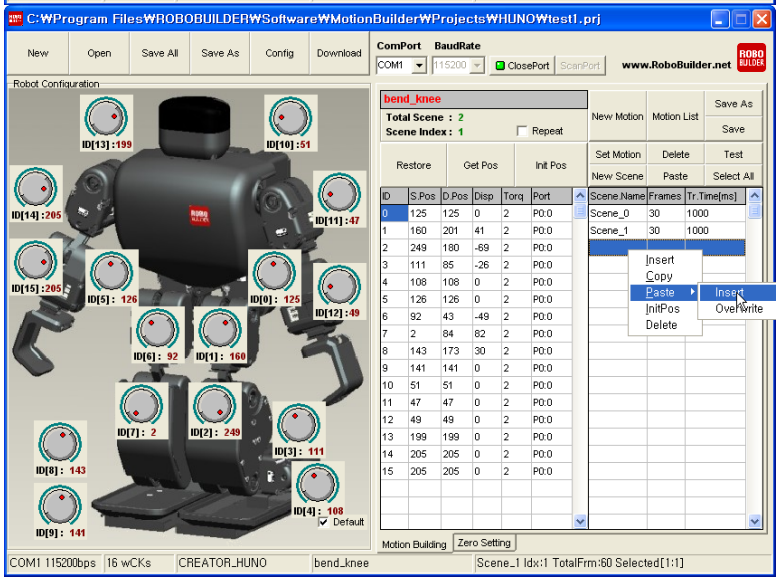


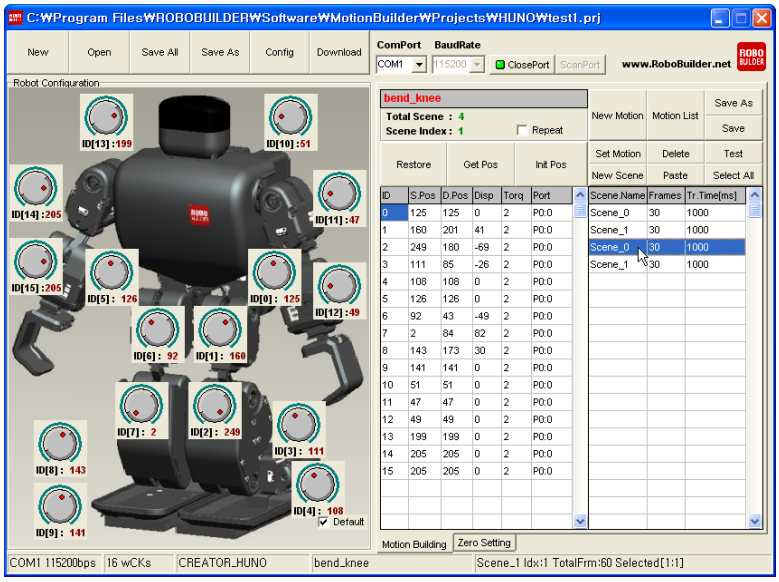
5) Let's capture the robot when it straighten the knees. Click Get Pos button and select the same six IDs(1, 2, 3, 6, 7, 8). Touch and adjust the robot to make it straightent the knees and stand straight upward. Click Capture button then you will see the data changed as shown below.



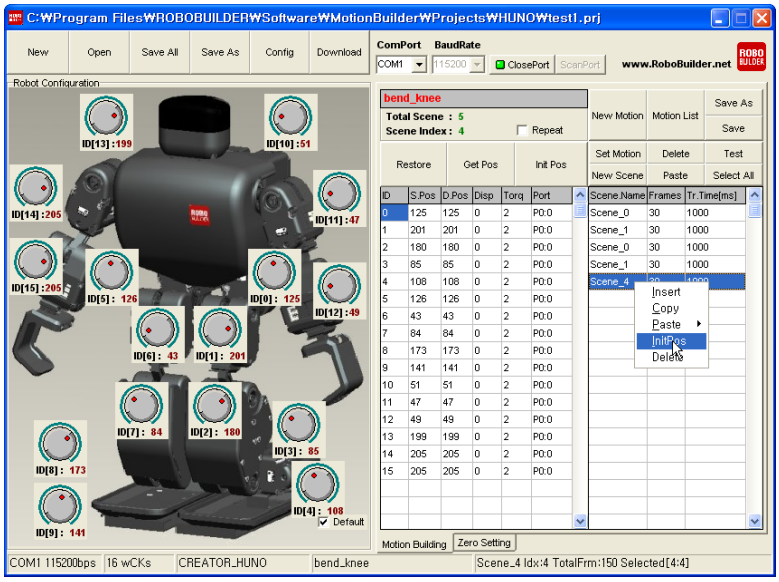
6) Now two scenes have been completed(bending the knees, straightening the knees). Now let's copy and paste the two scenes so that the robot can repeat the knee bending motion. As explained in chapter **5.1 14~15)** copy and paste the Scene_0 and Scene_1.







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.

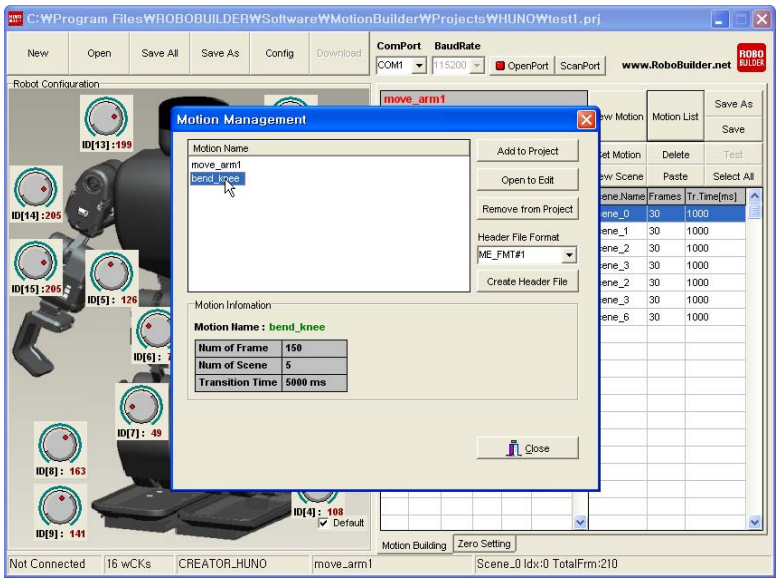
9) Let's test the created motion. Click Select All button and click Test button.

