ROBOTIS Tech Support v1.00

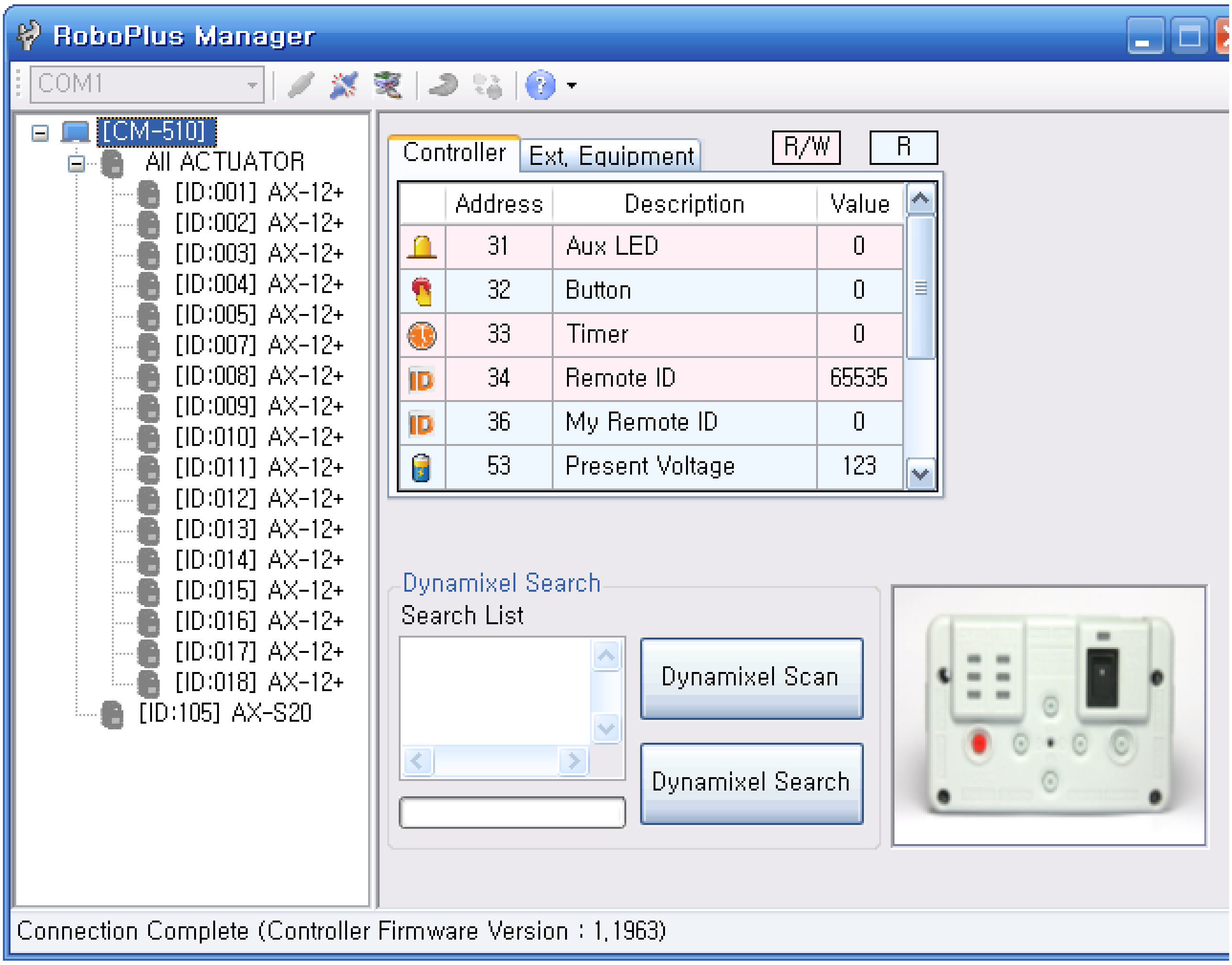
RoboPlus Manager

Last updated 2010.1.21 (v1.0 Eng)

RoboPlus Manager is used to handle devices used by a robot.

Major functions of this program are as follows.

- Manage controller firmware. (Update and Restore)
- Inspect the status of the controller and peripheral devices. (Test)
- Set the required modes. (Settings)



RoboPlus Manager

COM1

[CM-510]
All ACTUATOR
[ID:001] AX-12+
[ID:002] AX-12+
[ID:003] AX-12+
[ID:004] AX-12+
[ID:005] AX-12+
[ID:007] AX-12+
[ID:008] AX-12+
[ID:009] AX-12+
[ID:010] AX-12+
[ID:011] AX-12+
[ID:012] AX-12+
[ID:013] AX-12+
[ID:014] AX-12+
[ID:015] AX-12+
[ID:016] AX-12+
[ID:017] AX-12+
[ID:018] AX-12+
[ID:105] AX-S20

Controller | Ext. Equipment | R/W | R

Address	Description	Value
31	Aux LED	0
32	Button	0
33	Timer	0
34	Remote ID	65535
36	My Remote ID	0
53	Present Voltage	123

Dynamixel Search

Search List

Dynamixel Scan

Dynamixel Search

Connection Complete (Controller Firmware Version : 1.1963)

Home > Software Help > RoboPlus > RoboPlus Manager > Getting Started > Connect Controller

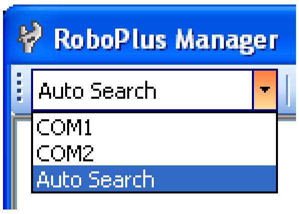
ROBOTIS Tech Support v1.00

Connect Controller

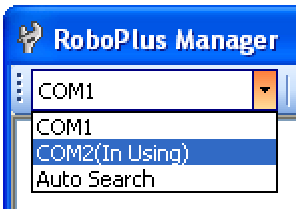
Last updated 2010.1.21 (v1.0 Eng)

- **Connect controller to the PC. (Please refer to controller information for information on how to connect the controller to the PC.)**
- **Select the communication port to use.**

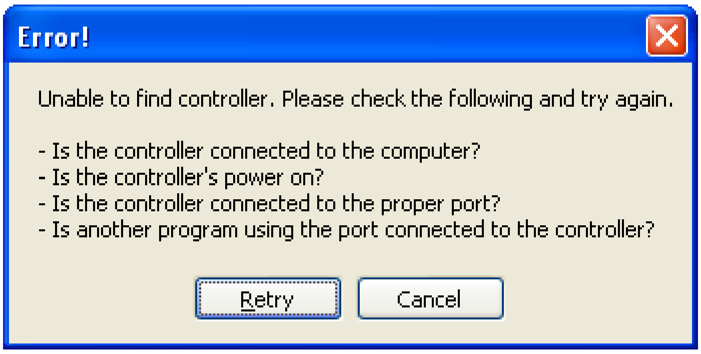
  Use the "Automatic Search" function to easily select the appropriate port.

  If the chosen communication port is being used by another program, you must first find and stop the program.

  If RoboPlus Manager is unable to find a controller, the following error message will be shown.

  Error!

  Unable to find controller. Please check the following and try again.

  - Is the controller connected to the computer?
  - Is the controller's power on?
  - Is the controller connected to the proper port?
  - Is another program using the port connected to the controller?

  - Check if the controller is connected to the PC. (See controller information for information on how to connect the controller.)
  - Check if the controller is turned on.
  - Check if the correct communication port was chosen.

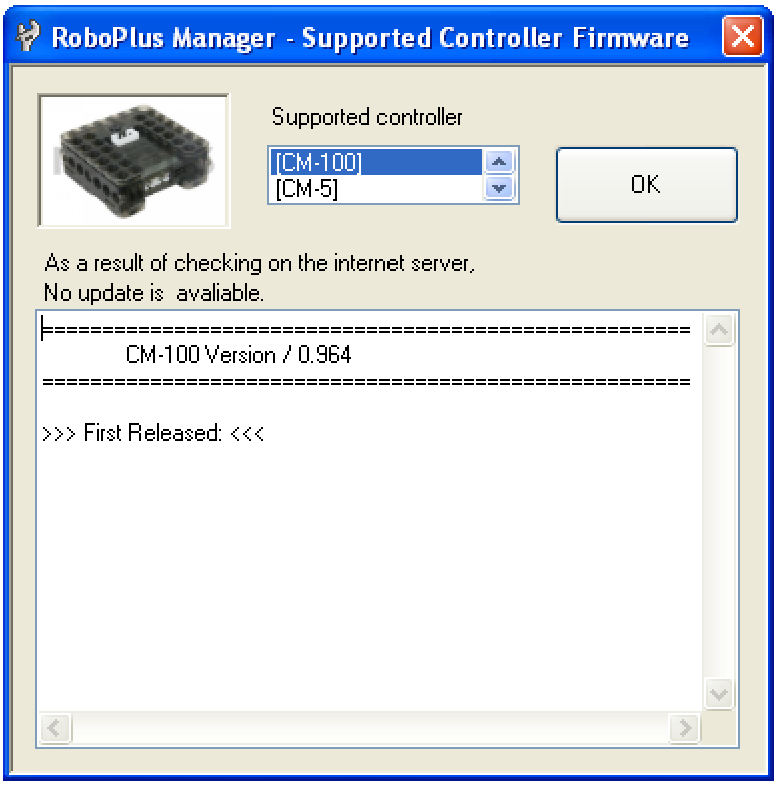
- **Start management. (Please refer to the managing information of each controller.)**

ROBOTIS Tech Support v1.00

Firmware Update

Last updated 2010.1.21 (v1.0 Eng)

Firmware is the program installed in the controller, and is used to execute .tsk programs or to manage the controller. RoboPlus Manager automatically connects to the internet and searches for firmware updates.

