HYUNDAI ROBOTICS

WARNING

INSTALLATION SHOULD ONLY BE PERFORMED BY QUALIFIED INSTALLATION PERSONNEL AND MUST CONFORM TO ALL NATIONAL AND LOCAL CODES

HD
HYUNDAI
ROBOTICS

Hyundai Robot

Hi6PA230427FMEN2

Hi6 Controller Function Manual

Palletize

HD HYUNDAI ROBOTICS

HD HYUNDAI ROBOTICS

Printed in Korea - April. 2023. 2st Edition

Contents

1. Overview 1-1

1.1. Hyundai Robotics robot palletizing software 1-2
1.2. Features 1-2

2. HRpal v1.0 2-1

2.1. HRpal flow chart 2-2
2.2. Pallet, package, and slip sheet information 2-3
2.2.1. Pallet information 2-3
2.2.2. Package and slip sheet information 2-4
2.3. Pattern creating 2-6
2.3.1. Pattern creating menu 2-6
2.3.2. Pattern creating result 2-8
2.4. Pattern changing 2-10
2.4.1. Package position changing 2-11
2.4.2. Order changing 2-13
2.4.3. Adding 2-14
2.4.4. Direction shifting 2-14
2.4.5. Deleting 2-15
2.4.6. Pallet coordinate origin setting 2-16
2.4.7. Hand open direction setting 2-17
2.4.8. Saving 2-18
2.5. Pattern managing 2-19
2.5.1. Job creating 2-20
2.5.2. Option changing 2-20
2.5.3. Pattern deleting, saving, and ending 2-21
2.6. User pattern defining 2-22

3. Palletizing Standard Program 3-1

3.1. Variable configuration 3-2
3.2. Overall flow chart 3-5
3.3. Program configuration 3-6
3.3.1. Main program 3-7
3.3.2. Vacuum On, Off 3-8
3.3.3. Process parameter setting 3-8
3.3.4. Initialization of variables and signals 3-10
3.3.5. Home position and workpiece position pose program 3-11
3.3.6. Tool control program 3-12
3.3.7. Loading pattern shift variable program for individual pallets 3-13
3.3.8. Pallet coordinate creating program 3-14
3.3.9. Pallet and slip sheet parameter setting 3-16
3.3.10. Palletizing and slip sheet motion program 3-18

3.4. Standard program basic usage.....3-19
3.4.1. Variable setting 3-19
3.4.2. Position recording.....3-19

Figure of Contents

Figure 1.1 HRpal v2.1 main screen 1-2
Figure 2.1 HRpal flow chart 2-2
Figure 2.2 Pallet information menu 2-3
Figure 2.3 Package information menu 2-4
Figure 2.4 Slip sheet information menu 2-4
Figure 2.5 Pattern creator menu 2-6
Figure 2.6 Example of the layer sagging compensation function 2-7
Figure 2.7 Pattern creating result menu configuration 2-8
Figure 2.8 Pattern menu changing 2-10
Figure 2.9 Interference occurring mutually while changing the pattern 2-11
Figure 2.10 The center of gravity of a package moving outside the pallet 2-12
Figure 2.11 Loading order changing function 2-13
Figure 2.12 Package adding function 2-14
Figure 2.13 Package direction shifting function 2-14
Figure 2.14 Package deleting 2-15
Figure 2.15 Pallet coordinate origin setting function 2-16
Figure 2.16 Palletizing grippers 2-17
Figure 2.17 Hand open direction designation function 2-17
Figure 2.18 Pattern managing function 2-19
Figure 2.19 Option changing menu 2-21
Figure 2.20 User pattern defining menu 2-22
Figure 3.1 Standard program overall flow chart 3-5
Figure 3.2 Coordinate information to be used for the palletizing process 3-9
Figure 3.3 Example of the user coordinate recording 3-14
Figure 3.4 Right hand coordinate configuration 3-14
Figure 3.5 Palletizing motion position values 3-18

Table of Contents

Table 3-1 Configuration and usage of the processInfo variable in the standard program 3-2
Table 3-2 Configuration and usage of the simulator variable in the standard program 3-3
Table 3-3 Configuration and usage of the pos variable in the standard program.....3-4
Table 3-4 Configuration and usage of the shift variable in the standard program.....3-4
Table 3-5 Job numbers and roles in the standard program 3-6

HYUNDAI ROBOTICS

1 Overview

1.1. Hyundai Robotics robot palletizing software

Robot palletizing is an operation in which workpieces are picked up and placed down onto a pallet or a rack repetitively at a certain position.

To configure an efficient robot palletizing system, the conveyors and pallets should be installed by taking into consideration the robot moving path that would affect the productivity of the line.

HRpal™ v2.1 is a program based on a PC (HRpal) to create Hyundai Robotics robot palletizing loading pattern and job programs automatically. They make it possible to carry out quick installation and pilot operation and review to configure an efficient robot palletizing system.

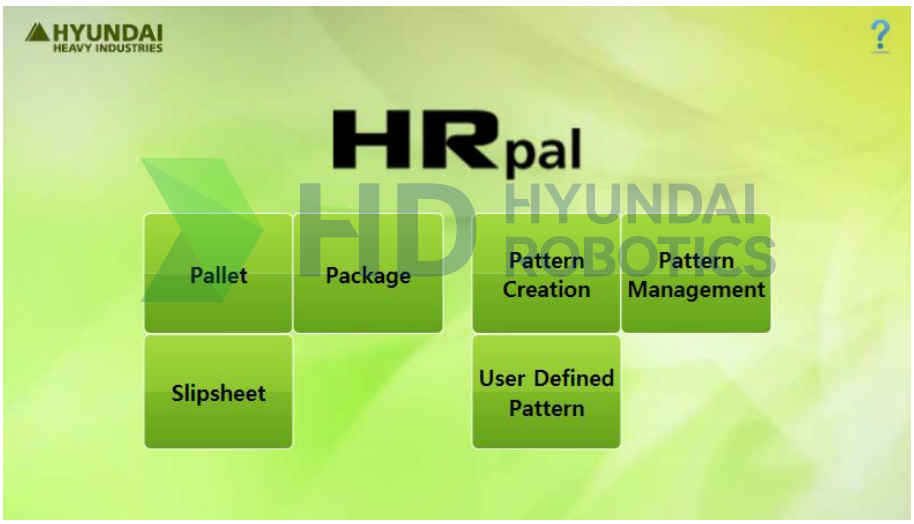


Figure 1.1 HRpal v2.1 main screen

1.2. Features

- Palletizing optimal loading pattern auto creating function (users can defined the pattern)
- Palletizing standard job program creating function
- Palletizing loading pattern managing function

HYUNDAI ROBOTICS

2

HRpal V2.1

2.1. HRpal flow chart

HRpal is PC-based application software for palletizing. The function to create a HRSpace3 simulating project file is added.

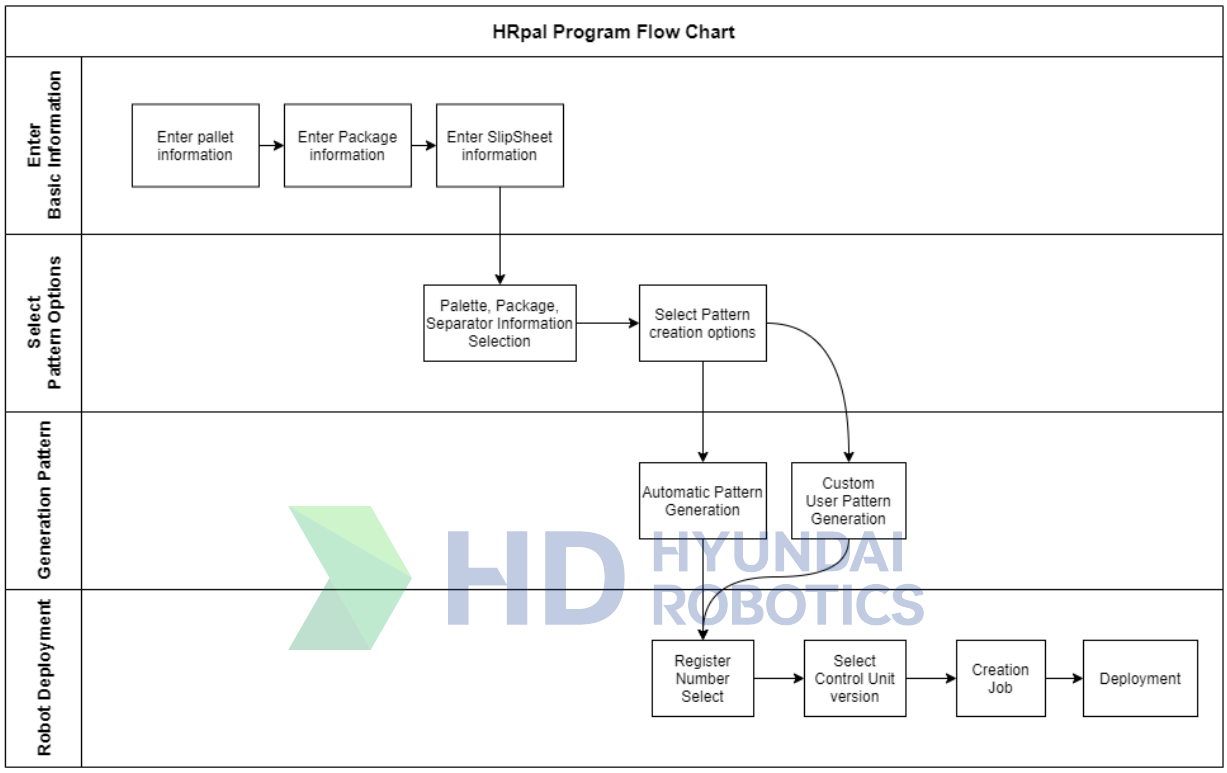


Figure 2.1 HRpal flow chart

2.2. Pallet, package, and slip sheet information

2.2.1. Pallet information

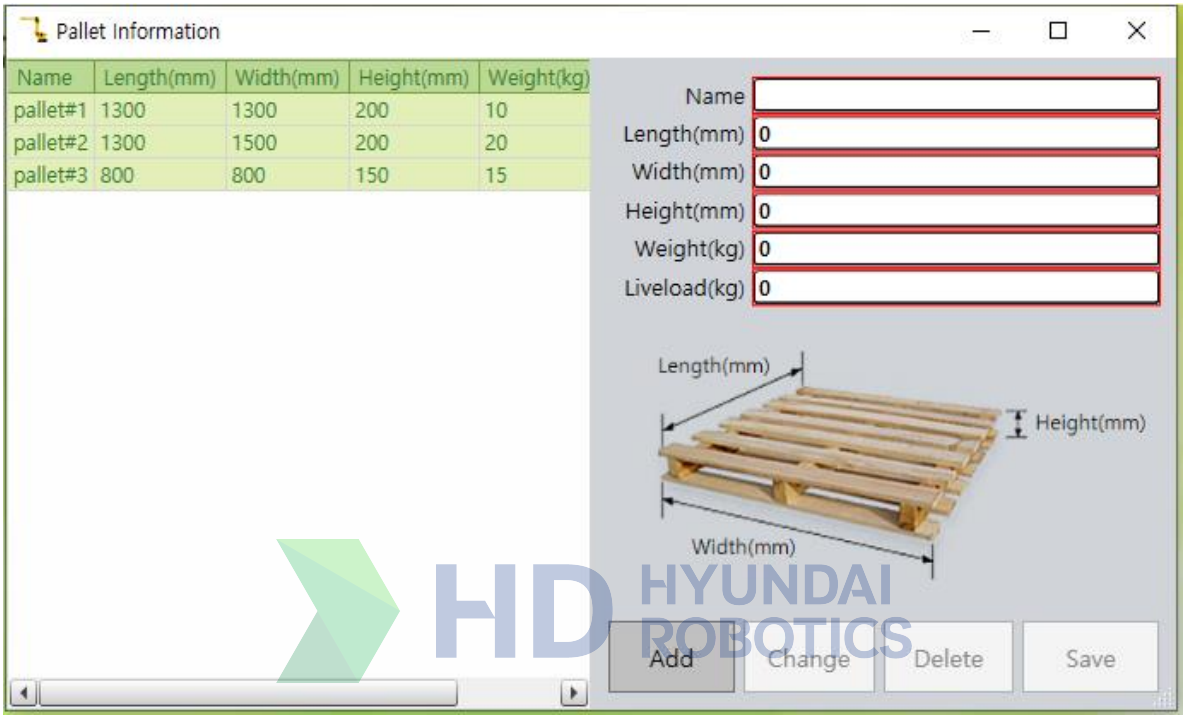


Figure 2.2 Pallet information menu

The pallet information consists of the name, length, width, height, weight and weight capacity (Figure3.2). Weight means the weight of the pallet itself and the weight capacity means the maximum load that the pallet can endure. The button menu includes Add, Change, Delete, and Save with the following functions individually

(1) Adding
The Add button is used to add new data. It will be enabled normally when all data are inputted.