5.3. Changing the speed of a motion

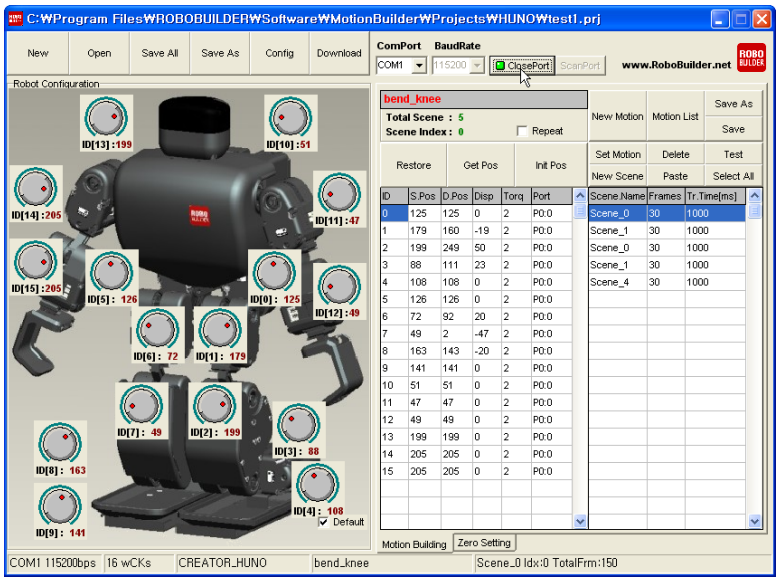
In previous chapters, the frame and transition time were default values(frame 30, transition time 1000ms). In this chapter, let's study how to change frame and transition time.

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'.

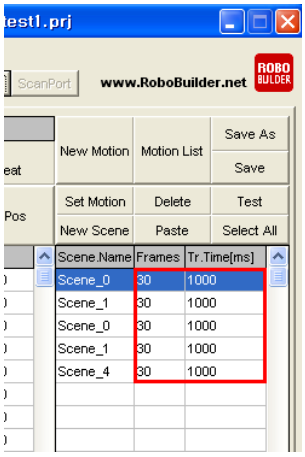
2) Click Motion List button and double click 'bend_knee'.



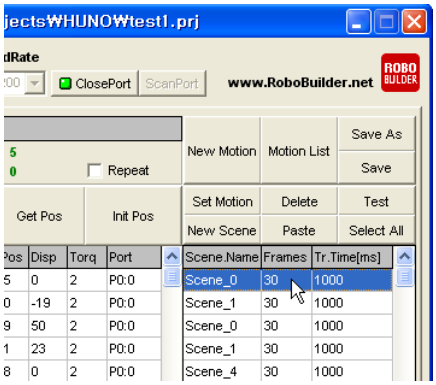
The data for 'bend_knee.rbm' appear. Connect RoboBuilder to PC and open the com port.



3) You can see the frame 30, Transition time 1000ms in the scene editing are.

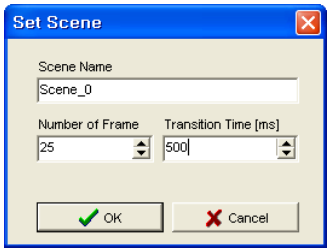


4) Let's increase the speed of Scene_0. Double click the scene.

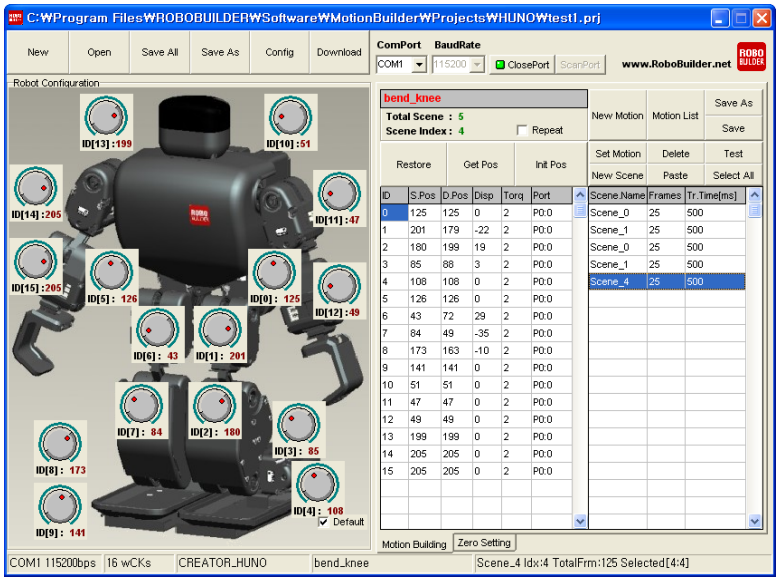


The scene's property window appears. Input 25 for the Number of Frame and 500 for the Transition Time and click OK. The frame ranges from 1 up to 100 and the minimum transition time per frame is 20ms. Therefore, the transition time is affected by frame number and ranges from 20 up to 6000ms. (refer to 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 can not change it to 500 without changing the Transition Time first.



5) As shown in the screen below, the frames and transition times for all scens 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.

6. Merging motion files

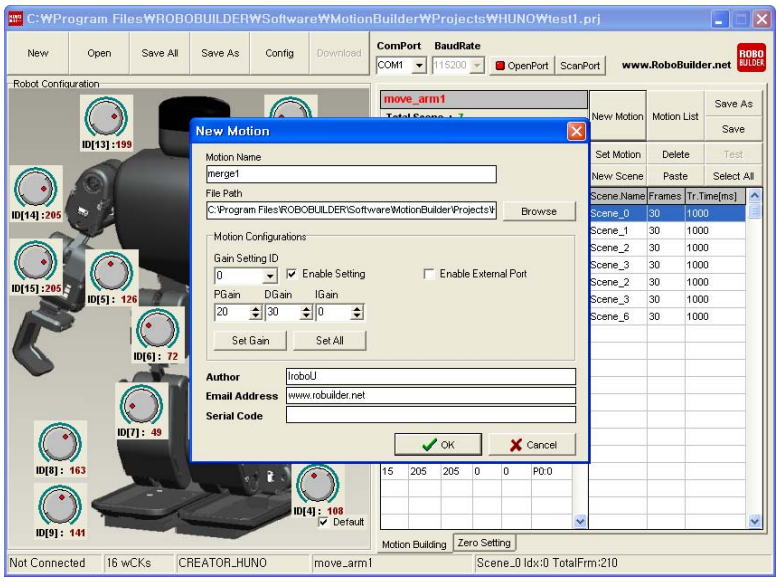
This chapter explains how to merge multiple motions.