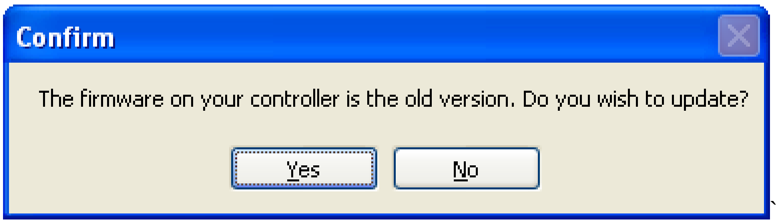


As a result of checking on the internet server,
No update is avaliable.

```
======================================================
          CM-100 Version / 0.964
======================================================

>>> First Released: <<<
```

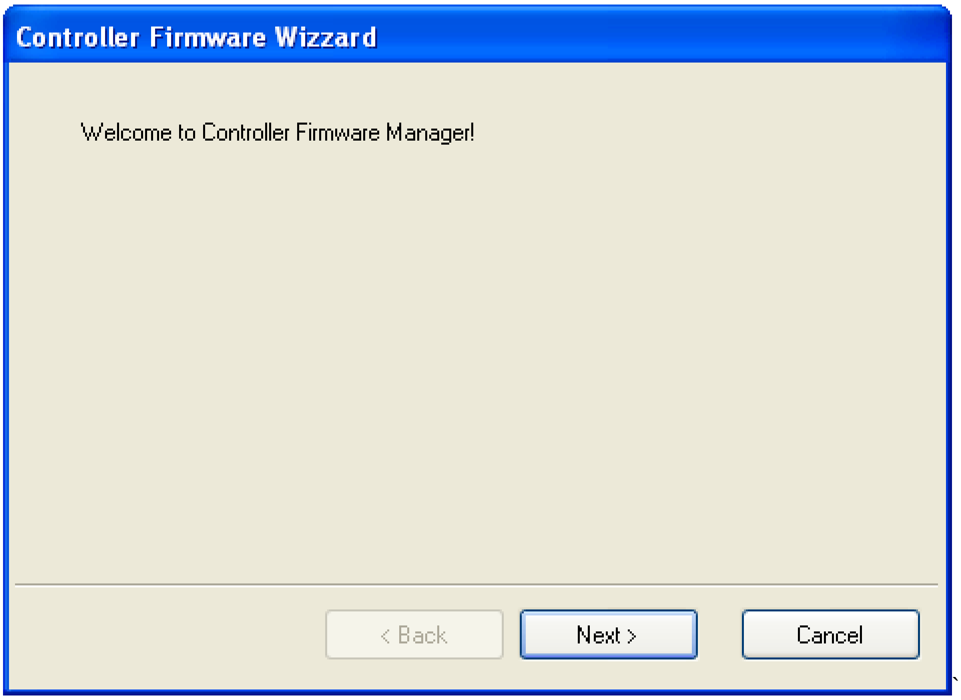
Updating the Controller's Firmware

1. When the controller is connected, the controller's firmware version will be retrieved. If a newer firmware is available, it will ask whether to download the latest firmware.



The firmware on your controller is the old version. Do you wish to update?

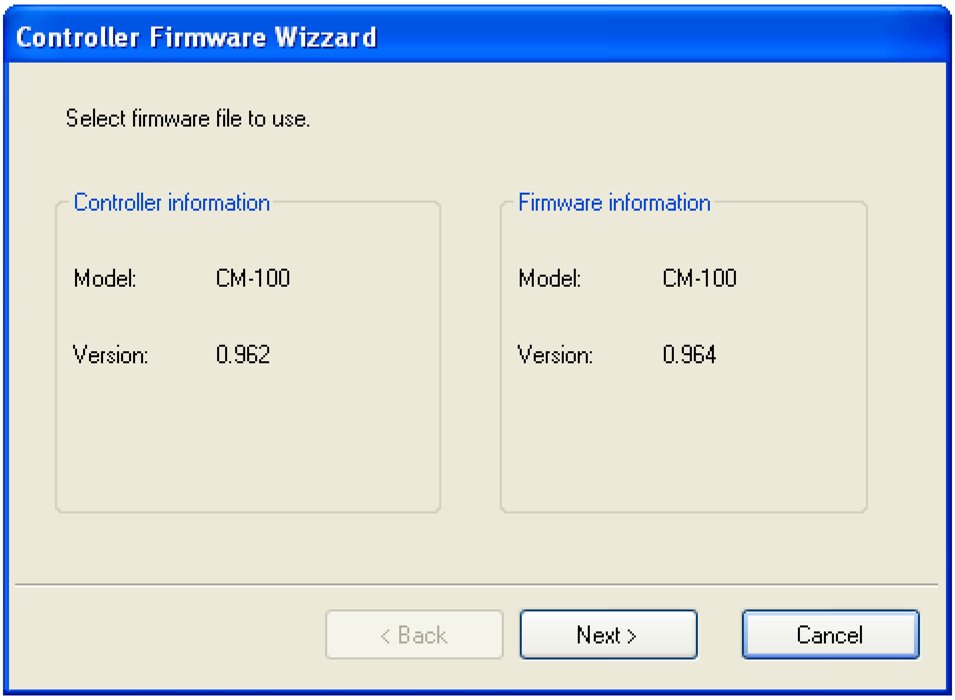
2. When the 'Yes(Y)' button is pressed, it will begin updating the firmware.



Controller Firmware Wizzard

Welcome to Controller Firmware Manager!

3. You can check your controller's model number and firmware version.



Controller Firmware Wizzard

Select firmware file to use.

Controller information

Model: CM-100

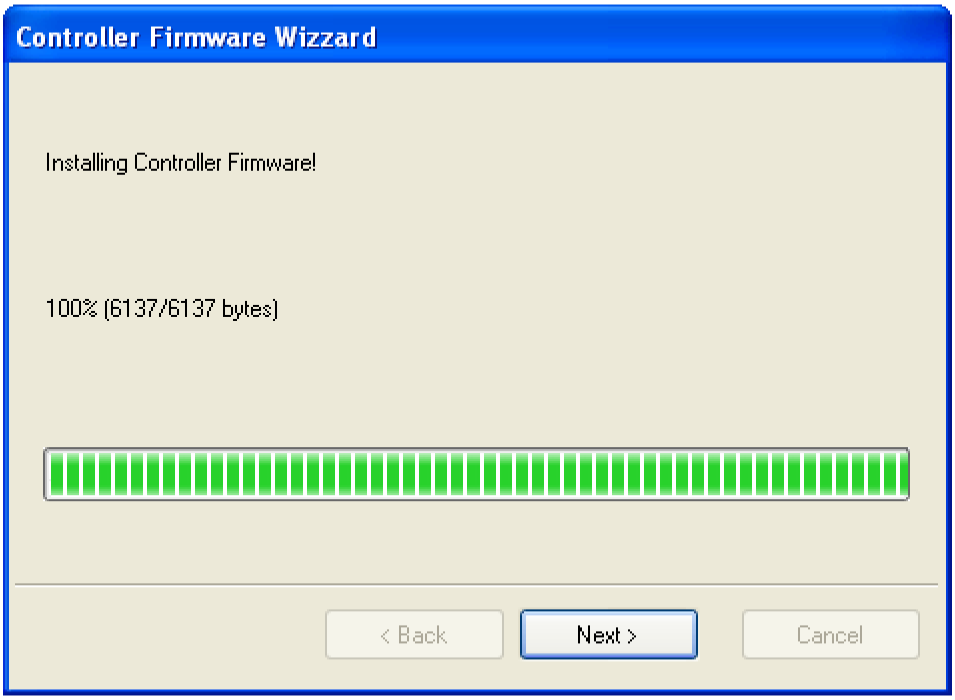
Version: 0.962

Firmware information

Model: CM-100

Version: 0.964

4. Press the "Next" button to begin updating your firmware. Be careful not to turn the power off or disconnect the cable while the firmware is being updated.

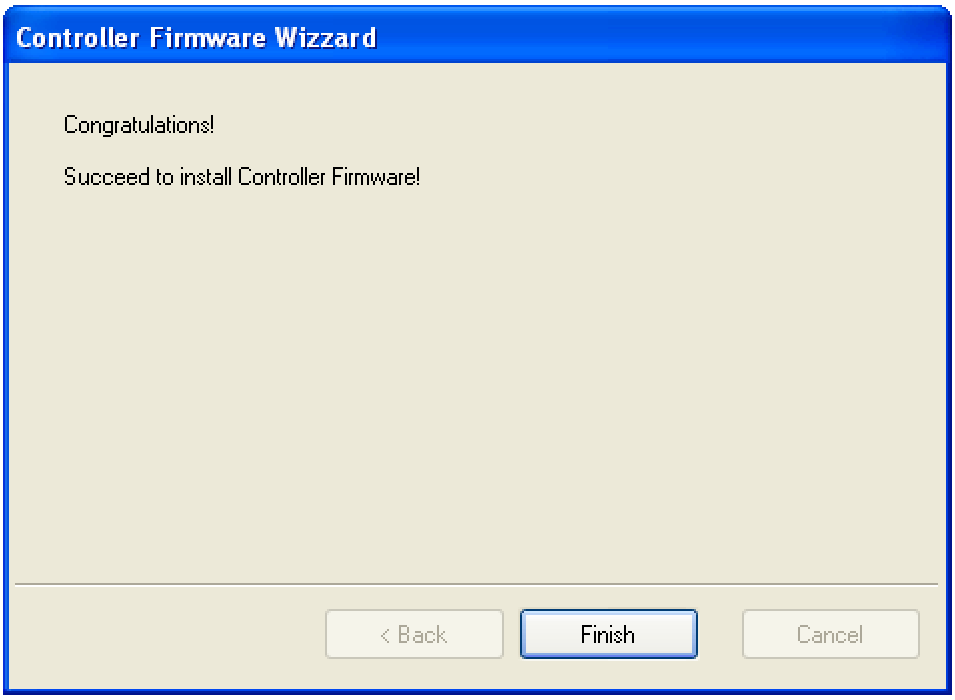


Controller Firmware Wizzard

Installing Controller Firmware!

100% (6137/6137 bytes)

5. Once the firmware has been updated, Press the "Finish" button to return to the controller maintenance page.



Controller Firmware Wizzard

Congratulations!

Succeed to install Controller Firmware!

Firmware Restoration

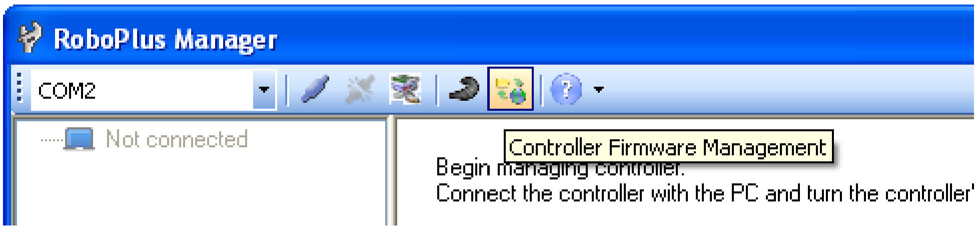
Last updated 2010.1.21 (v1.0 Eng)

The controller's firmware can be restored using RoboPlus Manager. This is used when the current firmware is causing problems and you would like to revert to an older version.

Restoring the Controller's Firmware

1. **Run "Controller Firmware Wizard."**

   Press the "Controller Firmware Management" icon.