(2) Changing
The Change button will be enabled when an item on the list on the left is selected. When one item of the list is selected, the relevant content will be displayed on the right side. Then, change the content and press the Change button, and the changed content will be saved in a file automatically.

(3) Deleting
Just like the Change button, the Delete button will be enabled when one item on the list on the left is selected. However, different from the Change button, the content will not be saved in a file automatically.

(4) Saving

The Save button is used to save relevant content after using the Delete button.

2.2.2. Package and slip sheet information

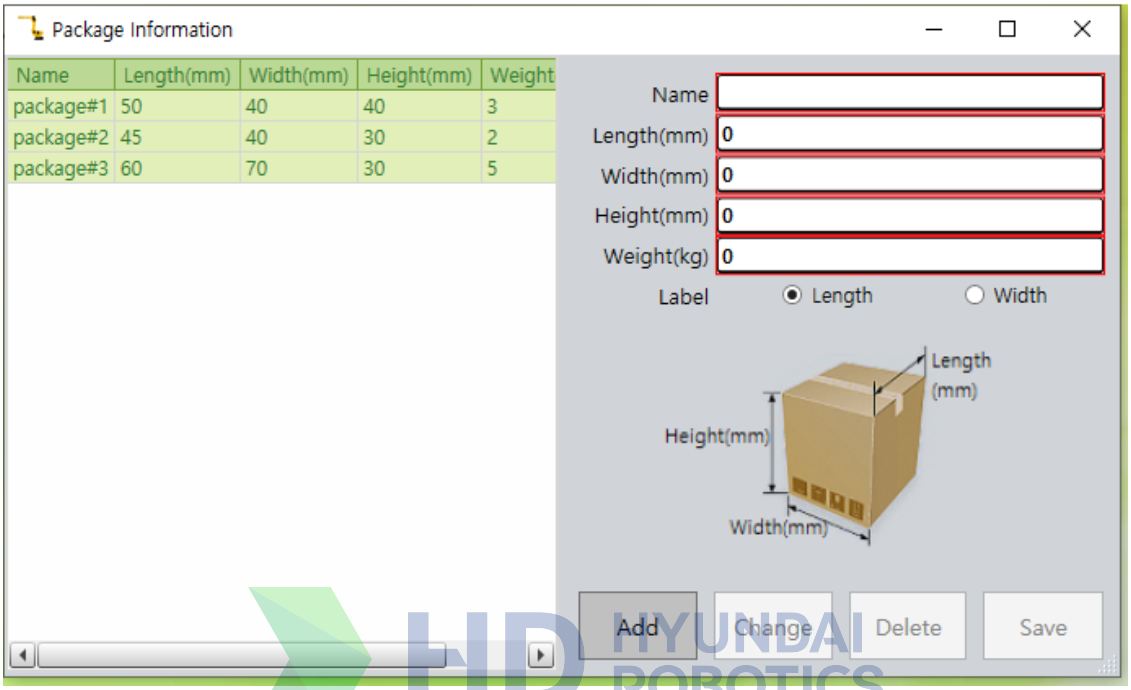


Figure 2.3 Package information menu

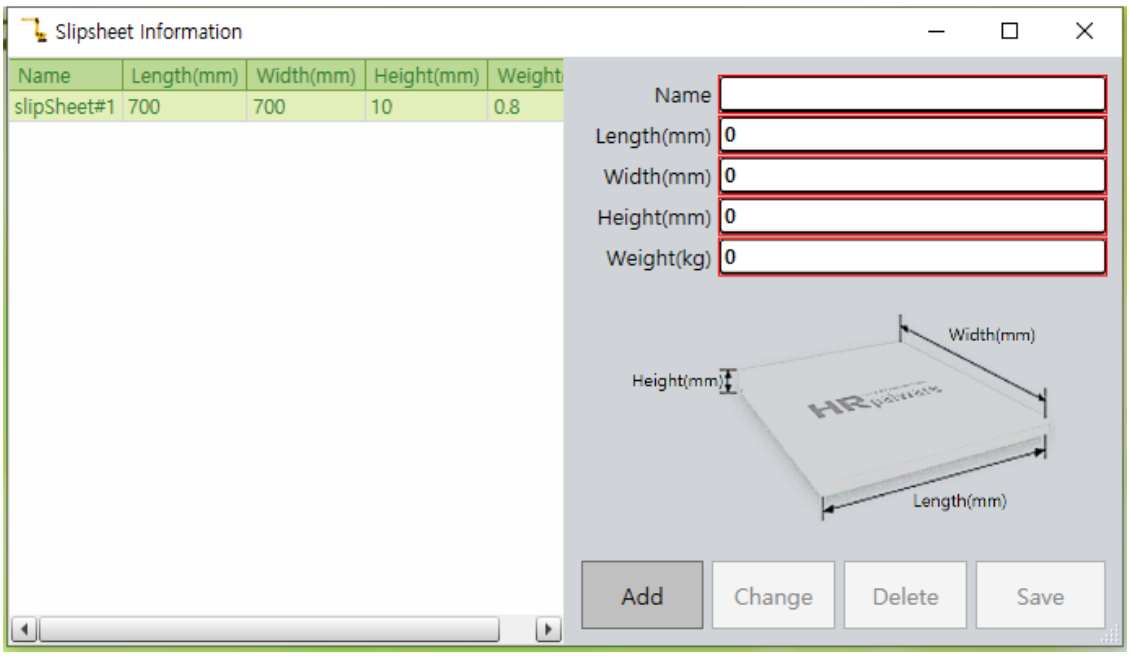


Figure 2.4 Slip sheet information menu

The package information (Figure 2.3) and the slip sheet information (Figure 2.4) can be used by inputting the name, length, width, height, and weight items. The other methods to use the information are the same with those used for the pallet information.

2.3. Pattern creating

2.3.1. Pattern creating menu

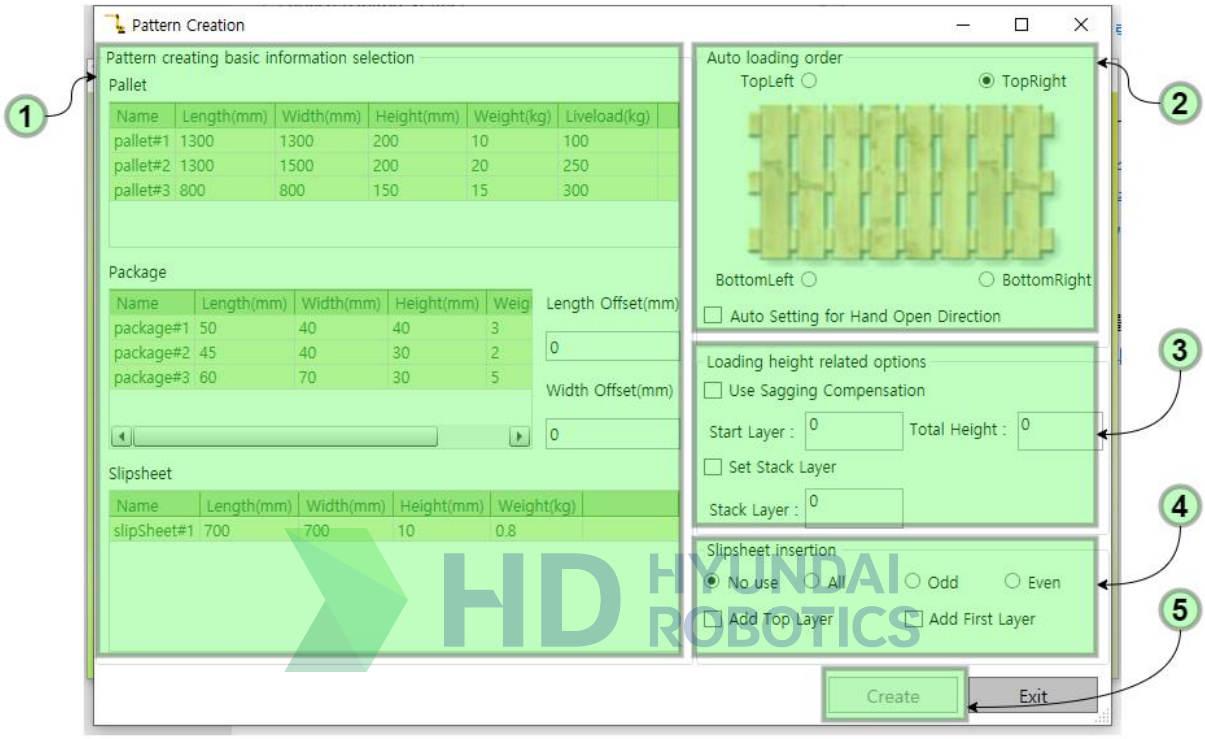


Figure 2.5 Pattern creator menu

The pattern creating menu is used to make optimal palletizing patterns by using the pallet, package, and slip sheet information (Figure 2.5). Individual menus are described below.

(1) Pattern creating basic information selection
This shows the contents added in the above Pallet, package, and slip sheet information section. The target object in which a pattern shall be created can be selected by clicking it.
The package length and width offset can be used when there is a need to put certain intervals among individual objects.

(2) Auto loading order
The auto loading order function is used to decide from which position, based on the pallet, the palletizing operation should start to be carried out. Basically, the top right side is the default.

(3) Loading height-related options

HRpal decides the layer count by taking into consideration the weight capacity of pallets. However, the layer count setting option ignores the function and allows the user's inputted layer count to be used instead.

The layer count function can be used when the layer count, such as 3 layers or 4 layers, are predefined.

The layer sagging function is an option to take measures for a palletizing operation in which deflection results from the packages are getting layered just like the case of bag palletizing. It enables the total compensation height to be applied after being divided from the layer in which the compensation starts to the last layer, as shown in Figure 2.6 below.

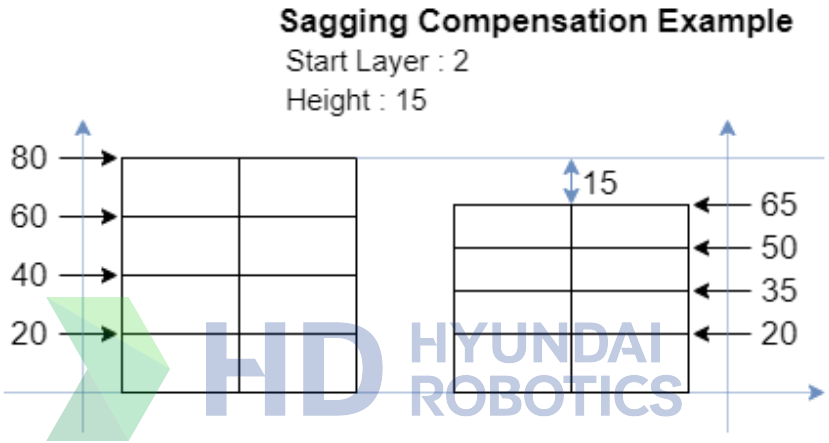


Figure 2.6 Example of the layer sagging compensation function

(4) Slip sheet insertion

In the slip sheet option, it is possible to select among the no use, all layers, odd layer, and even layer options

The first layer slip sheet insertion option is used to decide whether to place a slip sheet on the floor before starting the work.

The top layer slip sheet option is used to decide whether to cover the pallet with a slip sheet after completing the work.

(5) Pattern creating

When all the options necessary to create a pattern are selected, the pattern creating button will be enabled. When the button is pressed, an optimal loading pattern will be drawn based on the relevant data, bringing up a screen shown in (Figure 2.7)

2.3.2. Pattern creating result

The pattern creating result menu shows an optimal pattern and other patterns based on the user's selected data (Figure 2.7). When a pattern shall be used as it is, press the Save button to save the data so it is possible to extract the relevant pattern in the form of a job program through the pattern managing menu later.
The individual functions of the pattern creating result menu are shown below.