6.1. move_arm1.rbm + bend_knee.rbm

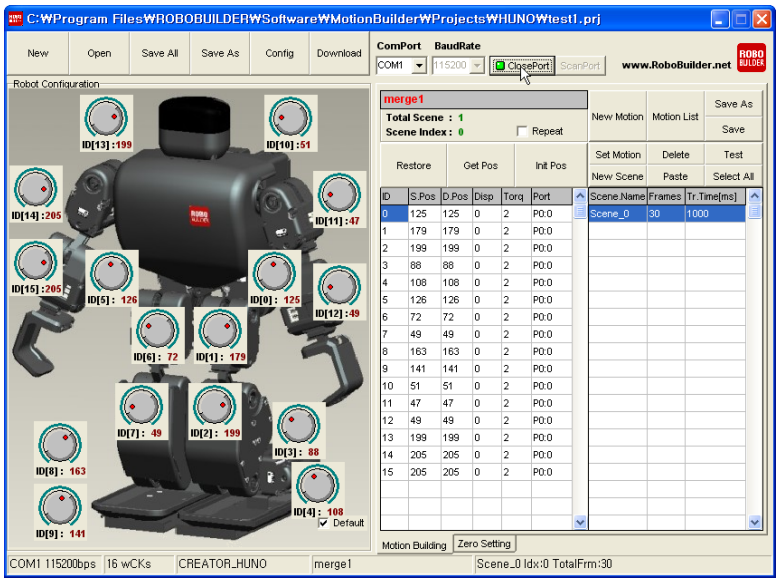
Let's combine 'move_arm1.rbm' and 'bend_knee.rbm' to make a new motion 'merge1.rbm'.

1) Run MotionBuilder and click Open button to open the project 'test1.prj'.

2) Click New Motion button.

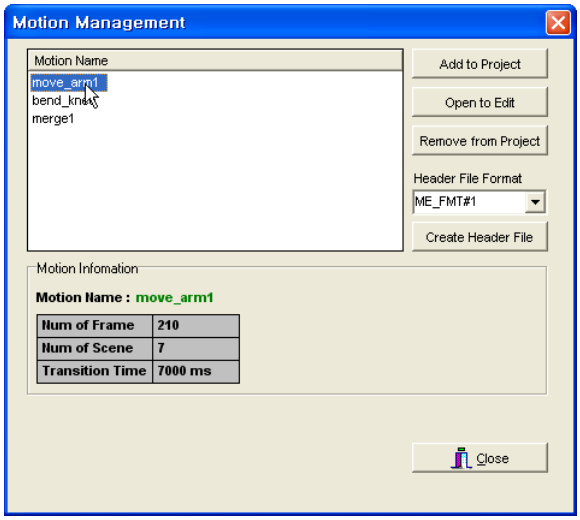


Set the Motion Name as 'merge1' and click OK button. If the new scene window appears, leave the default values and click OK and then connect com port.

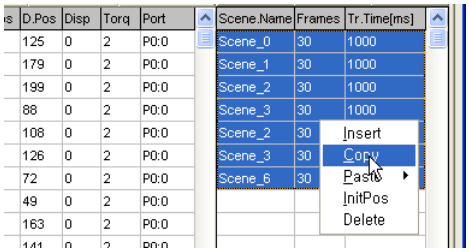


3) Click Save All button to save 'test1.prj' and 'merge1.rbm'.

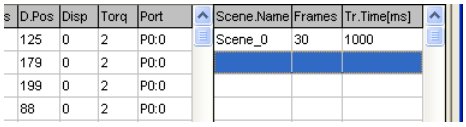
4) Click Motion List button and double click ‘move_arm1’.



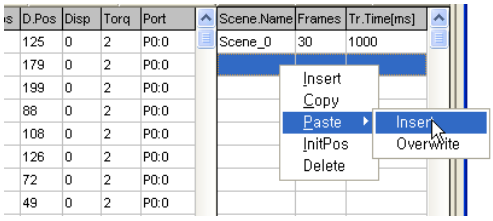
5) Click Select All button and select all scenes of the ‘move_arm1.rbm’. Right click the mouse and choose **Copy**.



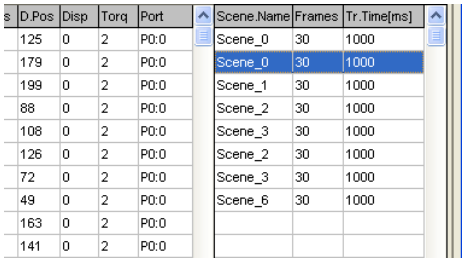
6) Click Motion List button and double click ‘merge1’.



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

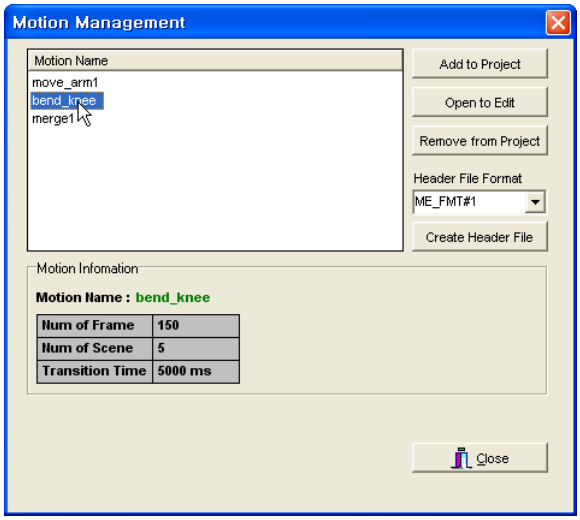


As seen below, the scenes of ‘move_arm1.rbm’ have been added to ‘merge1.rbm’.

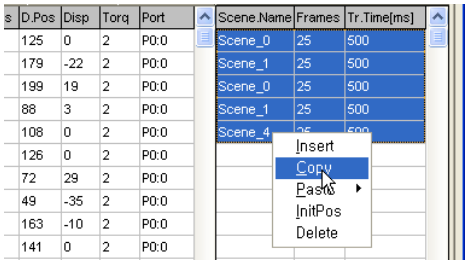


7) Click Save button to save the 'merge1.rbm’.

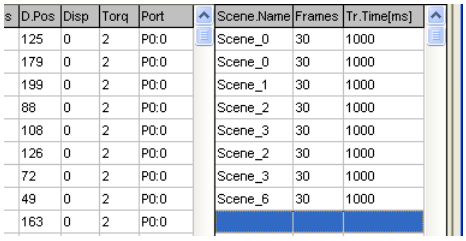
8) Click Motion List button and double click ‘bend_knee’.



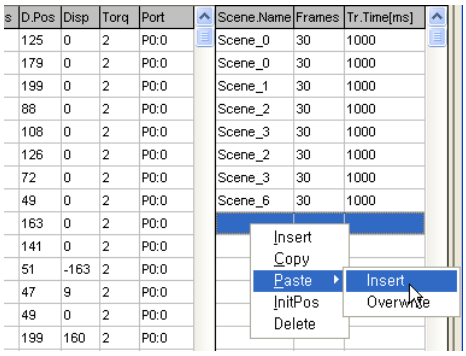
9) Click Select All to select all scenes of ‘bend_knee.rbm’. Right click and choose **Copy**.