RoboPlus Manager

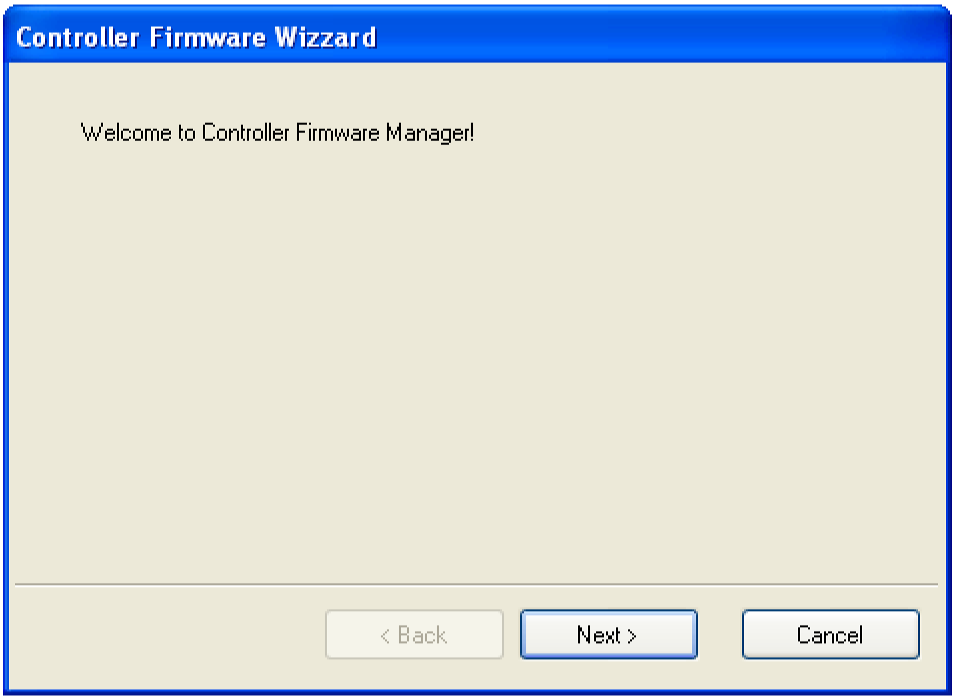
COM2

Not connected

Controller Firmware Management

Begin managing controller.

Connect the controller with the PC and turn the controller'

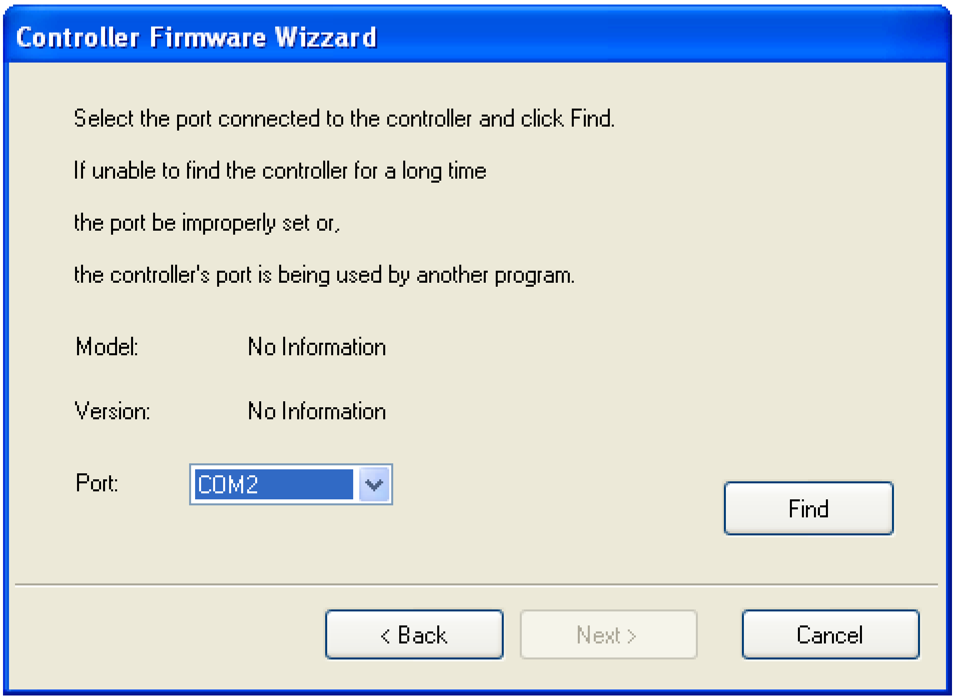


Controller Firmware Wizzard

Welcome to Controller Firmware Manager!

2. **Select the communication port to use**

   Because the firmware cannot be automatically detected, the user must manually select the port to which the controller is connected. If the chosen communication port is being used by another program, you must first find and stop the program. Press the "Find" button after selecting the port.



Controller Firmware Wizzard

Select the port connected to the controller and click Find.

If unable to find the controller for a long time

the port be improperly set or,

the controller's port is being used by another program.

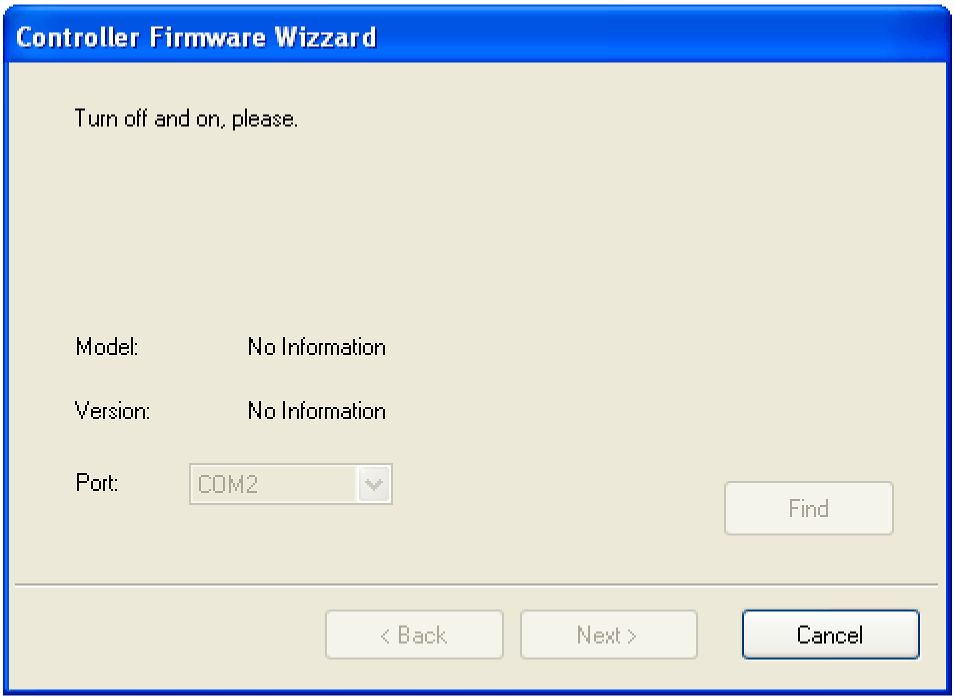
Model: No Information

Version: No Information

Port: COM2

3. **Restart the controller**

   The controller must be turned off and turned back on.



Controller Firmware Wizzard

Turn off and on, please.

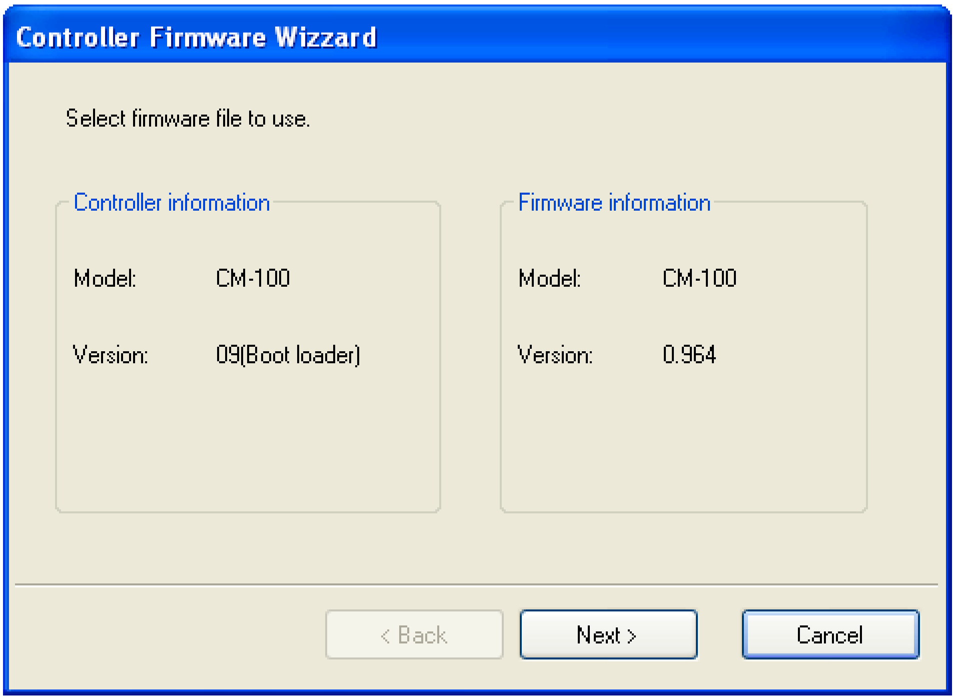
Model: No Information

Version: No Information

Port: COM2

4. **Check controller information**

   When the controller is detected, its model name and version number are shown. Verify that the controller information is correct. ("Version" in "controller information" refers to the bootloader version, not the firmware version.)



Controller Firmware Wizzard

Select firmware file to use.