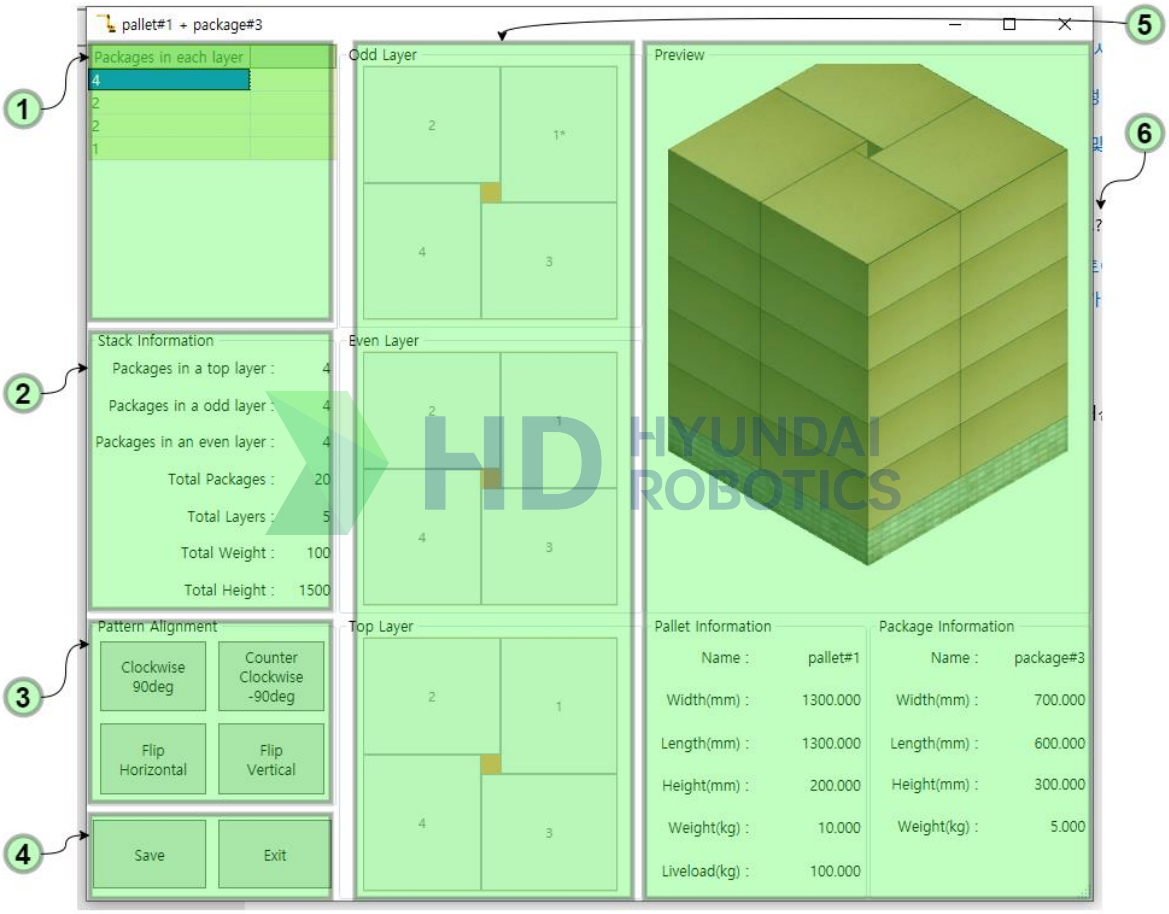


Figure 2.7 Pattern creating result menu configuration

(1) Pattern list
This shows the lists of the pattern results created based on the user's selected data.
Depending on selected data, the relevant contents, such as loading information, pattern type, and configuration type, may vary accordingly.
When the user specifically selects one of the top layer, odd layer, and even layer shown in ⑤ and selects other list, only the relevant section will change accordingly.
(Caution) If there are more packages on the top layer than on the odd and even layers, the number of packages of all the odd and even layers should be adjusted to the number of packages on the top layer.

(2) Loading information
The loading information shows specific information, such as weight and height.

(3) Pattern changing
This function is used to shift a pattern in a 90-degree direction or creating a pattern symmetrically for both the left and right sides.
When the user selects one of the top layer, odd layer, and even layer in ⑤ of (Figure 3.6) and changes the pattern, then only the relevant section will change.

(4) Saving and ending
This function is used to save and end a pattern.

(5) Each layer pattern data
This shows the pattern data of the odd layer, even layer, and top layer related to the currently selected data.
To change the pattern data of a specific layer, the mouse should be double-clicked.
The other details about changing are described in [3.4. Pattern changing].

(6) Configuration shape
This shows the expected configuration shape of the current pattern data in 3-D.
It is possible to adjust the angle that enables it to be seen by using left or right button of the mouse.
The bottom section shows the pallet and package information that is used to make the currently created pattern.

2.4. Pattern changing

The pattern changing menu (Figure 2.8) has the following functions, making possible to open a specific pattern among the odd layer, even layer, and top layer by double-clicking it.

(1) Package position changing

(2) Loading order changing

(3) Package adding and deleting

(4) Package direction shifting

(5) Hand open direction setting

(6) Pallet coordinate origin setting

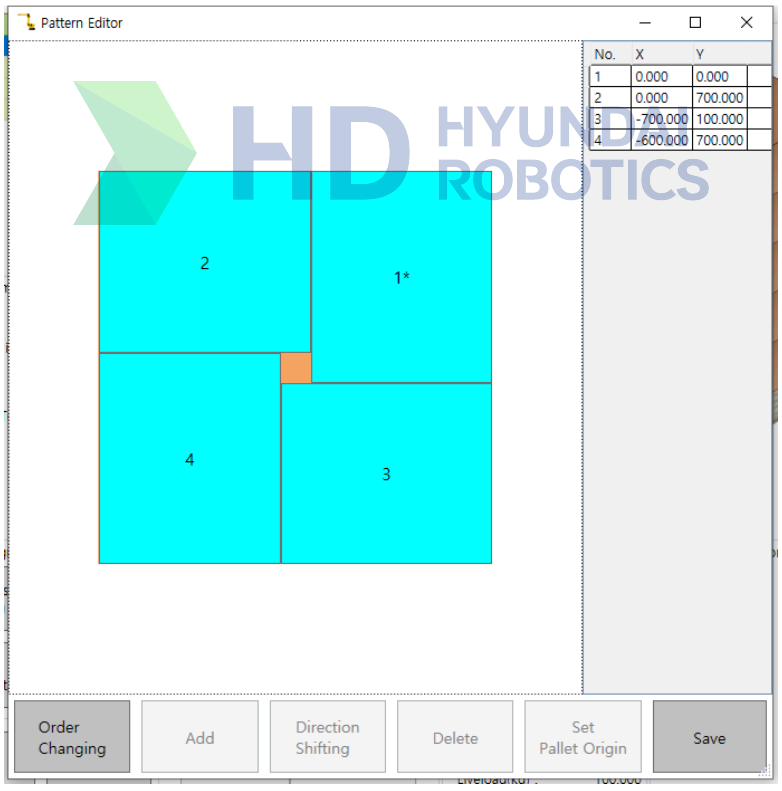


Figure 2.8 Pattern menu changing

2.4.1. Package position changing

The package position can be altered by applying the three following methods

(1) Select a package with a mouse and change its position by the drag and drop function.

(2) Select a package and change its position by using the arrow keys on the keyboard.

(3) Input the X and Y values on the grid window directly.

If interference between workpieces occurs while changing their positions, the objects that are interfering with each other will be displayed in red as shown in (Figure 2.9).

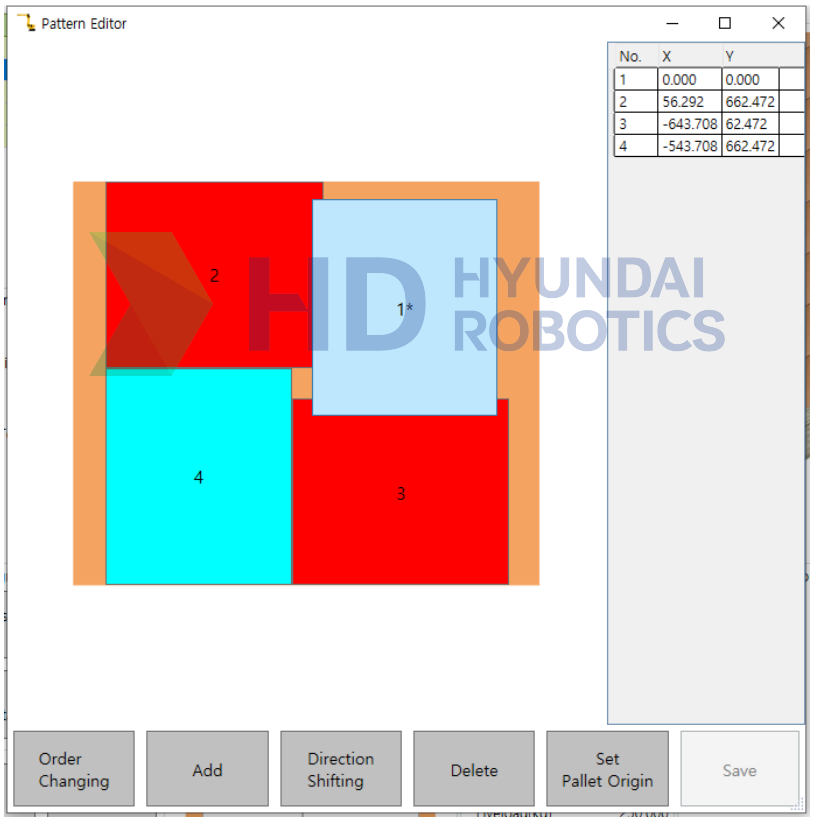


Figure 2.9 Interference occurring mutually while changing the pattern

In addition, a box could be arranged outside the pallet, and if its center is out of the pallet, the box will be displayed in red (Figure 2.10).

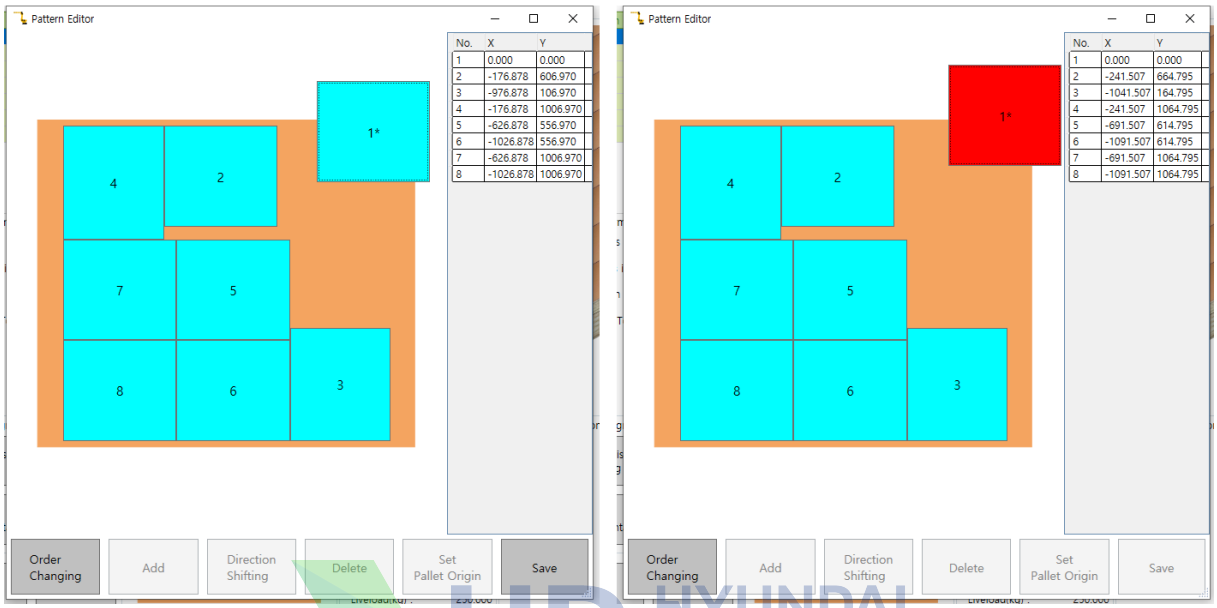
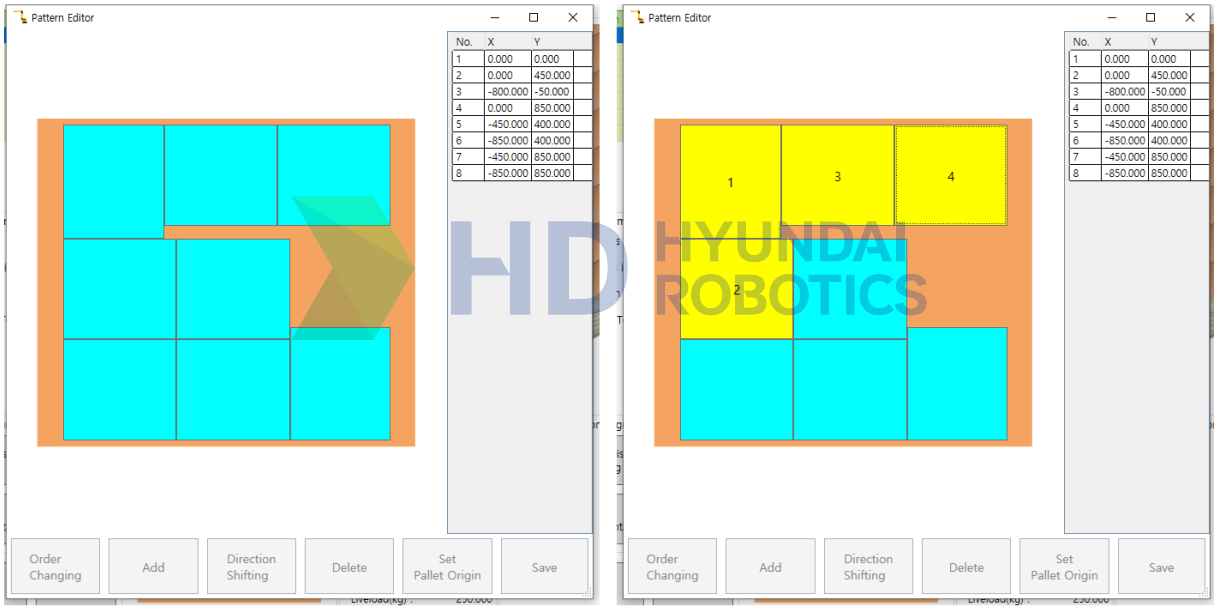