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**.

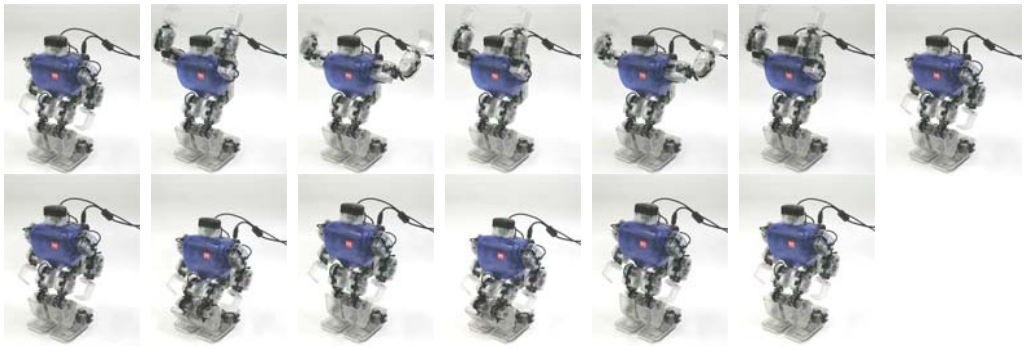


As shown below, all scenes of ‘bend_knee.rbm’ have been added to ‘merge1.rbm’.

s	D.Pos	Disp	Torq	Port	Scene.Name	Frames	Tr.Time[ms]
	125	0	2	P0:0	Scene_0	30	1000
	179	0	2	P0:0	Scene_0	30	1000
	199	0	2	P0:0	Scene_1	30	1000
	88	0	2	P0:0	Scene_2	30	1000
	108	0	2	P0:0	Scene_3	30	1000
	126	0	2	P0:0	Scene_2	30	1000
	72	0	2	P0:0	Scene_3	30	1000
	49	0	2	P0:0	Scene_6	30	1000
	163	0	2	P0:0	Scene_0	25	500
	141	0	2	P0:0	Scene_1	25	500
	51	-163	2	P0:0	Scene_0	25	500
	47	9	2	P0:0	Scene_1	25	500
	49	0	2	P0:0	Scene_4	25	500
	199	160	2	P0:0			

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 buttion. You can see the ‘move_arm1.rbm’ and ’bend_knee.rbm’ are merged now.



7. Using the wheel mode of wCK module

This chapter explains how to use the 360 degree wheel mode of wCK module.

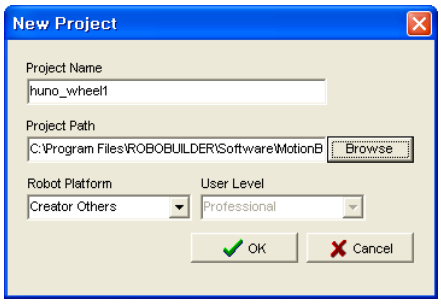
7.1. Adding a turning shield to HUNO's left arm

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

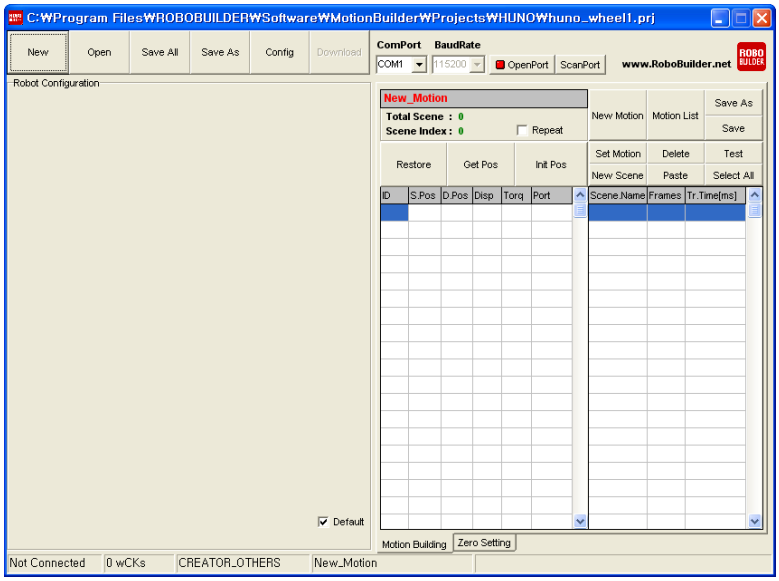
1) Disconnect ID12 wCK module from HUNO as below.

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

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

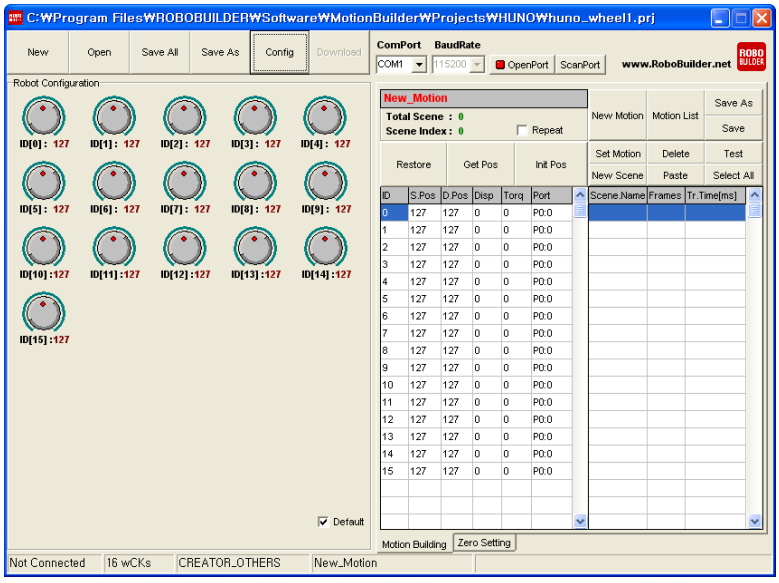


4) Set the Project Name as 'huno_wheel1' and click Browse button to set the save path. Choose '**Creator Others**' for Robot Platform and click OK to finish.



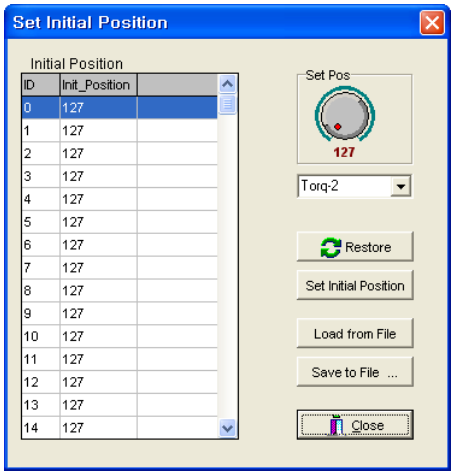
※ Standard platform robots(HUNO, DINO, DOGY) don't support the 360 degree wheel mode.