Controller information

Model: CM-100

Version: 09(Boot loader)

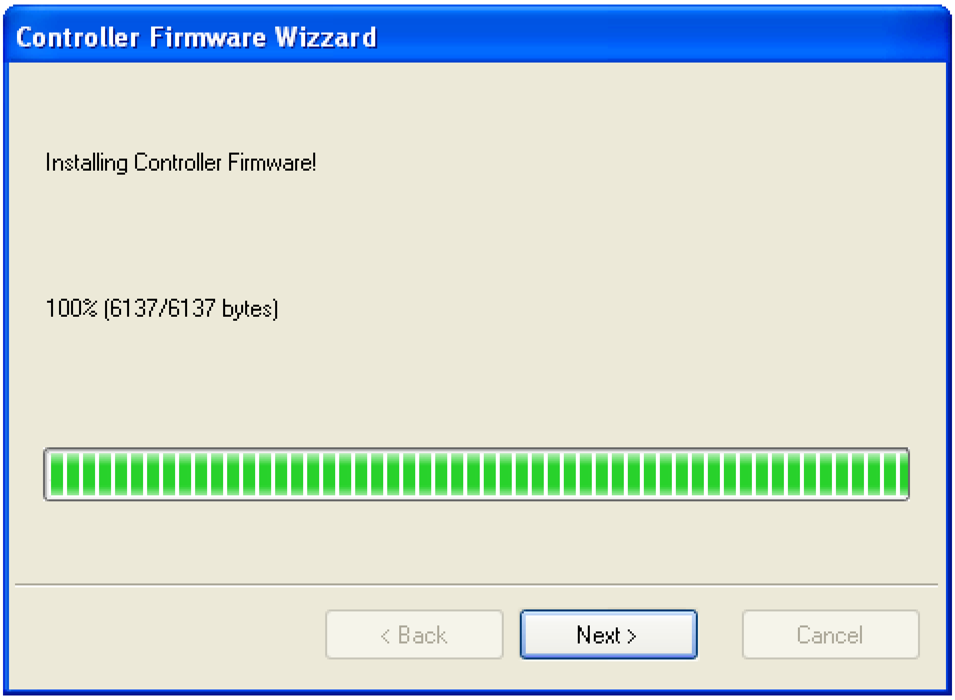
Firmware information

Model: CM-100

Version: 0.964

5. **Firmware Restoration**

 Press the "Next" button to begin restoring the firmware. Be careful not to turn the power off or to disconnect the cable while the firmware is being updated.

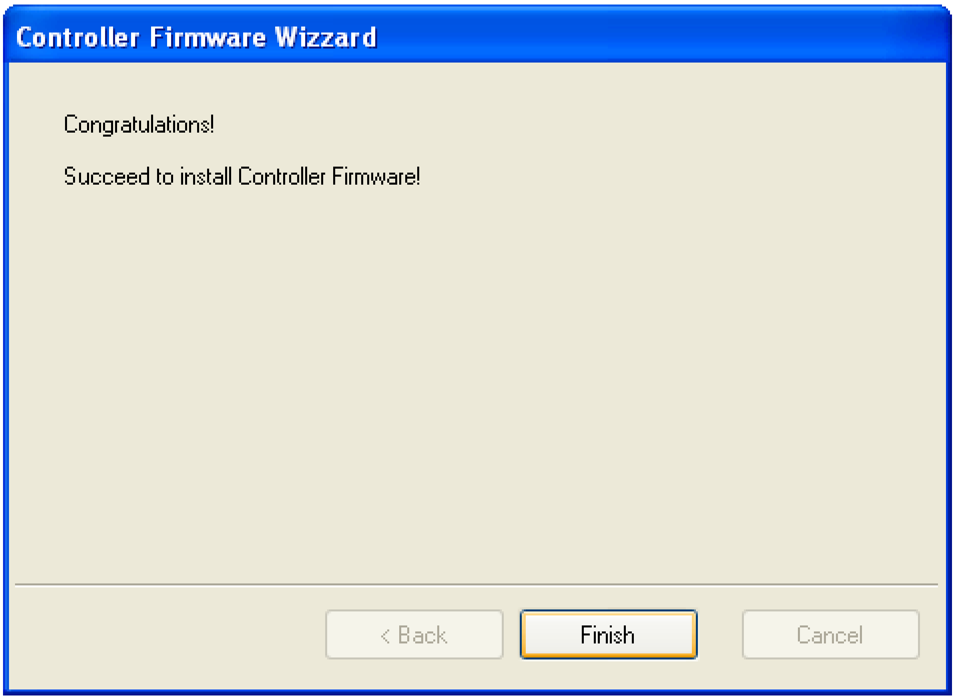


Controller Firmware Wizzard

Installing Controller Firmware!

100% (6137/6137 bytes)

6. Once the firmware has been updated, press the "Finish" button to return to the controller maintenance page.



Controller Firmware Wizzard

Congratulations!

Succeed to install Controller Firmware!

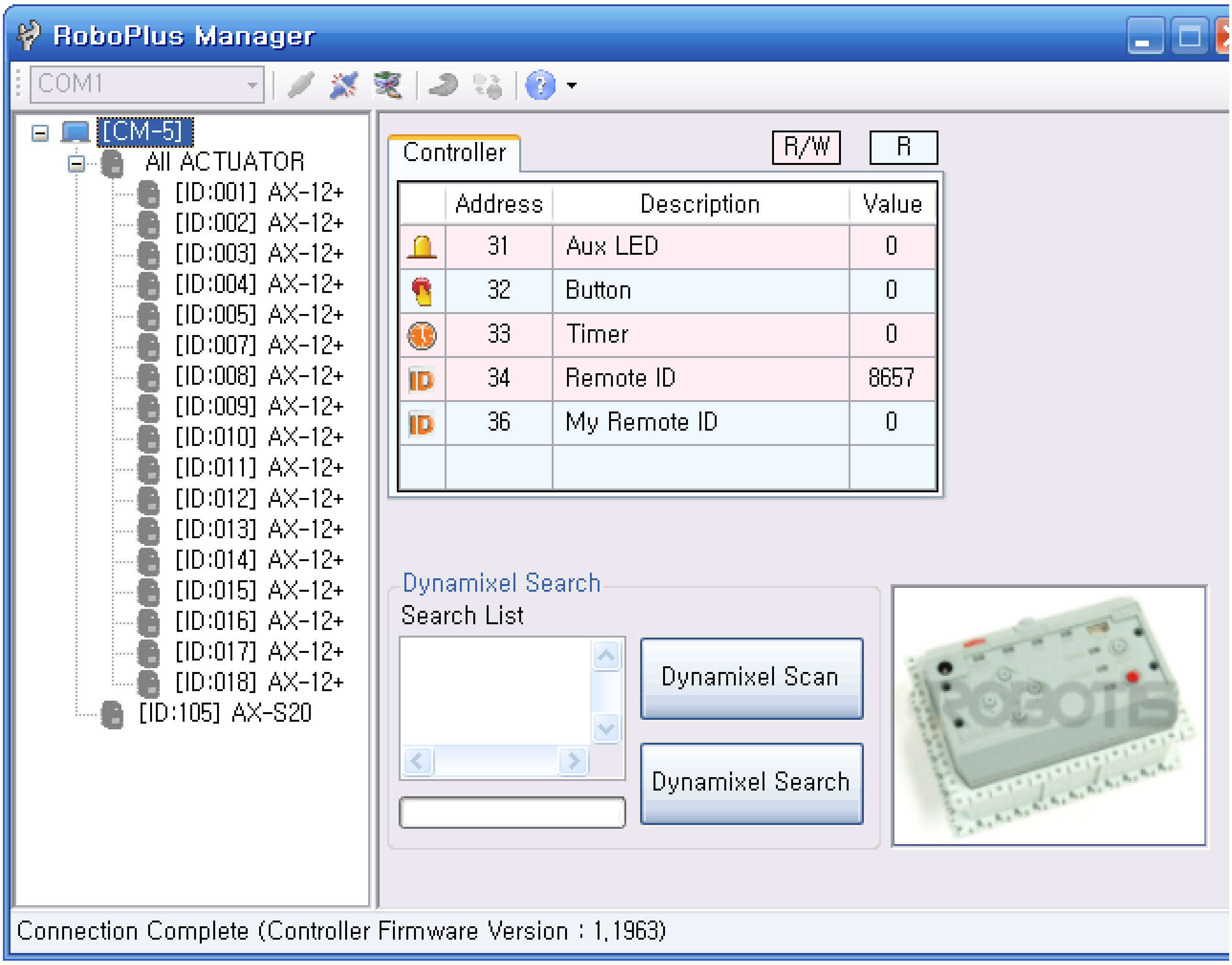
Home > Software Help > RoboPlus > RoboPlus Manager > Test and Set up > Controller > CM-5

ROBOTIS Tech Support v1.00

CM-5

Last updated 2010.1.21 (v1.0 Eng)

When the CM-5 controller is connected to RoboPlus Manager, the following screen is displayed.



- [CM-5]
  - All ACTUATOR
    - [ID:001] AX-12+
    - [ID:002] AX-12+
    - [ID:003] AX-12+
    - [ID:004] AX-12+
    - [ID:005] AX-12+
    - [ID:007] AX-12+
    - [ID:008] AX-12+
    - [ID:009] AX-12+
    - [ID:010] AX-12+
    - [ID:011] AX-12+
    - [ID:012] AX-12+
    - [ID:013] AX-12+
    - [ID:014] AX-12+
    - [ID:015] AX-12+
    - [ID:016] AX-12+
    - [ID:017] AX-12+
    - [ID:018] AX-12+
  - [ID:105] AX-S20

	Address	Description	Value
	31	Aux LED	0
	32	Button	0
	33	Timer	0
ID	34	Remote ID	8657
ID	36	My Remote ID	0

Connection Complete (Controller Firmware Version : 1.1963)

The controller and its peripheral devices are listed in the left window. In the right window is the "controller management" window. (Please refer to the "Test" page).

If the controller's firmware version is older than that of RoboPlus Manager, it can be updated. (Please refer to the "Firmware Update" page.)