Figure 2.10 The center of gravity of a package moving outside the pallet

2.4.2. Order changing

The order changing function is used to alter the package loading order. When the order changing button is clicked, the loading order for all packages will be initialized as shown in (1) of Figure 2.11 Loading order changing function. When the user touches the boxes using a mouse according to the desired loading order, each box will be colored in yellow and numbered as shown in (2) of Figure 2.11.
While clicking on the packages sequentially to make a change, click on the box if you want to apply a change on a box marked with a certain number. The information on the box and other boxes that come later in order will be initialized, allowing you to reinput their information. For example, while the individual orders for the boxes from nos. 1 to 5 have been inputted already, the individual orders for the boxes from nos. 3 to 5 will be erased if you select Box No. 3, enabling you to input the order information again, beginning from no. 3.



(1) Initialize the loading order

(2) Specify loading order

Figure 2.11 Loading order changing function

2.4.3. Adding

Select a box of desired shape to add and press the Add button. A box with the same shape as selected will be added (Figure 2.12)

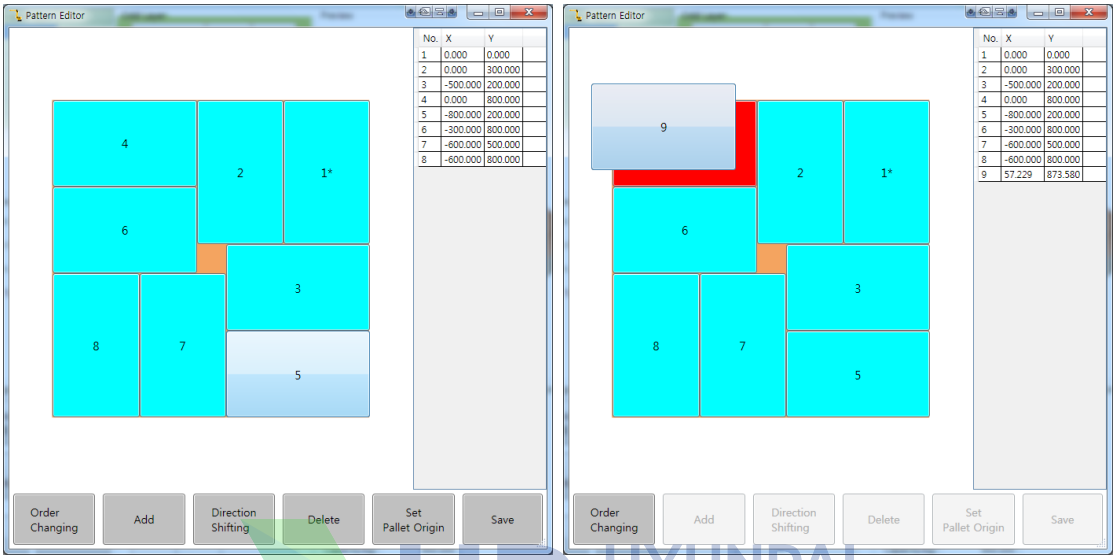


Figure 2.12 Package adding function

2.4.4. Direction shifting

The direction shifting function is used to shift the selected box in the horizontal or vertical direction. (Figure 2.13) below shows the result of selecting, shifting, and rearranging the box #3.

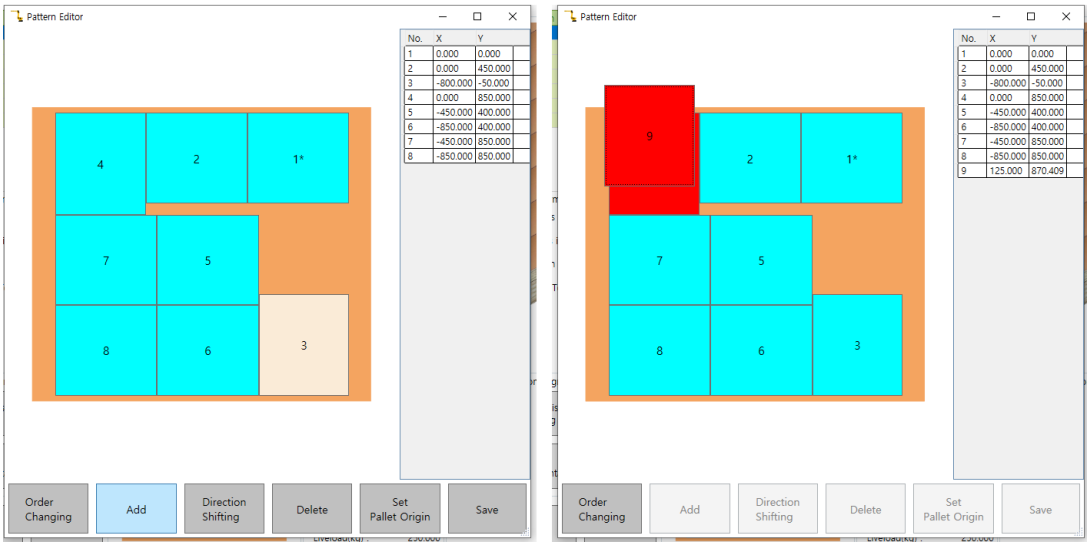


Figure 2.13 Package direction shifting function

2.4.5. Deleting

The deleting function is used to delete a specific package from a pattern. Select a package to delete and press the Delete button so the selected package will disappear. At this time, the packages in the later order than the deleted package will be rearranged in their loading order. (Figure 2.14)

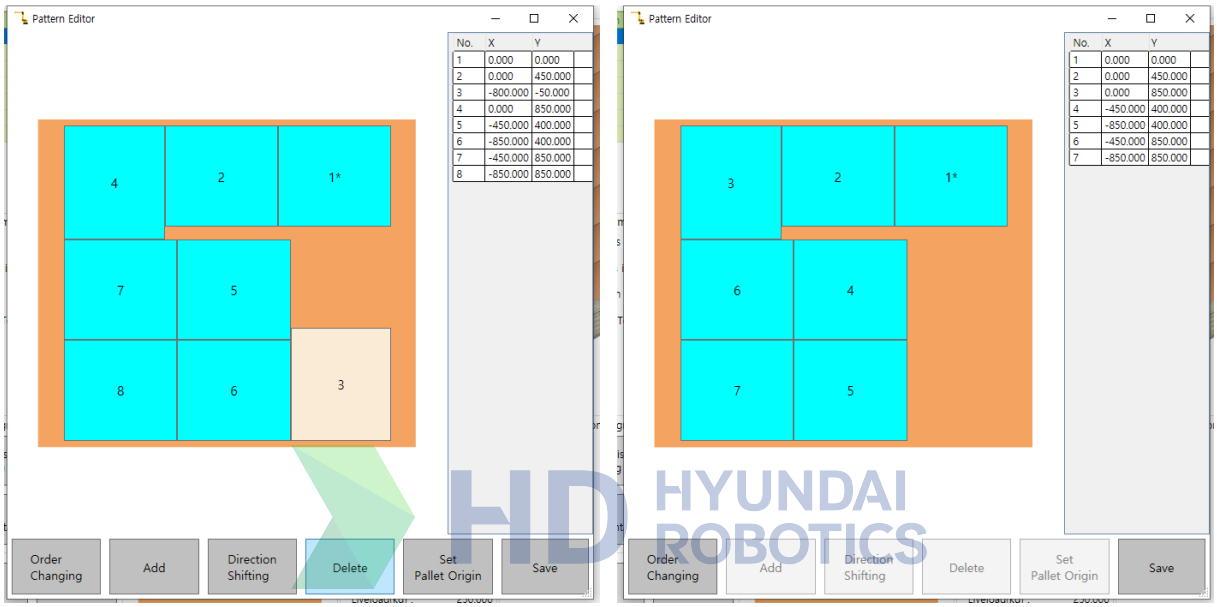


Figure 2.14 Package deleting

2.4.6. Pallet coordinate origin setting

In the standard job program to be created by using HRpal, the loading pattern will be created based on the pallet coordinate. The pallet coordinate means a coordinate defined by the user on the pallet. As shown in Figure 2.15 below, the coordinate, which of the same shape as a right hand is used.
Because the position, direction, and shape of pallets may vary in each case, the users are required to carry out teaching for the origin, X-axis direction, and Y-axis direction on the pallet directly. The pallet coordinate origin is a reference point to carry out the teaching of the pallet coordinate. Basically, the default is the odd layer workpiece #1. However, for the sake of user's convenience, it is possible to set other workpiece as the origin by using the pallet coordinate origin setting function.
Select the workpiece for which the pallet coordinate origin needs to be changed, and click the pallet coordinate origin setting button. Then, the * mark will be displayed on the workpiece on the relevant position, and the origin will be changed (Figure 2.15).

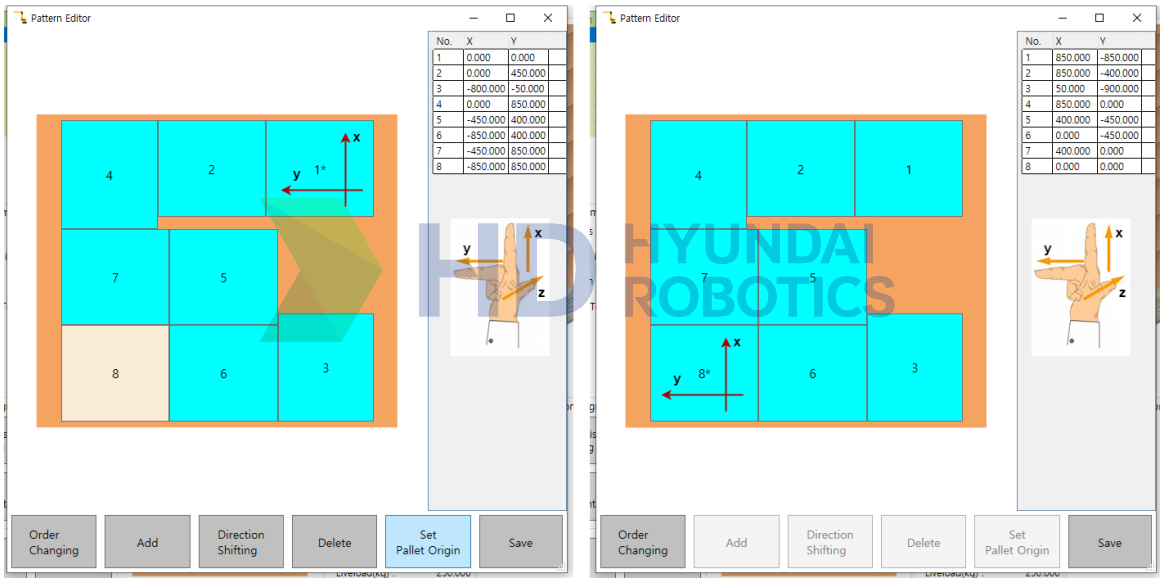


Figure 2.15 Pallet coordinate origin setting function

2.4.7. Hand open direction setting

The palletizing gripper is to be placed downward from above without the need to differentiate the direction as shown below. However, in some cases, the need to open the hand in a specific direction shall be considered (Figure 3.15). The hand open direction function is used to designate direction for a hand in which the hand open direction shall be considered.