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.



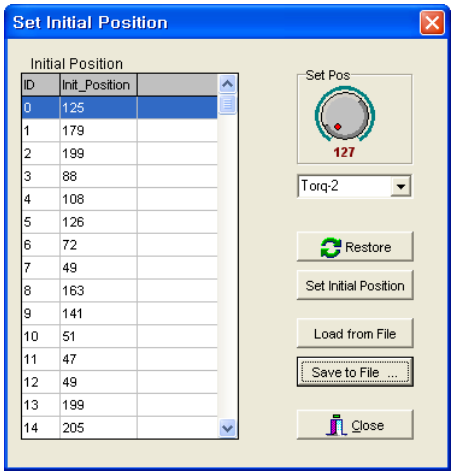
All initial positions are set to 127 now. But this should be changed because it is not appropriate for HUNO.

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 posotion file (*.mip)

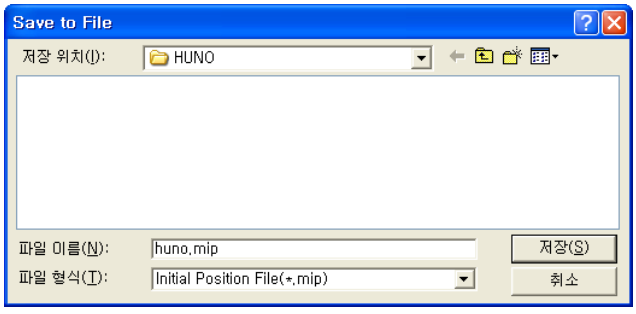
8) As the robot's structure is almost same with HUNO, let's call up the initial position file fo HUNO.

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'.

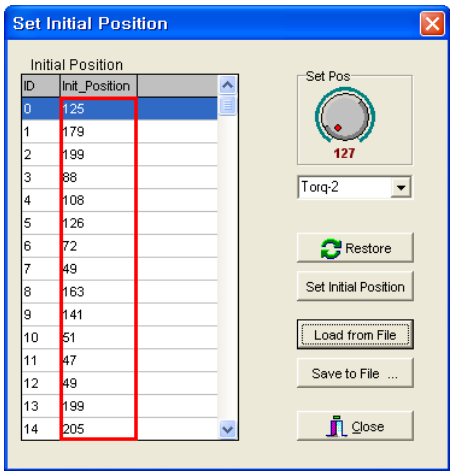


Click Close button to close 'Set Initial Postion'.

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.

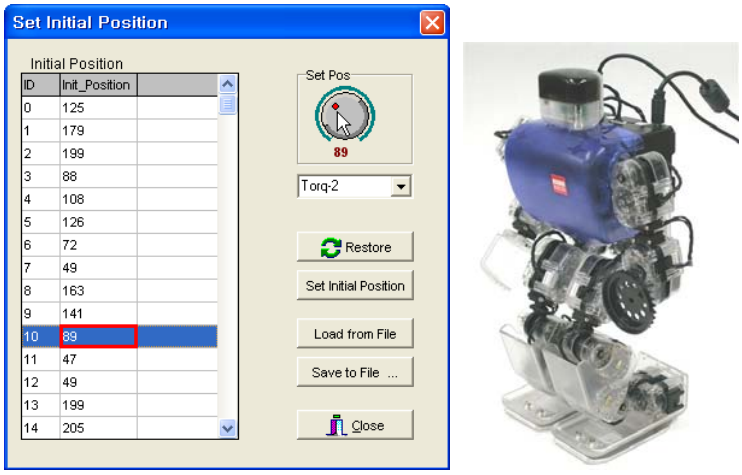
Click OK button for the New Scene window.

Connect COM port.

12) You can click Test button to test the intitial posture of the first scene.

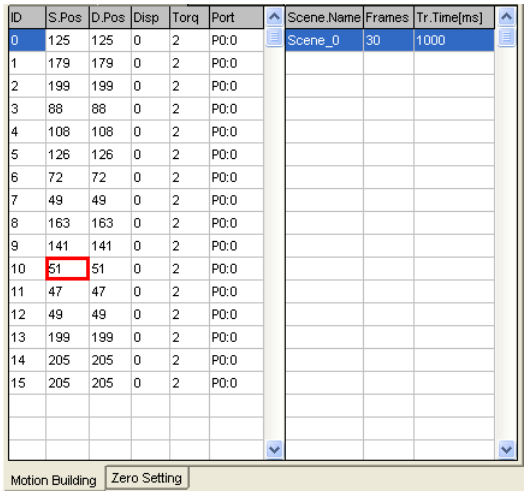
13) You can see the left arm is not very natural because of the modification. Let's change the initial posture.

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.



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

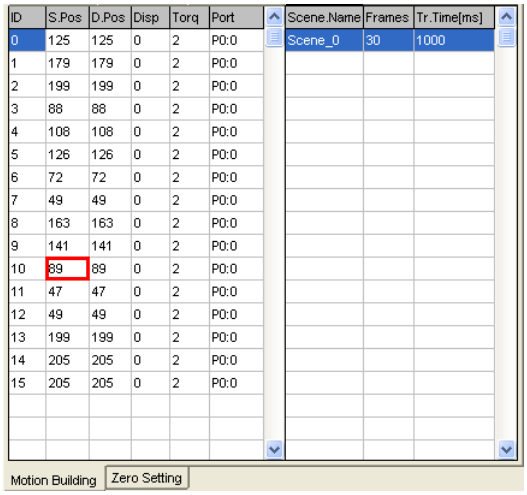
14) Although the initial posture is changed, the start posture of the first scene remains unchanged.



If you delete the first scene and make new scene, the changed initial posture is applied.

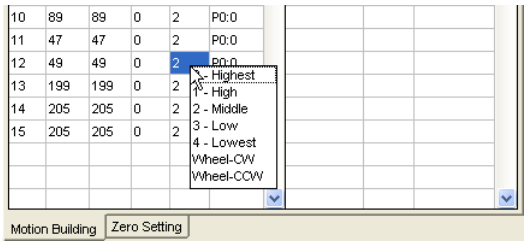
Click Delete button and the Scene_0 is deleted.

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

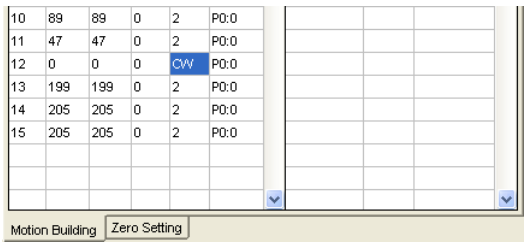


Now you can see the initial posture of the first scene has been changed.