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Edit > Select Multiple Lines

ROBOTIS Tech Support v1.00

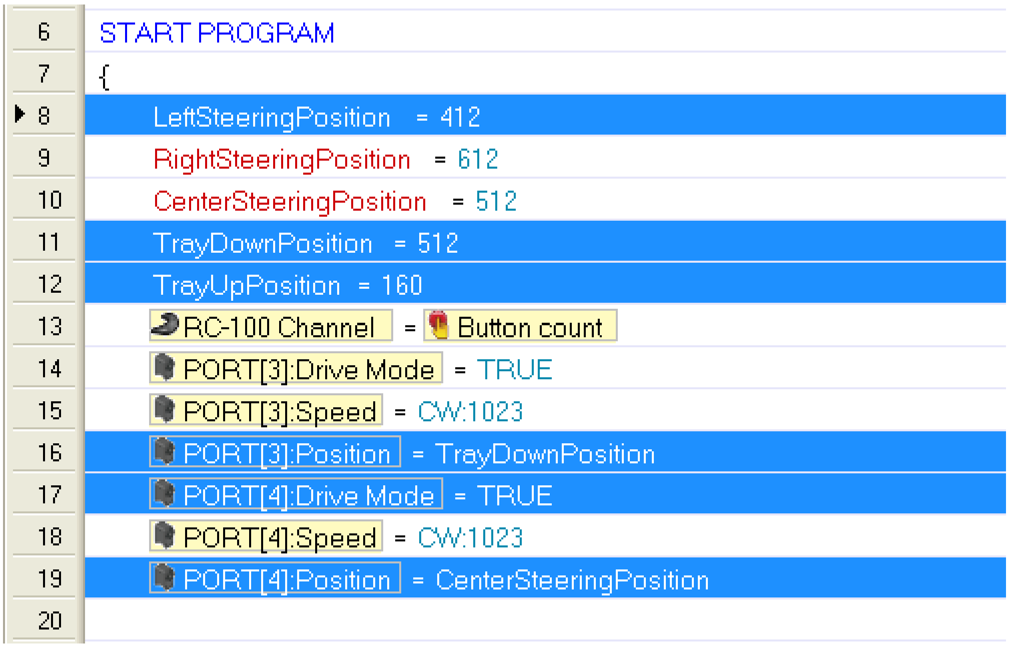
Select Multiple Lines

Last updated 2010.1.15 (v1.0 Eng)

RoboPlus Task provides a function to select and edit (cut, copy, delete, comment,etc) multiple lines of program code.

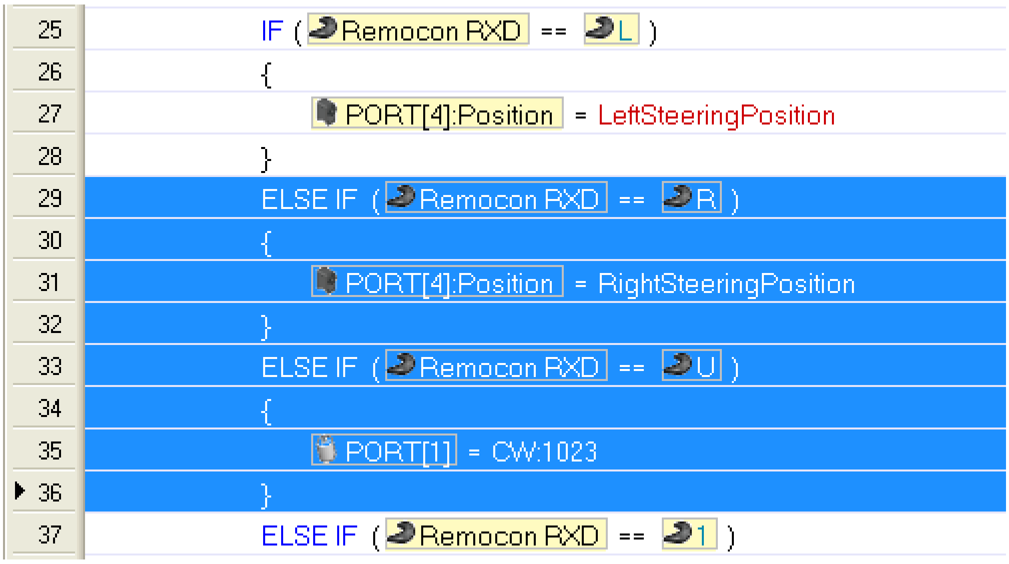
There are multiple ways to select multiple lines.

- While pressing the Ctrl key, click on the lines with the mouse.



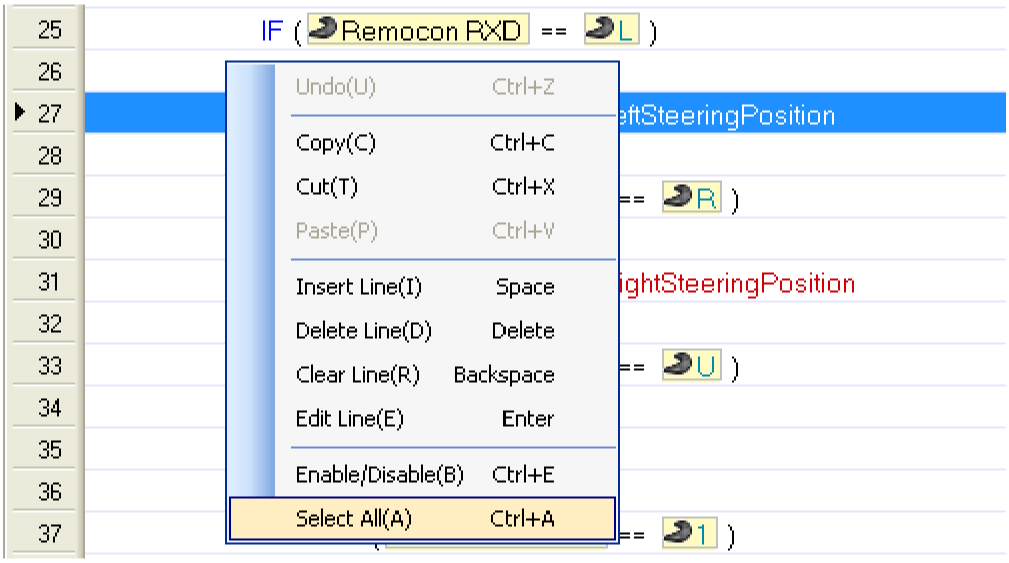
```
START PROGRAM
{
    LeftSteeringPosition   = 412
    RightSteeringPosition   = 612
    CenterSteeringPosition   = 512
    TrayDownPosition  = 512
    TrayUpPosition  = 160
    RC-100 Channel  = Button count
    PORT[3]:Drive Mode  = TRUE
    PORT[3]:Speed  = CW:1023
    PORT[3]:Position  = TrayDownPosition
    PORT[4]:Drive Mode  = TRUE
    PORT[4]:Speed  = CW:1023
    PORT[4]:Position  = CenterSteeringPosition

```

- Click on the first line, and while pressing the Shift key, click on the last. The lines between the two lines will be selected.
- Click and drag on the lines you want to select.



```
            IF ( Remocon RXD  ==  L )
            {
                PORT[4]:Position  = LeftSteeringPosition
            }
            ELSE IF ( Remocon RXD  ==  R )
            {
                PORT[4]:Position  = RightSteeringPosition
            }
            ELSE IF ( Remocon RXD  ==  U )
            {
                PORT[1]  = CW:1023
            }
            ELSE IF ( Remocon RXD  ==  1 )
```

- To select all lines, right-click on the code, then click select "SelectAll." You may also press Ctrl + A.


IF (Remocon RXD == L)
eftSteeringPosition
== R)
ightSteeringPosition
== U)
== 1)
Undo(U) Ctrl+Z
Copy(C) Ctrl+C
Cut(T) Ctrl+X
Paste(P) Ctrl+V
Insert Line(I) Space
Delete Line(D) Delete
Clear Line(R) Backspace
Edit Line(E) Enter
Enable/Disable(B) Ctrl+E
Select All(A) Ctrl+A

ROBOTIS Tech Support v1.00

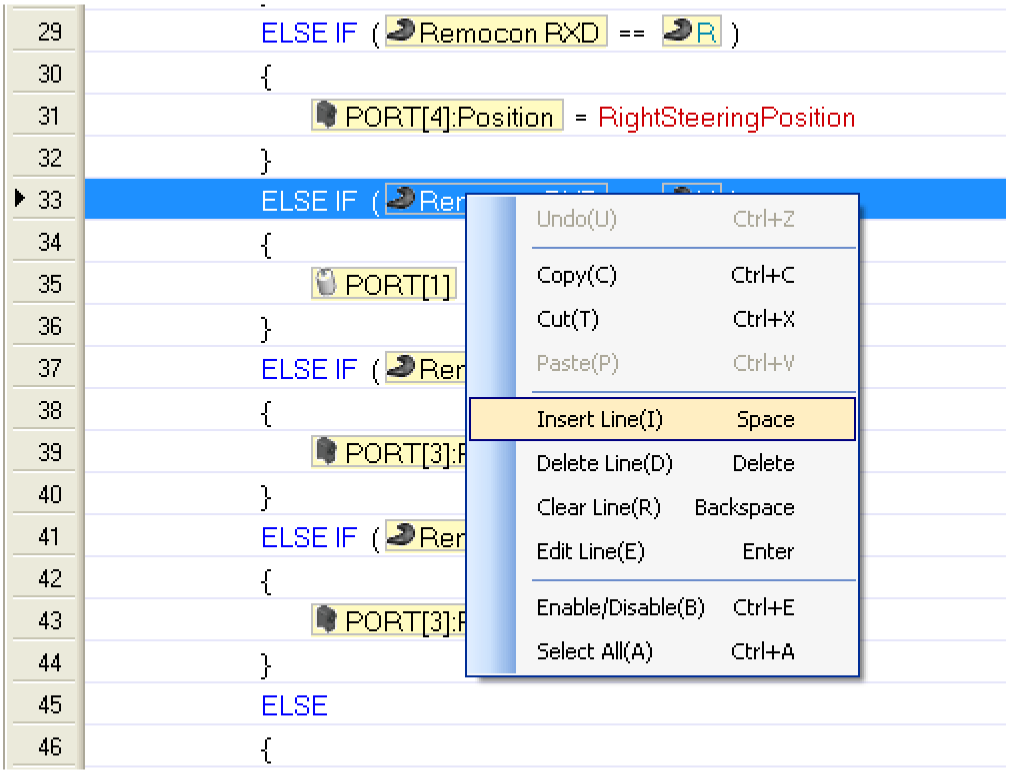
Insert a New Line

Last updated 2010.1.15 (v1.0 Eng)

RoboPlus Task provides a function to insert a line between existing lines.

There are multiple ways to insert new lines.(New lines will be inserted below the highlighted line.