Figure 2.16 Palletizing grippers

When the mouse right button is clicked on each package, the arrow direction key will be enabled, making it possible to automatically designate the direction as up, down, left, or right by moving the mouse (Figure 3.16). If the user does not want to designate the direction, click the mouse left button on the center right after pressing the mouse right button.
In this case, the arrow means that the hand will be opened in the arrow direction. Also, the designated data are to be used to calculate and save the rotational direction of the hand when a job program is created later.

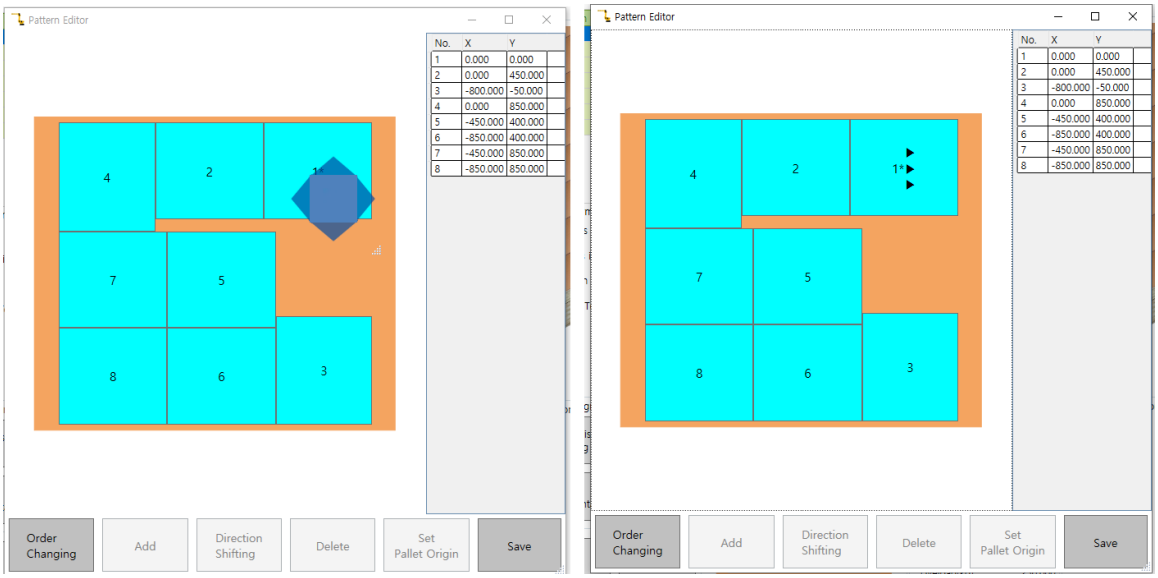


Figure 2.17 Hand open direction designation function

2.4.8. Saving

The Save button is used to reflect any alteration, such as position changing or deleting, after such alteration is made. When the Save button is pressed, the relevant window will be end automatically, making it possible to check if a change is made in the content of the selected layer.

2.5. Pattern managing

The pattern managing function is a menu to manage the patterns previously created by the user (Figure 3.17). This menu can be used to create a job program or to generate a POSE file. Pattern resister is a number that corresponds to a relevant pattern that is to be used in the job program and can be designated by inputting it directly. When it comes to the patterns for which the hand direction shall be set, it is possible to designate and use relevant options.

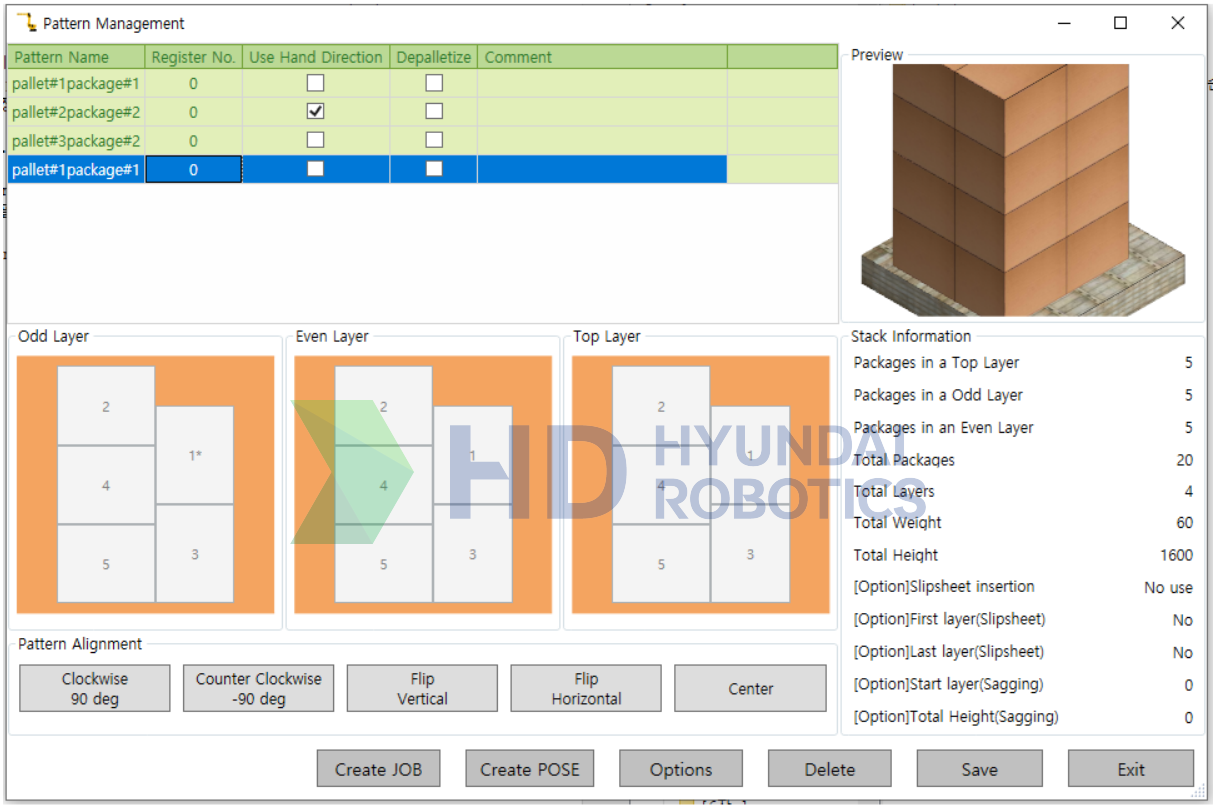


Figure 2.18 Pattern managing function

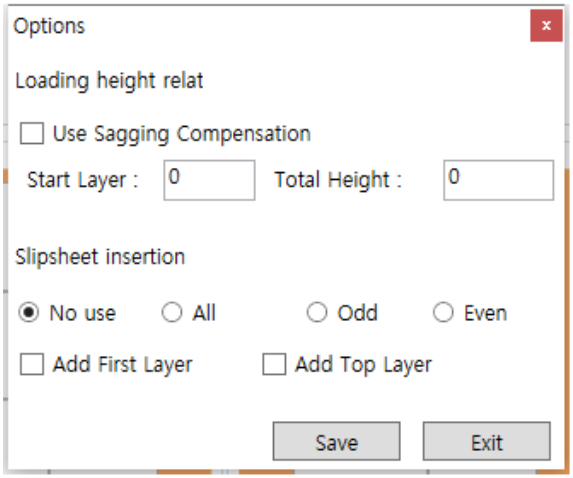
2.5.1. Job creating

The job creating function is used to create and produce a job program[1] and HRSpace4 simulation environment in order to execute palletizing for the pattern for which the register number is designated. The features of the relevant function are written below.

- Create and produce a standard program to make it possible to use the patterns for which pattern register numbers are assigned in the robot.
- Automatically create a package model for the relevant pattern for the sake of convenience in simulation in HRSpace3.
- The pattern register numbers that can be assigned range from 1 to 16, while they cannot be assigned in duplication.
- While the hand direction setting option is enabled, if the package hand open direction is not designated, a job program cannot be created.
- When the Job Create button is pressed, a menu will be displayed to select a folder to save the job program. Select the relevant folder and click the Check button so the job program will be saved in the folder.
- A menu will be displayed to select the controller version with which you want to perform the simulation when the "Job Create" button is pressed.
- Once the controller version is selected, a menu will be displayed to select a folder to save the job program, which will then be saved when you select a folder and press the "OK" button.
- It would be impossible to execute if any pattern register is not set.
- If the user does not want to create a job program for a pattern, the register shall be changed to 0

2.5.2. Option changing

The option changing function is used to make a change related to the sagging compensation options and the slip sheet insertion (Figure 2.19).



[1] Refer to "3. Palletizing Standard Program:" for more details about the standard program.

Figure 2.19 Option changing menu

2.5.3. Pattern deleting, saving, and ending

- The Pattern Delete button is used to delete the currently designated pattern.
  (Caution) Even when the pattern is deleted from the current screen, it will not be reflected in the file without pressing the Pattern Save button.

- The Pattern Save button is used to save the partially changed or altered content in a file.

2.6. User pattern defining

The user pattern defining function is for the user to create a pattern by arranging packages arbitrarily (Figure 2.20). Different from the auto pattern creating function, this function makes it possible to create a pattern by arranging different packages. The following show the order of using the function.

(1) Select a pallet that is to be placed on the floor.
(2) Select a package to load.
(3) Configure a pattern by using the adding, direction shifting, and deleting functions.
(4) Place the created pattern on the center by using the pattern editing function, including the clockwise or anticlockwise direction.
(5) Define the patterns for the odd layer and the top layer by using the same methods as shown above.
(6) You can use the same pattern as the odd layer by using the (*) odd layer copy function.
(7) Designate the loading layer count.
(8) Designate based on the layer sagging compensation or slip sheet insertion
(9) Press the Save button[2] and designate the pattern name to save so the pattern registration will be completed.

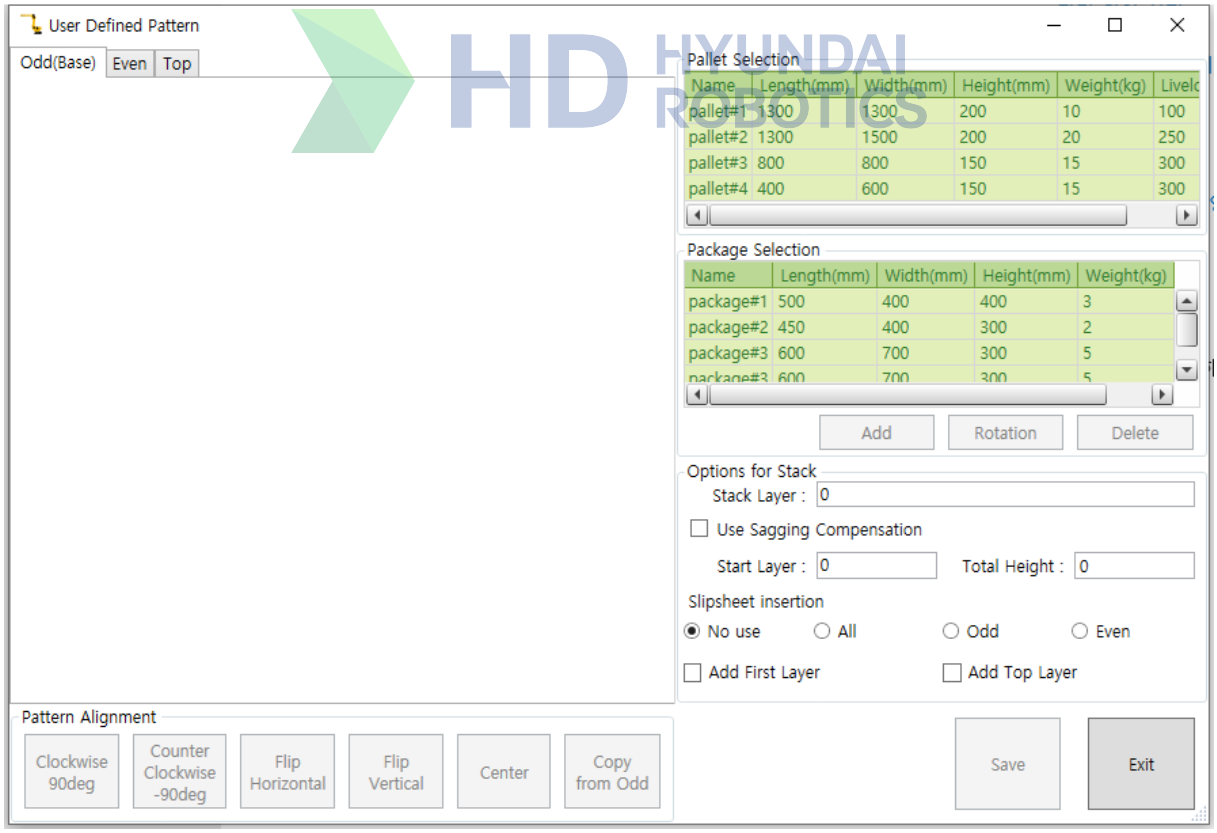


Figure 2.20 User pattern defining menu

[2] The saving function will be enabled only when the data for the odd layer, even layer, and top layer are filled out completely.