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.

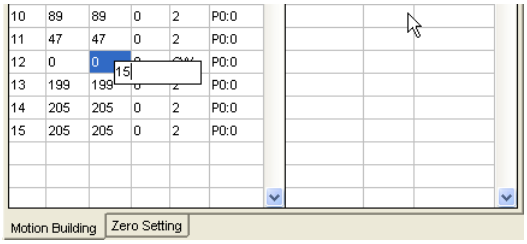


Choose 'Wheel-CW' .



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).

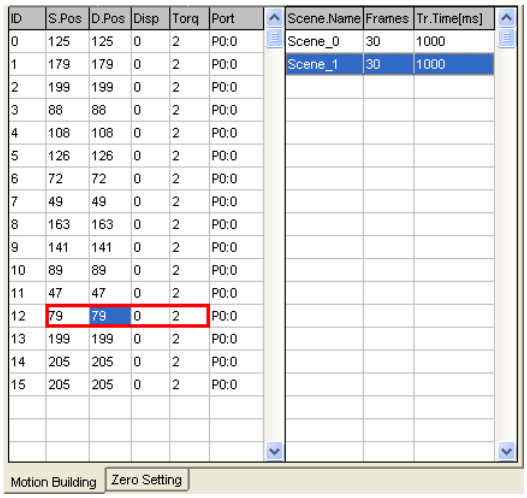
Double click the start position or destination position and input 15.



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.

ID	S.Pos	D.Pos	Disp	Torq	Port
0	125	125	0	2	P0:0
1	179	179	0	2	P0:0
2	199	199	0	2	P0:0
3	88	88	0	2	P0:0
4	108	108	0	2	P0:0
5	126	126	0	2	P0:0
6	72	72	0	2	P0:0
7	49	49	0	2	P0:0
8	163	163	0	2	P0:0
9	141	141	0	2	P0:0
10	89	89	0	2	P0:0
11	47	47	0	2	P0:0
12	0	0	0	CW	P0:0
13	199	199	0	2	P0:0
14	205	205	0	2	P0:0
15	205	205	0	2	P0:0

Scene.Name	Frames	Tr.Time[ms]
Scene_0	30	1000
Scene_1	30	1000

Motion Building | Zero Setting

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.

8. Controlling LED light(only for 5720T)

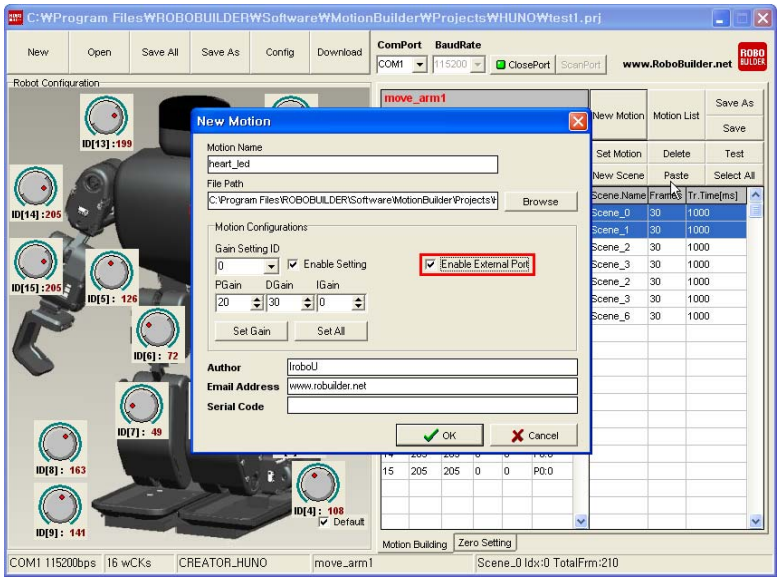
This chapter explains how to control the LEDs built in the wCK modules of 5720T kit.

8.1. Heart beating HUNO with LED

Let's learn to 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.



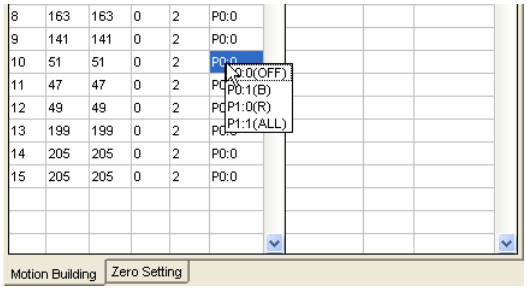
Set the Motion Name as 'heart_led'
Check the '**Enable External Port**' in order to control the external port where the LEDs are connected. Click OK button.

Click OK for the New Scene window.

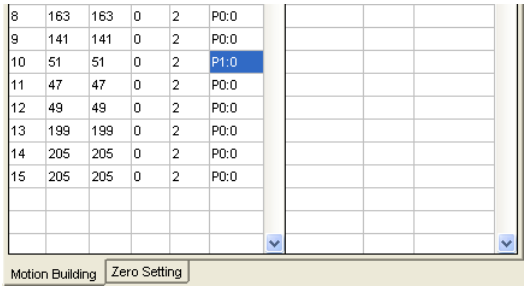
Connect COM port.

3) Let's contro the LED of ID10 in Scene_0.

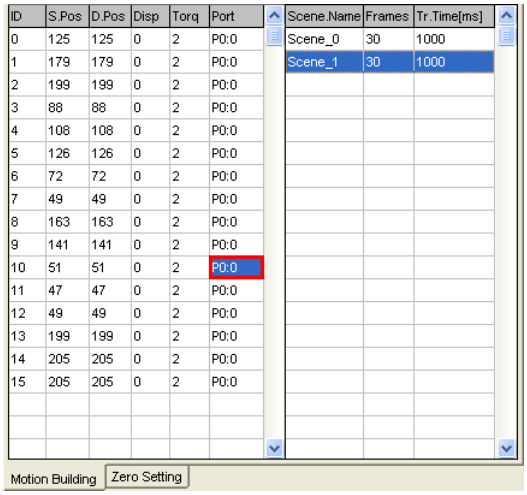
Double click 'Port' of ID10 then a small pop up appears.



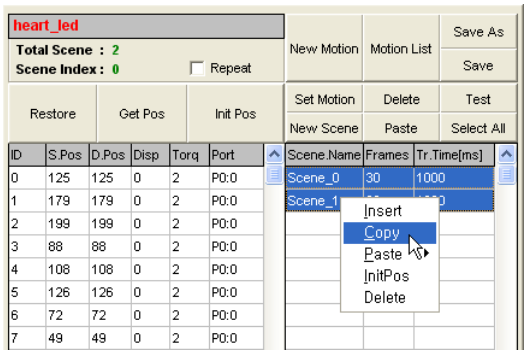
Choose '**P1:0(R)**' to turn on the red LED.