- Press the Space bar.
- Right click, then select "Insert Line'".



```
ELSE IF ( Remocon RXD == R )
{
    PORT[4]:Position = RightSteeringPosition
}
ELSE IF ( Rer
{
    PORT[1]
}
ELSE IF ( Rer
{
    PORT[3]:F
}
ELSE IF ( Rer
{
    PORT[3]:F
}
ELSE
{
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Edit > Delete Lines

ROBOTIS Tech Support v1.00

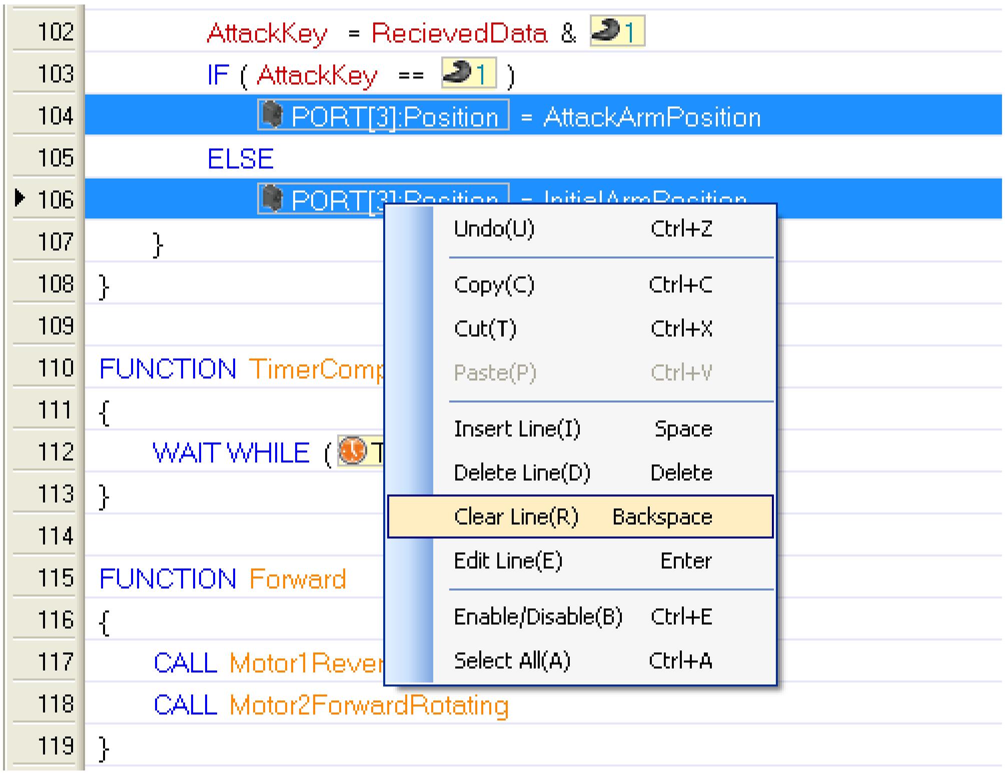
Delete Lines

Last updated 2010. 1.15 (v1.0 Eng)

RoboPlus Task provides a function to delete lines. One or more lines may be deleted at once.

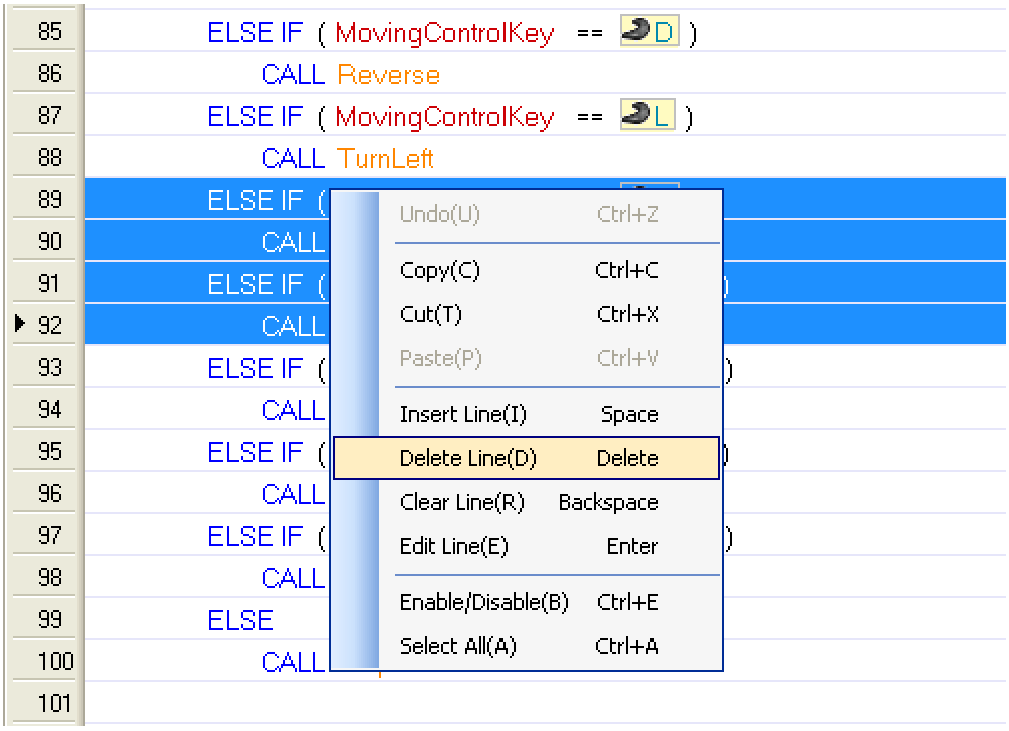
There are multiple ways to delete lines.

- Select the line(s) to delete, right-click, and click "Clear Line." The lines will be cleared, resulting in blank lines. You may also press the Backspace (←) key.



```
            AttackKey  = RecievedData & 1
            IF ( AttackKey  ==  1 )
                PORT[3]:Position = AttackArmPosition
            ELSE
                PORT[3]:Position = InitialArmPosition
        }
}

FUNCTION TimerComp
{
        WAIT WHILE ( T
}

FUNCTION Forward
{
        CALL Motor1Rever
        CALL Motor2ForwardRotating
}
```

- Select the line(s) to delete, right-click, and click "Delete Line." The lines will be deleted, and the lines below the deleted lines will move up. You may also press the Delete key.



```
ELSE IF ( MovingControlKey == D )
    CALL Reverse
ELSE IF ( MovingControlKey == L )
    CALL TurnLeft
ELSE IF (
    CALL
ELSE IF (
    CALL
ELSE IF (
    CALL
ELSE IF (
    CALL
ELSE IF (
    CALL
ELSE
    CALL
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Edit > Enable/ Disable Lines

ROBOTIS Tech Support v1.00

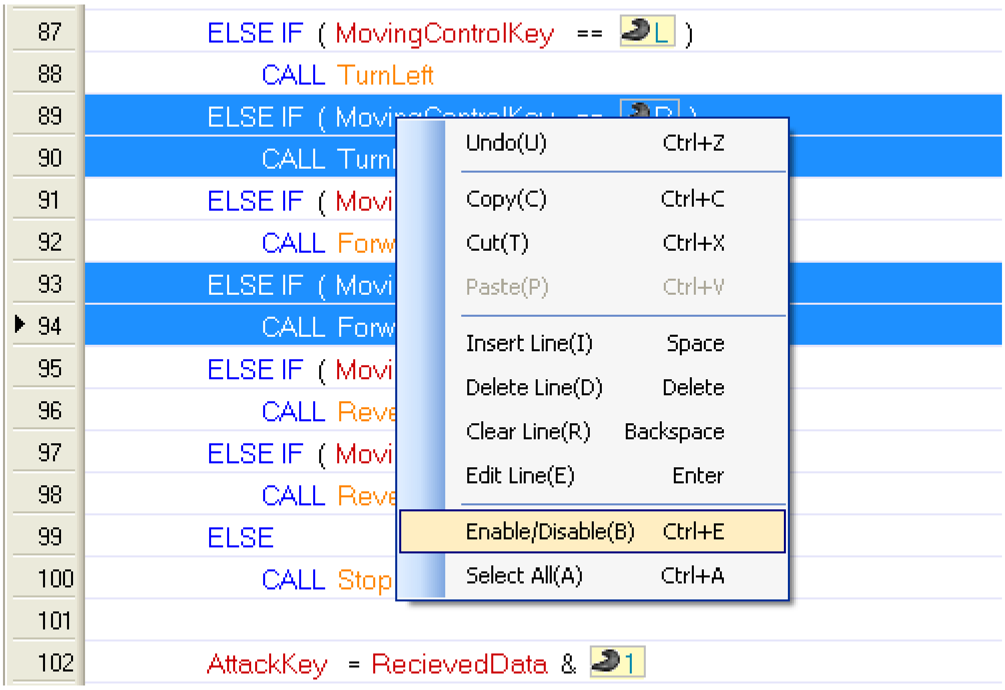
Enable/Disable Lines

Last updated 2010. 1.15 (v1.0 Eng)

RoboPlus Task provides a function to enable or disable lines.

There are multiple ways to enable/disable lines.

- Select the line to enable or disable, right-click, and click "Enable/Disable".
- Select the line to enable or disable, and press Ctrl + E



```
        ELSE IF ( MovingControlKey == L )
            CALL TurnLeft
        ELSE IF ( Moving
            CALL Turnl
        ELSE IF ( Movi
            CALL Forw
        ELSE IF ( Movi
            CALL Forw
        ELSE IF ( Movi
            CALL Reve
        ELSE IF ( Movi
            CALL Reve
        ELSE
            CALL Stop

        AttackKey = RecievedData & 1
```

- The function will enable disabled lines and disable enabled lines.
- This function is commonly used to keep certain commands from being executed in certain situations, such as when testing code.

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Edit > Copy/Cut/Paste

ROBOTIS Tech Support v1.00

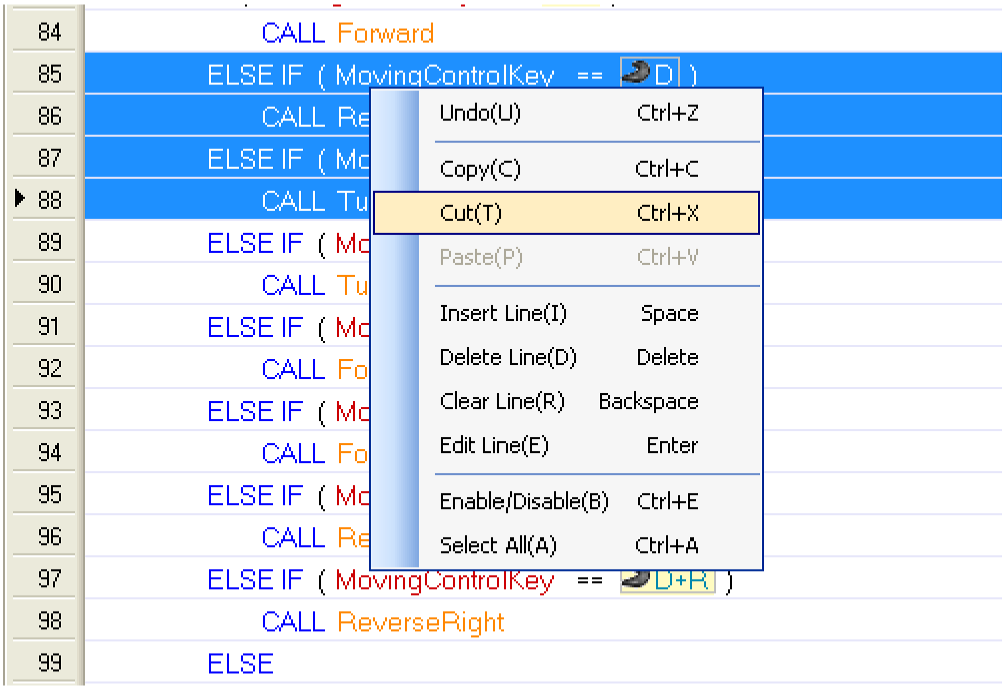
Copy/Cut/Paste

Last updated 2010.1.15 (v1.0 Eng)

RoboPlus Task provides a function to cut, copy and paste lines.