HD
HYUNDAI
ROBOTICS

HYUNDAI ROBOTICS

3

Palletizing Standard Program

The palletizing program is a program group configured in advance to contain frames that are necessary for palletizing. When the user carries out palletizing by using HRpal, if a relevant program is used, palletizing can be carried out easily and quickly only by recording the work-related position and changing variables.

3.1. Variable configuration

Table 3-1 Configuration and usage of the processInfo variable in the standard program

processInfo Global Variable		
Object Array	An array consisting of 16 objects; a variable for managing process-related information in an integrated manner A process not defined will be initialized to 0. For example, the configuration will be as follows if a job is created with process nos. 3 and 6. [0, 0, Object {regNo:3, ...}, 0, 0, Object {regNo:6, ...}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	
Type	**Key**	**Description**
Integer	regNo	Variable to store the process no.
Integer	totalPackages	Total count of the workpieces in the relevant process (slip sheet not included)
Integer	oddPackages	Count of the workpieces on the odd layers
Integer	evenPackages	Count of the workpieces on the even layers
Integer	topPackages	Count of the workpieces at the top layer
Integer	totalSlipsheet	Total count of the slip sheets in the relevant process
Boolean	firstLayerSlipSheet	Whether to insert the slip sheet for the bottom layer
Boolean	topLayerSlipSheet	Whether to insert the slip sheet for the top layer
Float	packageLength	Size of the workpiece (L)
Float	packageWidth	Size of the workpiece (W)
Float	packageHeight	Size of the workpiece (H)
Float	pickupHeight	Pickup height (The initial value is packageHeight * 1.5.)

Table 3-2 Configuration and usage of the simulator variable in the standard program

simulator Global Variable		
Object	Object to save information related to palletizing simulation	
Type	**Key**	**Description**
Integer	currRegNo	No. of the current process
Integer	prevRegNo	No. of the immediately previous process
Integer	currActionNo	Work no. of a process (0 - [total count of workpieces + total count of slip sheets] -1)
Integer	currPackageType	Type no. of the current package (will be determined as a slip sheet if it is 1–5 and 7)
Integer	prevPackageType	Type no. of the previous package (will be determined as a slip sheet if it is 1–5 and 7)
Integer	cntPackage	Workpiece counter
Integer	cntSlipsheet	Slip sheet counter
Integer	cntLayer	Layer counter
Pos Object	originUCS	Origin coordinates of the user coordinate system (target position of the first workpiece)
integer	error	Variable to save information when an error occurs during simulation
Pos Object	homePos	Placement of the home position
Pos Object	nearPickupPos	Just above the pickup position
Pos Object	pickupPos	Pickup position
Pos Object	nearTargetPos	Just above the loading pattern
Pos Object	targetPos	Loading pattern position

Table 3-3 Configuration and usage of the pos variable in the standard program

pos Global Variable		
Object Array	An array consisting of 16 objects; a variable for managing process-related position information The information of an undefined process will be initialized to 0. For example, the configuration will be as follows if a job is created with process nos. 3 and 6. [0, 0, Object {homePos:Pos Object(), ...}, 0, 0, Object {homePos:Pos Object(), ...}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	
Type	**Key**	**Description**
Pos Object	homePos	Basic posture (home position) for each process
Pos Object	pickupPos1	Position to pick up package #1 at each process (Five different types of packages can be handled at each process.)
Pos Object	pickupPos2	Position to pick up package #2 at each process
Pos Object	pickupPos3	Position to pick up package #3 at each process
Pos Object	pickupPos4	Position to pick up package #4 at each process
Pos Object	pickupPos5	Position to pick up package #5 at each process
Pos Object	SlipSheetPos	Position to pick up the slip sheet at each process

Table 3-4 Configuration and usage of the shift variable in the standard program

Shift Global Variable		
Type	**Key**	**Description**
Shift Array	packageShift1	Package shift variable for process #1 Ex) packageShift1[1], packageShift1[55]
Shift Array	packageShift2	Package shift variable for process #2
Shift Array	packageShift3	Package shift variable for process #3
...		
Shift Array	packageShift16	Package shift variable for process #16
Shift Array	slipSheetShift1	Slip sheet shift variable for the process #1 Ex) slipSheetShift1[1], slipSheetShift1[55]
Shift Array	slipSheetShift2	Slip shift variable for process #2
Shift Array	slipSheetShift3	Slip shift variable for process #3
...		
Shift Array	slipSheetShift16	Slip shift variable for process #16

3.2. Overall flow chart

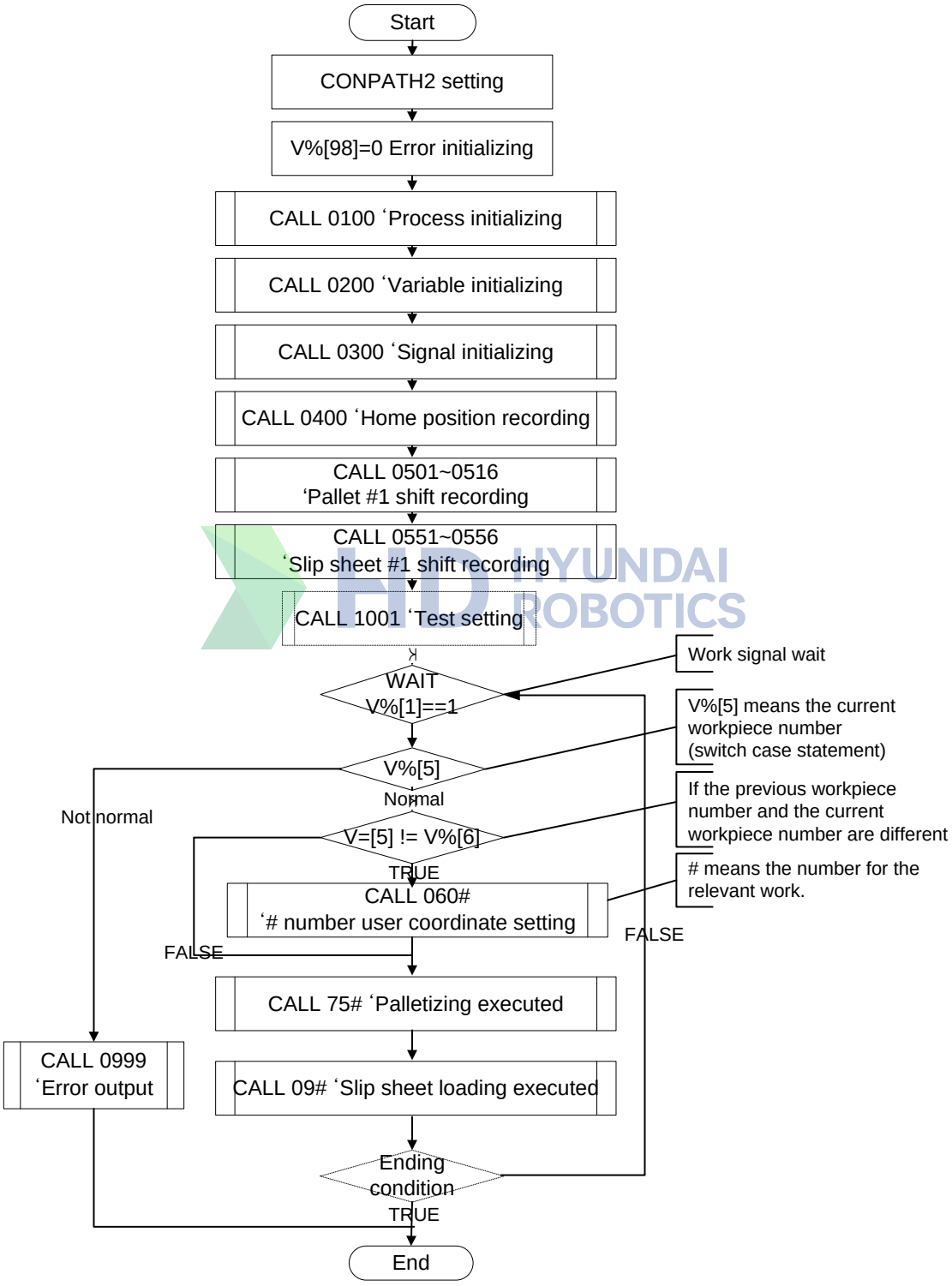


Figure 3.1 Standard program overall flow chart

3.3. Program configuration

Table 3-5 Job numbers and roles in the standard program

Job Number	Role	Description
1	Main	Overall flow managing
50	Vacuum On	Records signals for the griper to hold objects
51	Vacuum Off	Records signals for the griper to release objects
100	Process parameter setting	
200	Initialization of variables	
300	Initialization of signals	
400	Home position and workpiece position pose program	Needed to record the position by using the pose constant in the defined pose variable.
451~466	Tool control program	Controls the positions of individual tools
501~516	Each pallet loading pattern shift variable program	To be created automatically by HRpal and HRpalware
551~566	Slip sheet shift variable program	To be created automatically by HRpal and HRpalware
601~616	Pallet coordinate creating program	Needed to create the user coordinate on the pallet and to record the origin of the work as well as the points on the X axis and X-Y plane
701~716	Pallet parameter setting	
751~766	Palletizing motion program	Executes palletizing by moving the robot
801~816	Slip sheet parameter setting	
851~866	Slip sheet motion program	Places a slip sheet by moving the robot
999	Error handling program	Program for indicating errors when they occur
1001	(Simulation) test setup program	To allow users to use simulated values to check whether the set values work normally.
1002	(Simulation) test ending condition check	To record the ending condition to check the cycle times

3.3.1. Main program

The main program is divided into the following 4 sections.

(1) To initialize processes and variables
(2) To record tool control and shift variables
(3) To judge and execute palletizing conditions
(4) To process errors

When it is necessary to run processes by using the standard program, the user must run the main program at least once. This is because the main program processes signals and carries out initialization. Up to 16th lines in [Program 1 Main program configuration] are related to the basic initialization. If the initialization section is run at least once and the relevant data are saved, the job programs such as pallet movement and slip sheet movement can be run normally without additional processing.

Program0001_MainJob.job

```
1  Hyundai Robot Job File; { version: 1.6, mech_type: "0()", total_axis: 0, aux_axis: 0 }
2        # Main Job Program
3        #-----------------------
4        contpath 2
5        #---------------------------------------
6        call 100 # process Initialize
7        call 200 # variable Initialize
8        call 300 # signal Initialize
9        call 400 # home position Initialize
10        #---------------------------------------
11        call 451   # Record Tool Control for 1 Process
12        call 501   # Record Pallet Shift Information for 1 Process
13        call 551   # Record InsertSlipsheet Shift Information for 1 Process
14        call 452   # Record Tool Control for 2 Process
15        call 502   # Record Pallet Shift Information for 2 Process
16        #---------------------------------------
17        call 1001 # Simulator Initialize
18        var startTimeStamp = timer()
19        #---------------------------------------
20  *procStart
21        switch simulator.currRegNo
22
23        #---------------------------------------
24        case 1 # Work # 1
25              if simulator.currRegNo != simulator.prevRegNo
26                    call 601 # set User CRD
27              endif
28              if simulator.currPackageType == 7
29                    call 851 # move Slipsheet (#1 process)
30              else
31                    call 751 # move package (#1 process)
```

```
32              endif
33              simulator.prevRegNo = simulator.currRegNo
34              break
35
36          #--------------------------------------
37          case 2 # Work # 2
38              if simulator.currRegNo != simulator.prevRegNo
39                  call 602 # set User CRD
40              endif
41              call 752 # move package (#2 process)
42              simulator.prevRegNo = simulator.currRegNo
43              break
44          default
45              simulator.error = 1
46              call 999 # Error Handling
47              goto *procEnd
48              break
49          end_switch
50
51          #--------------------------------------
52          call 1002 # load next work info
53          if simulator.currRegNo > 16 then *procEnd
54          goto *procStart # Next Process Start
55
56          #--------------------------------------
57      *procEnd
58          print "Total Cycle Time =", timer()-startTimeStamp
59          end
```

3.3.2. Vacuum On, Off

The Vacuum on and off programs (Job program #50 and #51) are used in the pallet moving program. When it comes to job programs that are created basically, they are defined in the job programs by using DO10 as a signal. When they are applied for real processes, the job programs shall only be changed by using proper signals for relevant processes.