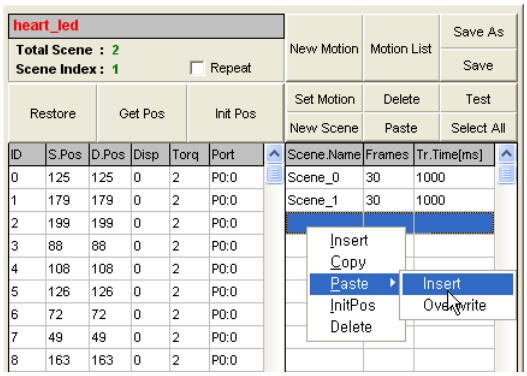
4) Click New Scene button to add Scene_1.
If you click Scene_1, you can see the 'Port' of ID10 is '**P0:0(OFF)**'.



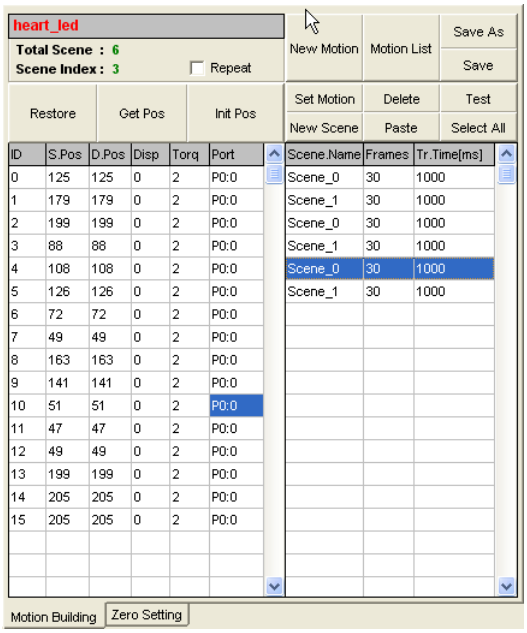
5) **In order to repeat the LED lighting two more times, select** Scene_0 and Scene_1. Right click the mouse and choose **Copy**.



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.

9. Using advanced features

This chapter explains how to use the advanced features of MotionBuilder.

9.1. Advanced use of frame and transition time

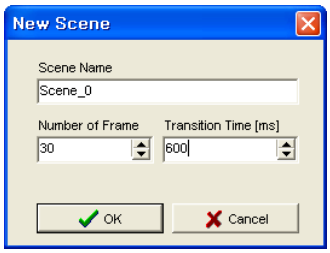
This chapter explains how to best use the transition time and frame in motion building procedure.

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 wil 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.

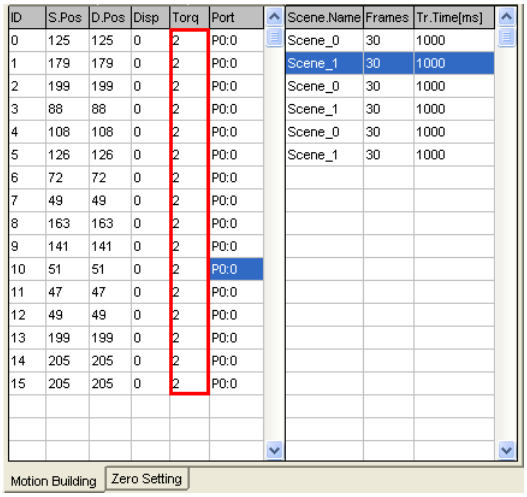


9.2. Advanced use of torque

This chapter explains how to adjust the torque level of each wCK module and help to build sophisticated motions.

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 resonse time and lower rotation force.

4) Generally spealing, 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.

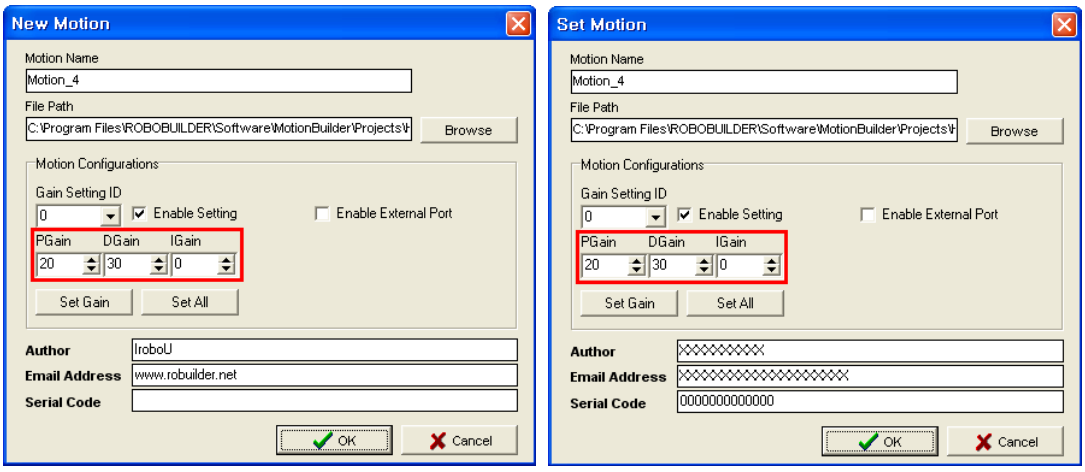
9.3. Advanced use of control gains

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 characterristics 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.



The gains can be differently set for different ID wCK modules.

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.

10. C programming with a 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.

10.1. 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

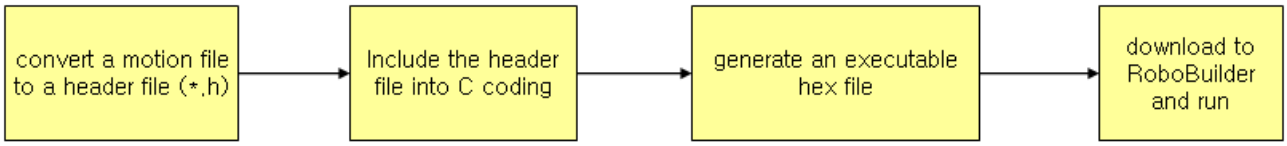
4) This material can be downloaded from the 'Tips for Specialist' page fo the homepage.

10.2. 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.