Cut

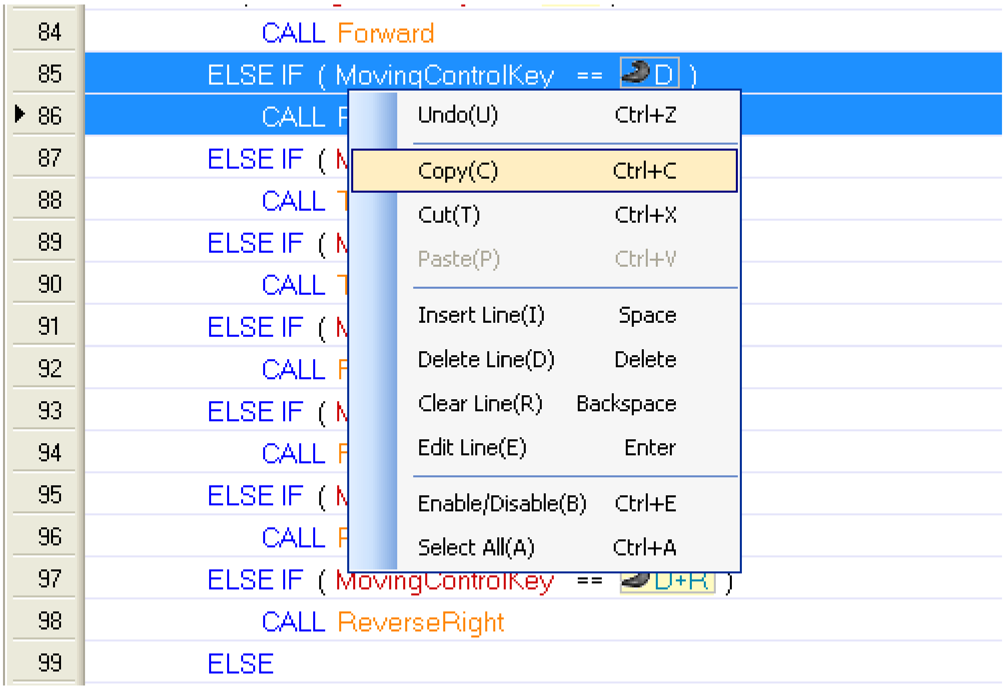
- Select one or more lines, right-click, and click "Cut".
- The selected line will be deleted and stored in a temporary clipboard.
- When you perform the "Cut" function, the data in the clipboard will be replaced with the new selection.
- The shortcut is Ctrl+X.



```
            CALL Forward
        ELSE IF ( MovingControlKey == D )
            CALL Re
        ELSE IF ( Mc
            CALL Tu
        ELSE IF ( Mc
            CALL Tu
        ELSE IF ( Mc
            CALL Fo
        ELSE IF ( Mc
            CALL Fo
        ELSE IF ( Mc
            CALL Re
        ELSE IF ( MovingControlKey == D+R )
            CALL ReverseRight
        ELSE
```

Copy

- Select one or more lines, right-click, and click 'Copy'.
- The selected line will be copied to a temporary clipboard.
- When you perform the 'Copy' function, the data in the clipboard will be replaced with the new selection.
- The shortcut is Ctrl+C.

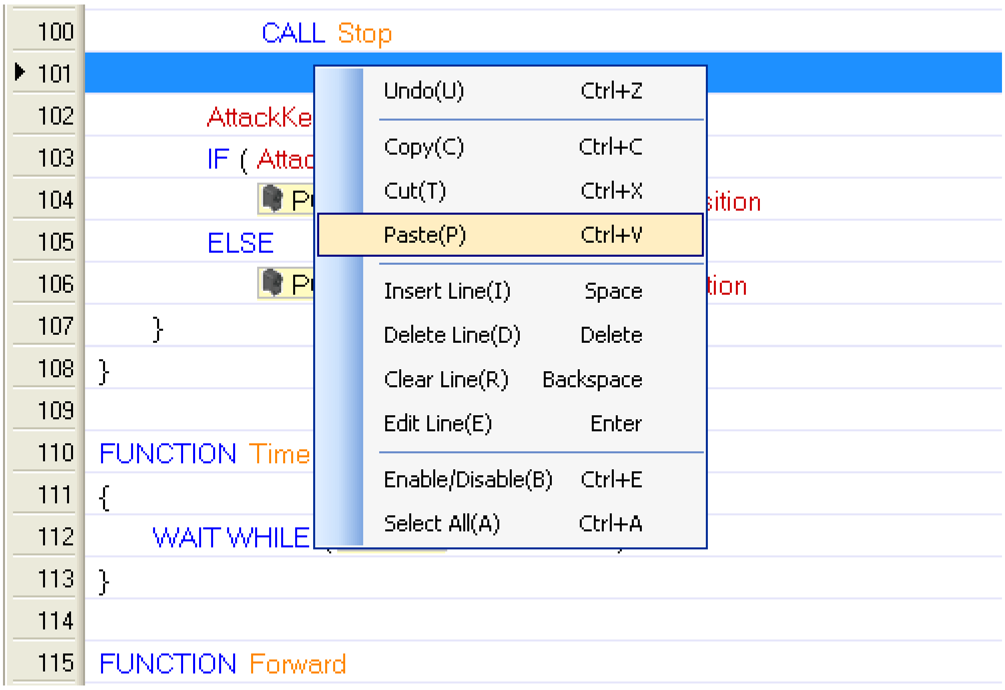


```
                CALL Forward
            ELSE IF ( MovingControlKey == D )
                CALL F
            ELSE IF ( M
                CALL T
            ELSE IF ( M
                CALL T
            ELSE IF ( M
                CALL F
            ELSE IF ( M
                CALL F
            ELSE IF ( M
                CALL F
            ELSE IF ( MovingControlKey == D+R )
                CALL ReverseRight
            ELSE
```

Paste

- You can use this function only when there is data in the clipboard.
- Select the line where the data will be pasted, right-click, and click "Paste."
- The data in the clipboard will remain even after it has been pasted, so you can paste the same data many times.
- If you perform the "Paste" function on a line with code, the code will be overwritten with data from the clipboard.

- The shortcut is Ctrl+V.



```
                CALL Stop

            AttackKe
            IF ( Attac
                P                          ition
            ELSE
                P                          tion
        }
}

FUNCTION Time
{
        WAIT WHILE
}

FUNCTION Forward
```

ROBOTIS Tech Support v1.00

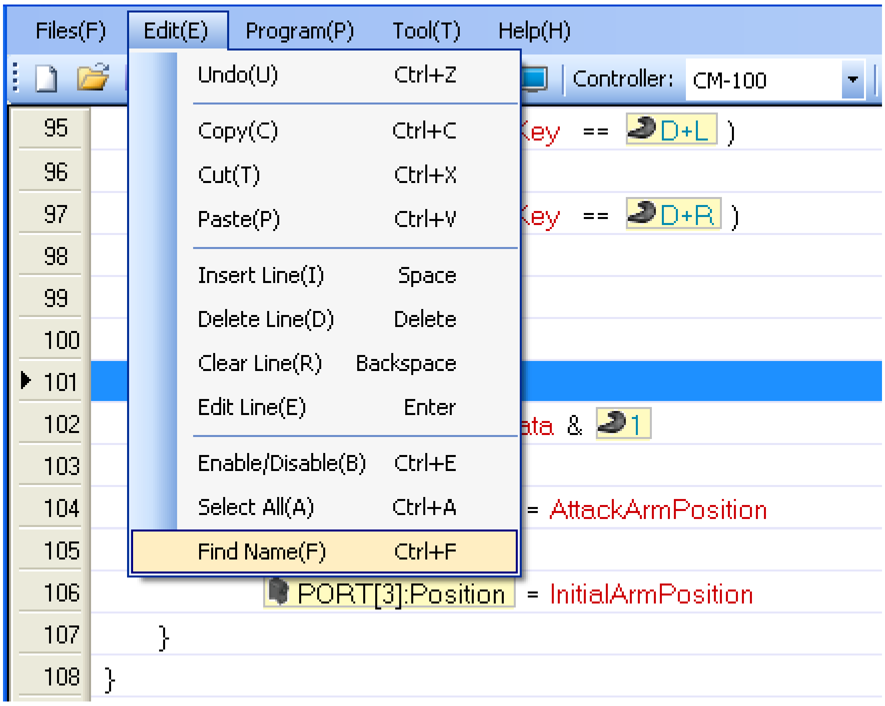
Find Name

Last

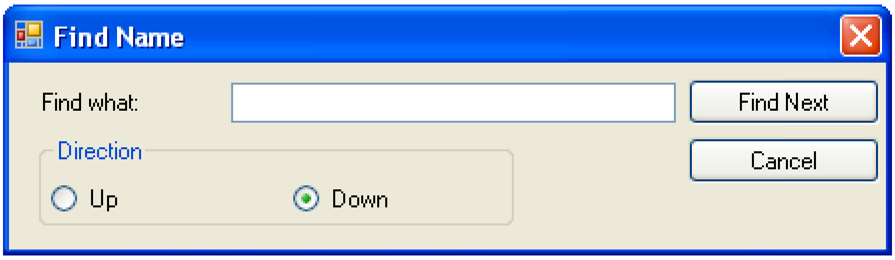
updated 2010.1.15 (v1.0 Eng)

RoboPlus Task provides a function to search for used elements.