3.3.3. Process parameter setting

The Process parameter setting program (Job program #100) is a job program used to decide whether to use the relevant process. In case process-related information has been written in the values of processInfo[0] - processInfo[15] and a specific process must be set to “Disable,” the process registration information must be handled as a comment.
Thus, assigned parameter values can be used to check whether the relevant processes are enabled by monitoring them later. Most values will be automatically created in HRpal. However, the pickupHeight value in lines 19 and 35 of the sample codes shown below can be used after the user’s direct modification. The basic value is 1.5 times the height of the workpiece (Figure 3-2).

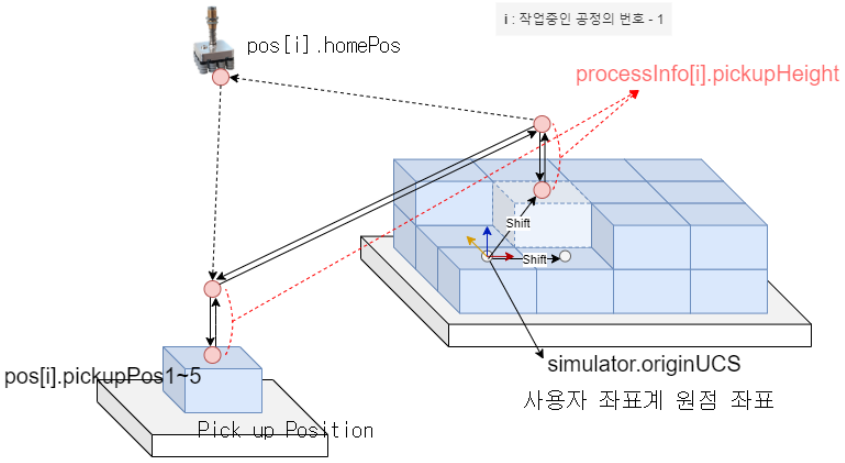


Figure 3.2 Coordinate information to be used for the palletizing process

Program 0100_ProcessSetting.job

```
1   Hyundai Robot Job File; { version: 1.6, mech_type: "0()", total_axis: 0, aux_axis: 0 }
2       # Process Parameter Setting
3       global processInfo = Array(16)
4
5       #-------------------------------------
6       # Process Info for #1 Process
7       processInfo[0] = {regNo:0, totalPackages:0, oddPackages:0, evenPackages:0,
                          topPackages:0, totalSlipsheet:0, firstLayerSlipSheet:0,
                          topLayerSlipSheet:0, packageLength:0, packageWidth:0,
                          packageHeight:0, pickupHeight:0}
8       processInfo[0].regNo = 1 # RegisterNo
9       processInfo[0].totalPackages = 24 # TotalPackageCount
10      processInfo[0].oddPackages = 12 # OddPackageCount
11      processInfo[0].evenPackages = 12 # EvenPackageCount
12      processInfo[0].topPackages = 12 # TopPackageCount
13      processInfo[0].totalSlipsheet = 1 # SlipsheetStackLayer
14      processInfo[0].firstLayerSlipSheet = false # SlipsheetStackFirst
15      processInfo[0].topLayerSlipSheet = false # SlipsheetStackTop
16      processInfo[0].packageLength = 180 # Package Length
17      processInfo[0].packageWidth = 450 # Package Width
18      processInfo[0].packageHeight = 160 # Package Height
19      processInfo[0].pickupHeight = 240 # PickupHeight
20
21      #-------------------------------------
22      # Process Info for #2 Process
23      processInfo[1] = {regNo:0, totalPackages:0, oddPackages:0, evenPackages:0,
                          topPackages:0, totalSlipsheet:0, firstLayerSlipSheet:0,
                          topLayerSlipSheet:0, packageLength:0, packageWidth:0,
                          packageHeight:0, pickupHeight:0}
24      processInfo[1].regNo = 2 # RegisterNo
```

```
25        processInfo[1].totalPackages = 36 # TotalPackageCount
26        processInfo[1].oddPackages = 18 # OddPackageCount
27        processInfo[1].evenPackages = 18 # EvenPackageCount
28        processInfo[1].topPackages = 18 # TopPackageCount
29        processInfo[1].totalSlipsheet = 0 # SlipsheetStackLayer
30        processInfo[1].firstLayerSlipSheet = false # SlipsheetStackFirst
31        processInfo[1].topLayerSlipSheet = false # SlipsheetStackTop
32        processInfo[1].packageLength = 200.52 # Package Length
33        processInfo[1].packageWidth = 300.11 # Package Width
34        processInfo[1].packageHeight = 200.2 # Package Height
35        processInfo[1].pickupHeight = 300.3 # PickupHeight
36        end
```

3.3.4. Initialization of variables and signals

The Variable initialization program (Job program #200) is a job program used to designate the initial values of various variables that are used in the standard program.

Program 0200_VariableInit.job

```
1   Hyundai Robot Job File; { version: 1.6, mech_type: "0()", total_axis: 0, aux_axis: 0 }
2         # Initialize Variable
3         global simulator = {currRegNo:0, prevRegNo:0, currActionNo:0, currPackageType:0,
                              prevPackageType:0, cntPackage:0, cntSlipsheet:0, cntLayer:0,
                              originUCS:0, error:0, homePos:0, nearPickupPos:0,
                              pickupPos:0, nearTargetPos:0, targetPos:0}
4         print "Modify various variable values, and then delete PRINT and STOP."
5         stop
6         simulator.currRegNo = 0
7         simulator.prevRegNo = 0
8         simulator.currPackageType = 1
9         simulator.prevPackageType = 1
10        simulator.cntPackage = 0 # Package Counter Reset
11        simulator.cntSlipsheet = 0 # SlipSheet Counter Reset
12        simulator.cntLayer = 0 # Layer Counter Reset
13        end
```

The signal initialization program (JOB program #300) is used to record the signals that must be initialized before a process is executed. Initially, there is no content defined, and it is required to record signals in advance such that it fits the process.

The Signal initializing program (Job program #300) is used to record the signals that shall be initialized before executing relevant programs. At the initial stage, there is nothing defined. Signals shall be recorded

according to relevant processes.

3.3.5. Home position and workpiece position pose program

The Home position and workpiece position program (Job program #400) is a program for recording the home position of a specific process and the position to move to for the pickup operation. The homePos variables are used for designating the home positions while the pose variables of pickupPos1~pickupPos5 are used for designating pickup positions for individual processes. In each process, up to 5 different workpieces can be handled.

Program 0400_HomePos.job

```
1   Hyundai Robot Job File; { version: 1.6, mech_type: "0()", total_axis: 0, aux_axis: 0 }
2       # Home, Package, Slipsheet Position
3       global pos = Array(16)
4       print "Modify various variable values, and then delete PRINT and STOP."
5       stop
6
7       #---------------------------------------
8       # Position Info for #1 Process
9       pos[0] = {homePos:0, pickupPos1:0, pickupPos2:0, pickupPos3:0, pickupPos4:0,
                  pickupPos5:0, SlipSheetPos:0}
10      pos[0].homePos = cpo("robot", "cur") # Home Position #1
11      pos[0].pickupPos1 = cpo("robot", "cur") # Pickup Position #Work 1_1
12      pos[0].pickupPos2 = cpo("robot", "cur") # Pickup Position #Work 1_2
13      pos[0].pickupPos3 = cpo("robot", "cur") # Pickup Position #Work 1_3
14      pos[0].pickupPos4 = cpo("robot", "cur") # Pickup Position #Work 1_4
15      pos[0].pickupPos5 = cpo("robot", "cur") # Pickup Position #Work 1_5
16      pos[0].SlipSheetPos = cpo("robot", "cur") # Slipsheet Position #1
17
18      #---------------------------------------
19      # Position Info for #2 Process
20      pos[1] = {homePos:0, pickupPos1:0, pickupPos2:0, pickupPos3:0, pickupPos4:0,
                  pickupPos5:0, SlipSheetPos:0}
21      pos[1].homePos = cpo("robot", "cur") # Home Position #2
22      pos[1].pickupPos1 = cpo("robot", "cur") # Pickup Position #Work 2_1
23      pos[1].pickupPos2 = cpo("robot", "cur") # Pickup Position #Work 2_2
24      pos[1].pickupPos3 = cpo("robot", "cur") # Pickup Position #Work 2_3
25      pos[1].pickupPos4 = cpo("robot", "cur") # Pickup Position #Work 2_4
26      pos[1].pickupPos5 = cpo("robot", "cur") # Pickup Position #Work 2_5
27      end
```

3.3.6. Tool control program

The Tool control programs (Job programs #451 ~ #466) are used for carrying out defining regarding the control of tools that are used to control jigs and positions. Whether to use tool control programs depend on individual processes, and the positions for using them may vary even when they are used. That is why their calling positions are not designated specifically. Considering this, the users need to prepare control programs to meet the needs of individual processes and designate the calling positions directly.

3.3.7. Loading pattern shift variable program for individual pallets

The Loading pattern shift variable programs for individual pallets (Job programs #501 ~ #516) and the Slip sheet shift variable programs (Job programs #551 ~ #566) are the programs where the loading patterns and the slip sheet shift values are saved. The patterns created by using HRpal are saved automatically and the users can change the shift values directly.
What is to be indicated as a note in each shift variable is Layer_Order. For example, 1_3 means the 3rd workpiece on the 1st layer.

Program 0501_PatternShift_1.job

```
1   Hyundai Robot Job File; { version: 1.6, mech_type: "0()", total_axis: 0, aux_axis: 0 }
2
3   global packageShift1 = Array(24) # init packageShift Variable
4   packageShift1[0] = Shift(0.000, 0.000, 0.000, 0.000, 0.000, 0.000, "u1") # 1_1
5   packageShift1[1] = Shift(0.000, 180.000, 0.000, 0.000, 0.000, 0.000, "u1") # 1_2
6   packageShift1[2] = Shift(0.000, 360.000, 0.000, 0.000, 0.000, 0.000, "u1") # 1_3
7   packageShift1[3] = Shift(-315.000, 135.000, 0.000, 0.000, 0.000, 90.000, "u1") # 1_4
8   packageShift1[4] = Shift(135.000, 675.000, 0.000, 0.000, 0.000, 90.000, "u1") # 1_5
9   packageShift1[5] = Shift(-495.000, 135.000, 0.000, 0.000, 0.000, 90.000, "u1") # 1_6
10  packageShift1[6] = Shift(-45.000, 675.000, 0.000, 0.000, 0.000, 90.000, "u1") # 1_7
11  packageShift1[7] = Shift(-675.000, 135.000, 0.000, 0.000, 0.000, 90.000, "u1") # 1_8
12  packageShift1[8] = Shift(-225.000, 675.000, 0.000, 0.000, 0.000, 90.000, "u1") # 1_9
13  packageShift1[9] = Shift(-540.000, 450.000, 0.000, 0.000, 0.000, 0.000, "u1") # 1_10
14  packageShift1[10] = Shift(-540.000, 630.000, 0.000, 0.000, 0.000, 0.000, "u1") # 1_11
15  packageShift1[11] = Shift(-540.000, 810.000, 0.000, 0.000, 0.000, 0.000, "u1") # 1_12
16  packageShift1[12] = Shift(0.000, 0.000, 170.000, 0.000, 0.000, 0.000, "u1") # 2_1
17  packageShift1[13] = Shift(0.000, 180.000, 170.000, 0.000, 0.000, 0.000, "u1") # 2_2
18  packageShift1[14] = Shift(0.000, 360.000, 170.000, 0.000, 0.000, 0.000, "u1") # 2_3
19  packageShift1[15] = Shift(-315.000, 135.000, 170.000, 0.000, 0.000, 90.000, "u1") # 2_4
20  packageShift1[16] = Shift(135.000, 675.000, 170.000, 0.000, 0.000, 90.000, "u1") # 2_5
21  packageShift1[17] = Shift(-495.000, 135.000, 170.000, 0.000, 0.000, 90.000, "u1") # 2_6
22  packageShift1[18] = Shift(-45.000, 675.000, 170.000, 0.000, 0.000, 90.000, "u1") # 2_7
23  packageShift1[19] = Shift(-675.000, 135.000, 170.000, 0.000, 0.000, 90.000, "u1") # 2_8
24  packageShift1[20] = Shift(-225.000, 675.000, 170.000, 0.000, 0.000, 90.000, "u1") # 2_9
25  packageShift1[21] = Shift(-540.000, 450.000, 170.000, 0.000, 0.000, 0.000, "u1") # 2_10
26  packageShift1[22] = Shift(-540.000, 630.000, 170.000, 0.000, 0.000, 0.000, "u1") # 2_11
27  packageShift1[23] = Shift(-540.000, 810.000, 170.000, 0.000, 0.000, 0.000, "u1") # 2_12
28  end
```

3.3.8. Pallet coordinate creating program

The shift data of the patterns created by HRpal are all recorded as relative shift values against the #1 package position. Accordingly, it is needed to create the user coordinate based on the #1 package loading position (or teaching position and marked with *) on the pallet and the relevant information is defined in the Pallet coordinate creating program (Job programs #601 ~ #616).
The user coordinate can be created by recording 3 coordinate values. The coordinate values to be recorded are as shown below (Program 5, Figure 3.3).