10.3. Work Flow

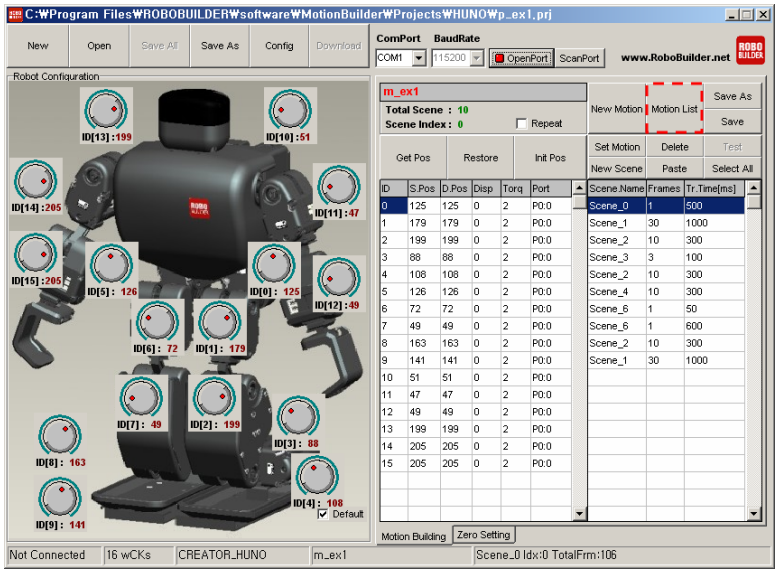


1) Convert a motion file to a header file(*.h)

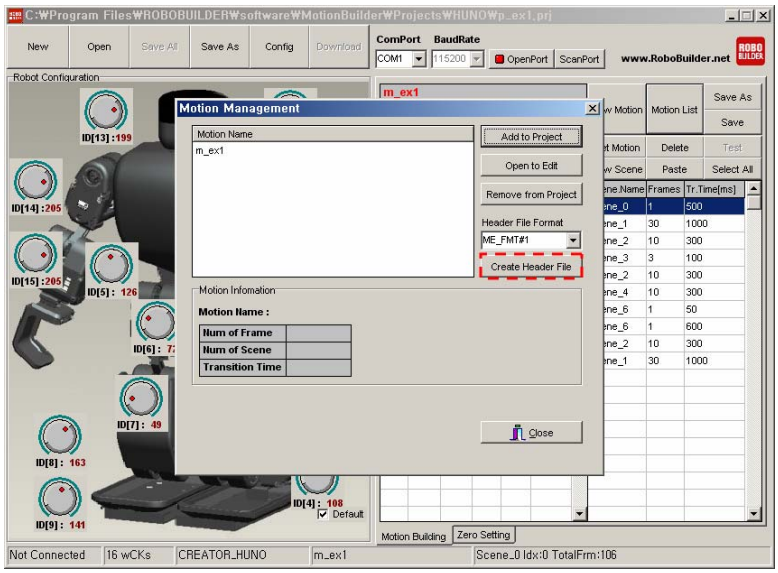
① Open MotionBuilder (version 1.10 beta or higher)

② Click Open button to open [p_ex1.prj] file in [motion_exam] folder

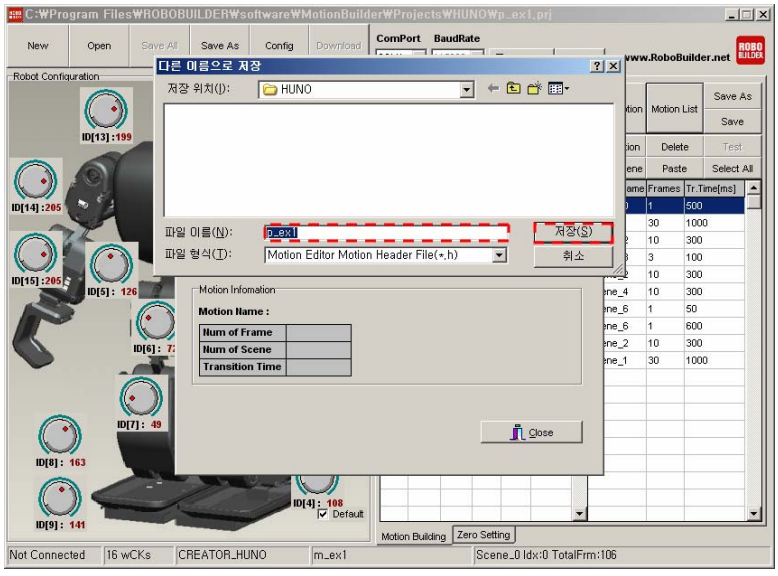
③ Click Motion List button



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



⑤ Assign a name for the header file(*.h) and push [Save] button. For example, a name “p_ex1.h” was used in this example.



⑥ A pop-up window appears and ask if you want to read the generated header file. Choose as you wish.

⑦ 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.

```
#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'

```
gpT_Table              = M_EX1_Torque;
gpE_Table              = M_EX1_Port;
gpPg_Table             = M_EX1_RuntimePGain;
gpDg_Table             = M_EX1_RuntimeDGain;
gpIg_Table             = M_EX1_RuntimeIGain;
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.

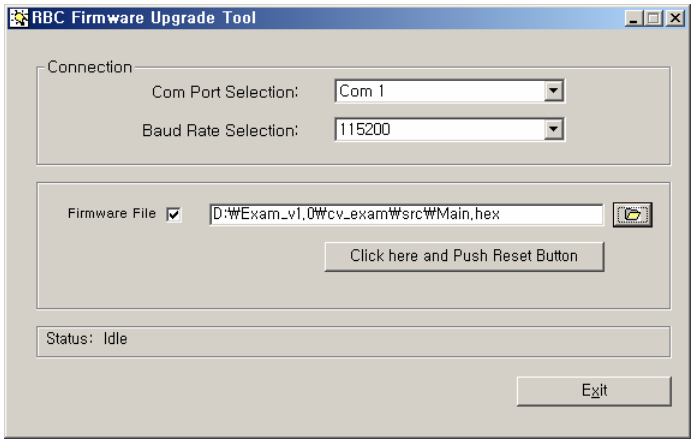
② Executable hex file generation is completed

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.

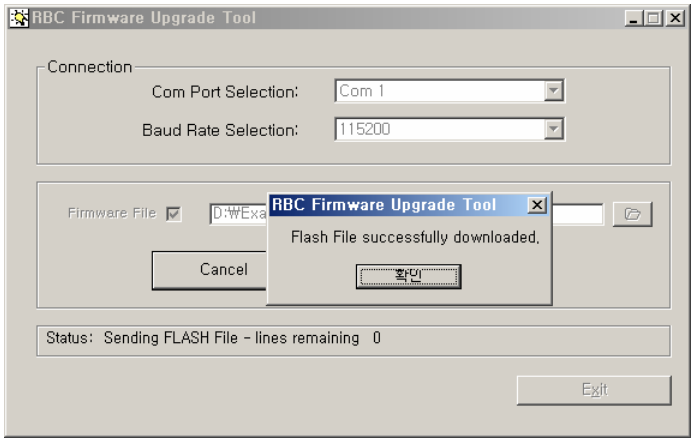
③ Set the 'Firmware File' to the “main.hex” file which was generated in “cv_exam/src” folder.



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.