- P1 – Record the first package loading starting position, which is the coordinate origin at the same time.
- P2 – Set a point on the x axis randomly based on the P1 position (x axis in the + direction).
- P3 – Set a point on the x-y plane randomly based on the P1 position (x-y plane in the + direction).

The user coordinate system is created based on the right-hand coordinate system, which may cause the recording positions of points P1, P2, and P3 to become the coordinates outside the pallet, depending on the loading start position (for example, top right or bottom left). (Refer to Figure 3.3)

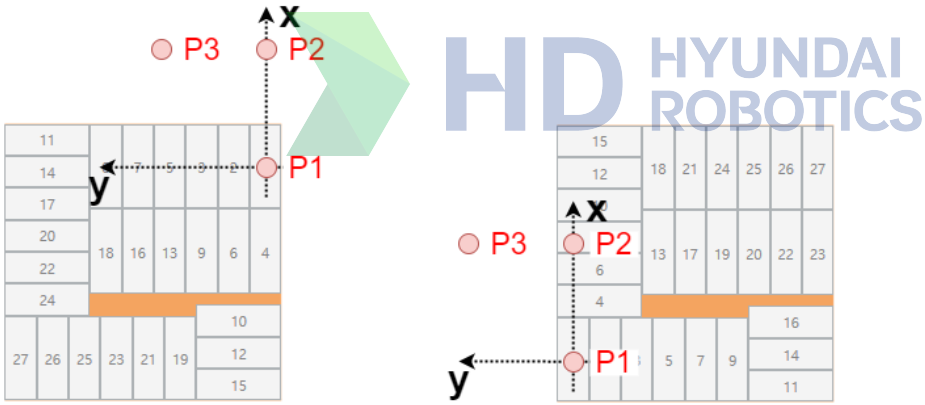


Figure 3.3 Example of the user coordinate recording

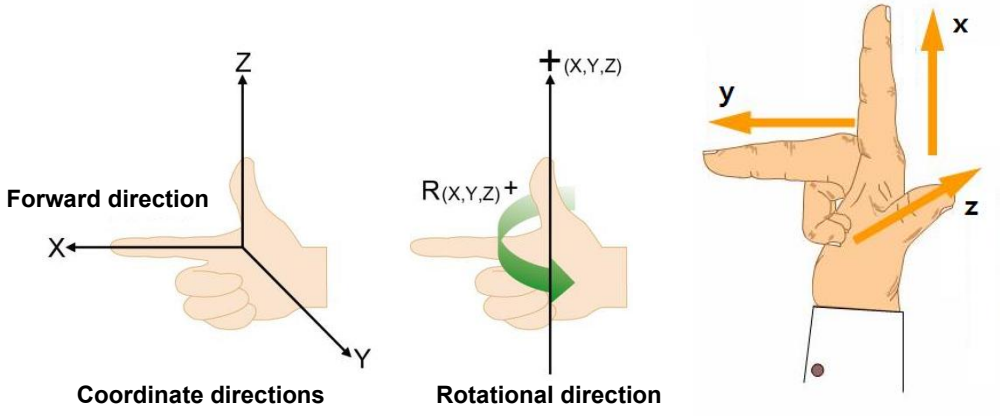


Figure 3.4 Right hand coordinate configuration

Program 0601_UserCrd_1.job

```
1   Hyundai Robot Job File; { version: 1.6, mech_type: "0()", total_axis: 0, aux_axis: 0 }
2         # HHI User Coordinate Define Program
3         print "Record the user coordinate system using the pose constant value."
4         stop
5         var p1 = cpo("robot", "cur") # Origin Position
6         var p2 = cpo("robot", "cur") # Any position on the +X-axis
7         var p3 = cpo("robot", "cur") # Any position on the X-Y plane
8         var result = mkucs(1, p1, p2, p3) # Register user coordinate system
9         selucrd 1
10        simulator.originUCS = p1
11        end
```

3.3.9. Pallet and slip sheet parameter setting

The Pallet and slip sheet parameters setting programs (Job program #701 ~ #716) are used for assigning shift variables to progress palletizing or for initializing variables necessary for palletizing. Basically, there is no need to change the variables of relevant job programs. However, if the step is to be added or changed, the relevant job program can be prepare by changing it.

Program 0701_PalParamSet_1.job

```
1   Hyundai Robot Job File; { version: 1.6, mech_type: "0()", total_axis: 0, aux_axis: 0 }
2         # Palletize Parameter Setting #1
3         simulator.error = 0 # Error Handling Variable Init
4         if simulator.cntPackage == processInfo[0].totalPackages
5               simulator.cntPackage = 0
6               simulator.cntSlipsheet = 0
7               simulator.cntLayer = 0
8         endif
9
10        var shiftHeight = Shift(0, 0, processInfo[0].pickupHeight, 0, 0, 0, "robot")
 # shift position for lift or place down
11        var shiftPackage = packageShift1[simulator.cntPackage]
12
13        var pickupPos
14        switch simulator.currPackageType
15        case 1
16              pickupPos = pos[0].pickupPos1
17              break
18        case 2
19              pickupPos = pos[0].pickupPos2
20              break
21        case 3
22              pickupPos = pos[0].pickupPos3
23              break
24        case 4
25              pickupPos = pos[0].pickupPos4
26              break
27        case 5
28              pickupPos = pos[0].pickupPos5
29              break
30        default
31              pickupPos = pos[0].pickupPos1
32              break
33        end_switch
34
35        simulator.homePos = pos[0].homePos # Home position during palletizing
36        simulator.nearPickupPos = pickupPos+shiftHeight
 # Just above the position to pick up the package or slipsheet.
```

```
37          simulator.pickupPos = pickupPos # The position to pick up the package or slipsheet.
38          simulator.nearTargetPos = simulator.originUCS+shiftPackage+shiftHeight
 # just above the position to put down the package or slipsheet.
39          simulator.targetPos = simulator.originUCS+shiftPackage
 # The position to put down the package or slipsheet.
40          end
```


HD HYUNDAI ROBOTICS

3.3.10. Palletizing and slip sheet motion program

The palletizing and slip sheet motion program is a combination of the Move statements that are needed to move the robot. The standard program needs 4 position values (package, pick up, near place down, and place down) to operate continuously. The 4 values are to be defined using the pickup height values designated in 4.3.5 Home position and workpiece position pose program, 4.3.8 Pallet coordinate creating program, and 4.3.4 Initialization of variables and signals. The home position means an initial position to move to when other workpiece enters or when there is a need to carry out noncontinuous operations, such as moving to insert slip sheet. When looking at the lines #3~#5, we can know that this will be executed only when there is a change.
If needed to add another via point somewhere between S4, UP, and S5 Near Place Down, it is necessary to add a new Move statement between S4 and S5.

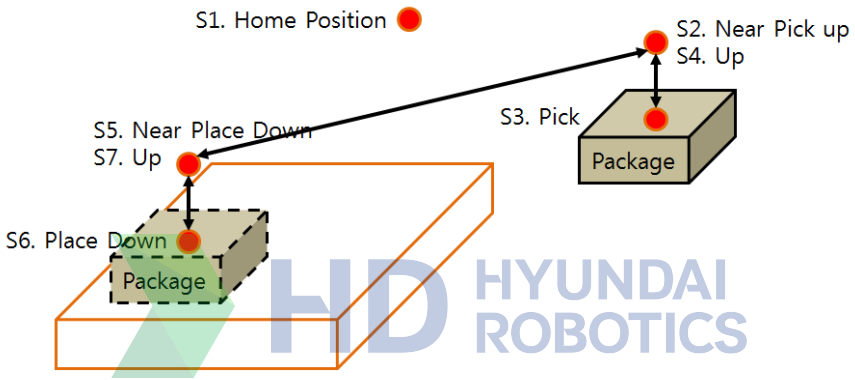


Figure 3.5 Palletizing motion position values

Program 0751_PalJob_1.job

```
1    Hyundai Robot Job File; { version: 1.6, mech_type: "0()", total_axis: 0, aux_axis: 0 }
2          # Palletize Job Program #1
3          call 701 # Motion Param Set
4          if (simulator.currRegNo!=simulator.prevRegNo or
 simulator.currPackageType!=simulator.prevPackageType)
5    S1          move P,tg=simulator.homePos,spd=90%,accu=1,tool=0 # Home Position
6          endif
7    S2    move P,tg=simulator.nearPickupPos,spd=90%,accu=5,tool=0 # Near Pick up
8    S3    move L,tg=simulator.pickupPos,spd=90%,accu=0,tool=0 # Pick
9          call 0050 # Vaccum On
10   S4    move L,tg=simulator.nearPickupPos,spd=90%,accu=5,tool=0 # Up
11   S5    move P,tg=simulator.nearTargetPos,spd=90%,accu=5,tool=0 # Near Place Down
12   S6    move L,tg=simulator.targetPos,spd=90%,accu=0,tool=0 # Place Down
13         call 0051 # Vaccum off
14         simulator.cntPackage = simulator.cntPackage+1 # Counter increment
15         simulator.prevPackageType = simulator.currPackageType
16   S7    move L,tg=simulator.nearTargetPos,spd=90%,accu=5,tool=0 # Up
17         end
```

3.4. Standard program basic usage

Most part of the standard program is configured automatically and can be run when the user changes some sections only. What needs to be changed includes variable setting and position recording. When the relevant sections are changed if needed, the robot will move automatically, making it possible to check the motion of the robot very quickly.

3.4.1. Variable setting

The variable the user needs to change in a standard program automatically generated is the pickupHeight value in the processInfo array. The variables are related to deciding whether to pick up or place down and the relevant heights of picking up and placing down.
Other variables can be changed by the user when necessary. Refer to [3.1. Variable configuration] for more details.

3.4.2. Position recording

There are 2 main basic positions that shall be recorded in the standard program. The first one is the pose variable that indicates the package position, and the second one is the variable that is needed to create the user coordinate on the pallet.

(1) Pose recording related to the pickup position

The variable related to the package position is defined in the 0400.HomePos file. The workpiece pickup position, slip sheet pickup position, and home position are defined for each process. The user can move the robot to a desired position by using the teaching pendant directly with the robot or through simulation program, and change the relevant pose variable to a pose constant value (Refer to '3.3.5 Home position and workpiece position pose program).
When a package is to be used for a loading pattern, teaching for the Home Position and the Pickup Position #1 shall only be carried out. If multiple packages are to be used, teaching for all relevant pickup positions shall be carried out.

(2) Defining the user coordinate on the pallet

The shift data of the patterns created by HRpal or HRpalware are all recorded as relative shift values against the #1 package position. Accordingly, it is needed to create the user coordinate based on the #1 package loading position (or teaching position and marked with *) on the pallet, and the relevant information is defined in the Pallet coordinate creating programs (Job programs #601 ~ #616).
Refer to [3.3.8 Pallet coordinate creating program] for more details.

GRC: 경기도 성남시 분당구 분당수서로 477
대구: 대구광역시 달성군 유가읍 테크노순환로 3 길 50
울산: 울산광역시 북구 매곡산업로 21 자동차조선기술관 201-5 호
중부: 충남 아산시 염치읍 송곡길 161
광주: 광주광역시 광산구 평동산단로 170-3 B 동 101 호
ARS 1588-9997 | 1 로봇영업 2 서비스영업 3 구매상담 4 고객지원 5 투자문의 6 채용 및 일반 문의
www.hyundai-robotics.com