1. From the "'edit" menu, select the"'Find Name" function. The shortcut is Ctrl+F.



2. Enter a keyword and click the "Find Next" button.



Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Start Program

ROBOTIS Tech Support v1.00

Start Program

Last updated 2010.1.15 (v1.0 Eng)

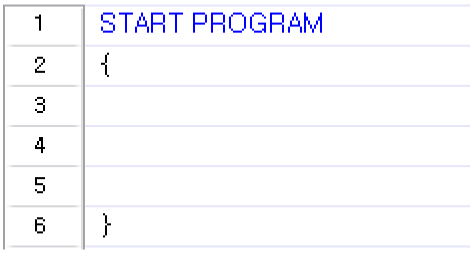
"Start Program" designates the beginning of a program. Regardless of the line number, the program will always start at this point. "Start Program" is like the "main" function in the C language.

Usage

- "Start Program" is executed regardless of its line number.
- A program cannot have more than one "Start Program" command.
- The body of the command must be enclosed by brackets.
- The program will end when the closing bracket(}) is reached.

Example

- Start a program with the "Start Program" command.



```
START PROGRAM
{



}
```

ROBOTIS Tech Support v1.00

Loop While

Last updated 2010. 1.18 (v1.0 Eng)

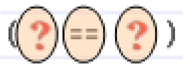
This command is used to repeat the command lines in the block while the clause is true.

It is the equivalent of the "while"function in C language..

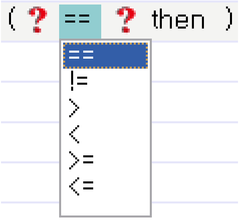
What is a conditional clause?

Conditional clause is a feature to perform different actions depending on whether the condition evaluates to true (condition is met) or false (condition is not met).

Conditional clause is composed of the following 3 parts: parameter 1, relational operator, and parameter 2.



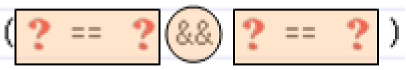
These are 6 types of relational operators.



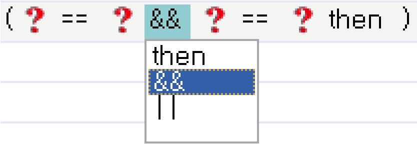
- **==** : True if the two parameters are equal.
- **!=** : True if the two parameters are not equal.
- **>=** : True if parameter 1 is greater than or equal to parameter 2.
- **>** : True if parameter 1 is greater than parameter 2.
- **<=** : True parameter 1 is less than or equal to parameter 2.
- **<** : True if parameter 1 is less than parameter 2.

Conditional clause can be combined into a complex conditional clause using conditional operators.

A complex conditional clause is composed of the following 3 parts: conditional clause 1, conditional operator, conditional clause 2.



There are 3 types of conditional operators.

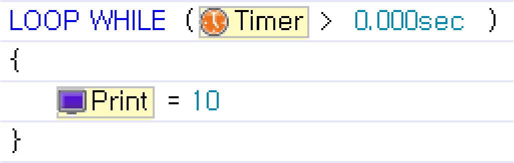


- **then** : Does not link any clauses.
- **AND(&&)**: True if both conditional clauses are true.
- **OR(||)**: True if one of the conditional clauses is true.

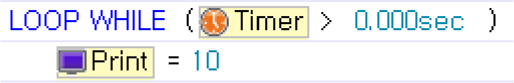
There is no limit to how many conditional clauses can be combined into one complex conditional clause. Each conditional clause is evaluated in order, and the final value will be either "true" or "false."

Usage

- A block is always required. (However, if the block consists of only one line, the block need not be enclosed with brackets.)



```
LOOP WHILE ( Timer > 0.000sec )
{
    Print = 10
}
```

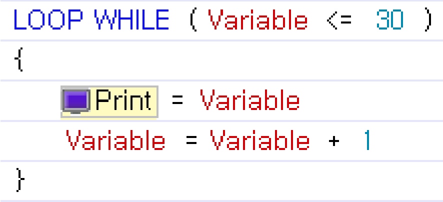


```
LOOP WHILE ( Timer > 0.000sec )
    Print = 10
```

- Use the Break Loop command to exit the loop.

Example

- Continuously prints the value of on the Program Output Monitor until the variable reaches 30.



```
LOOP WHILE ( Variable <= 30 )
{
    Print = Variable
    Variable = Variable + 1
}
```

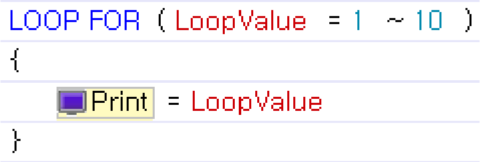
Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Loop For

ROBOTIS Tech Support v1.00

Loop For

Last updated 2010.1.18 (v1.0 Eng)

This command is used to repeat the command lines in the block for the specified number of times. Given an initial value and a terminal value, the loop will repeat while increasing the variable by 1.



```
LOOP FOR ( LoopValue  = 1  ~ 10  )
{
      Print  = LoopValue
}
```

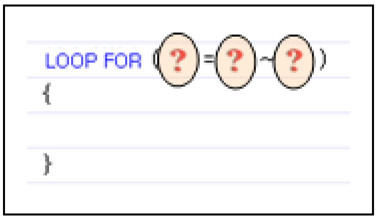
The number of executions can be calculated as:

- **Number to Executions = Terminal Value - Initial Value + 1**

This is the equivalent of the "for" function in C language.

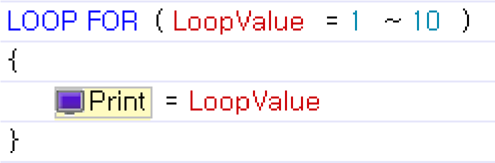
Usage

- Choose the appropriate 3 parameters (variable, Start value, End value) necessary for the command.

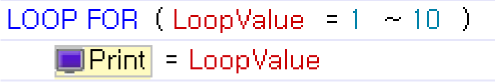


```
LOOP FOR (?=?~?)
{

}
```

- The initial value must be less than the terminal value. **If the initial value is greater than the terminal value, the loop will not be executed.**
- A block is always required. (However, if the block consists of only one line, the block need not be enclosed with brackets.)



```
LOOP FOR ( LoopValue  = 1  ~ 10  )
{
      Print  = LoopValue
}
```

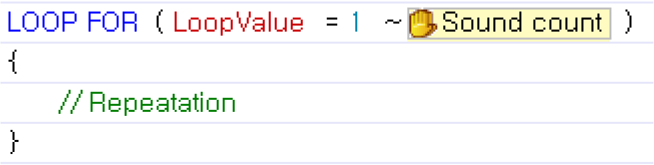


```
LOOP FOR ( LoopValue  = 1  ~ 10  )
      Print  = LoopValue
```

- Use the Break Loop command to exit the loop.

Example

- Repeat the loop for the number of detected sounds.



```
LOOP FOR ( LoopValue  = 1  ~ Sound count  )
{
    // Repeatation
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Break Loop

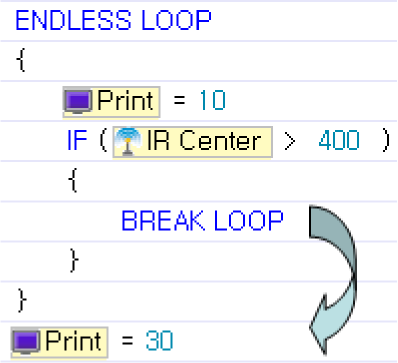
ROBOTIS Tech Support v1.00

Break Loop

Last updated 2010.1.18 (v1.0 Eng)

This command is used to exit the loop while it is being executed.

It is the equivalent of the "break" function in C language.



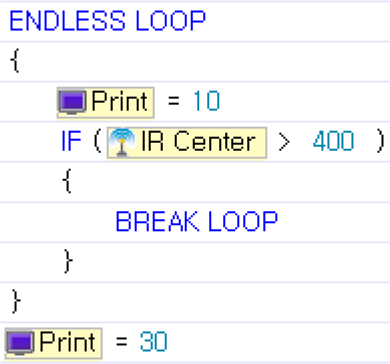
```
ENDLESS LOOP
{
    Print = 10
    IF ( IR Center > 400 )
    {
        BREAK LOOP
    }
}
Print = 30
```

Usage

- The command must always be used in the block being repeated.

Example

- Continuously prints the number "10" on the screen until the value of the center IR sensor becomes bigger than 400, in which case it exits the loop and prints "30" on the screen.



```
ENDLESS LOOP
{
    Print = 10
    IF ( IR Center > 400 )
    {
        BREAK LOOP
    }
}
Print = 30
```

[Home](#) > [Software Help](#) > [RoboPlus](#) > [RoboPlus Task](#) > [Programming](#) > [Type of Commands](#) > Wait While

ROBOTIS Tech Support v1.00

Wait While

Last updated 2010.1.18 (v1.0 Eng)

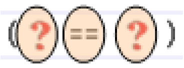
This command is used to pause execution while the condition is true.

It is the equivalent of the "while" function in C language.

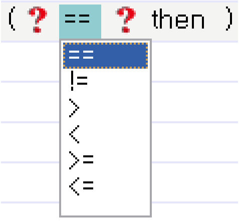
What is a conditional clause?

Conditional clause is a feature to perform different actions depending on whether the condition evaluates to true (condition is met) or false (condition is not met).

Conditional clause is composed of the following 3 parts: parameter 1, relational operator, and parameter 2.



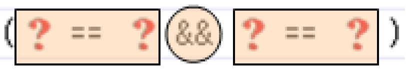
These are 6 types of relational operators.



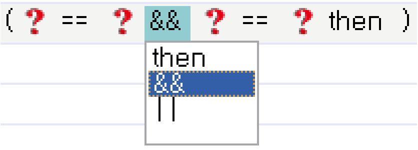
- **==** : True if the two parameters are equal.
- **!=** : True if the two parameters are not equal.
- **>=** : True if parameter 1 is greater than or equal to parameter 2.
- **>** : True if parameter 1 is greater than parameter 2.
- **<=** : True parameter 1 is less than or equal to parameter 2.
- **<** : True if parameter 1 is less than parameter 2.

Conditional clause can be combined into a complex conditional clause using conditional operators.

A complex conditional clause is composed of the following 3 parts: conditional clause 1, conditional operator, conditional clause 2.



There are 3 types of conditional operators.



- **then** : Does not link any clauses.
- **AND(&&)**: True if both conditional clauses are true.
- **OR(||)**: True if one of the conditional clauses is true.

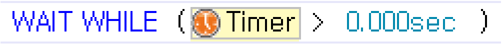
There is no limit to how many conditional clauses can be combined into one complex conditional clause. Each conditional clause is evaluated in order, and the final value will be either "true" or "false."

Usage

- Set the appropriate conditions without the use of blocks.

Example

- The program will wait for the timer.



```
WAIT WHILE ( Timer > 0.000sec )
```

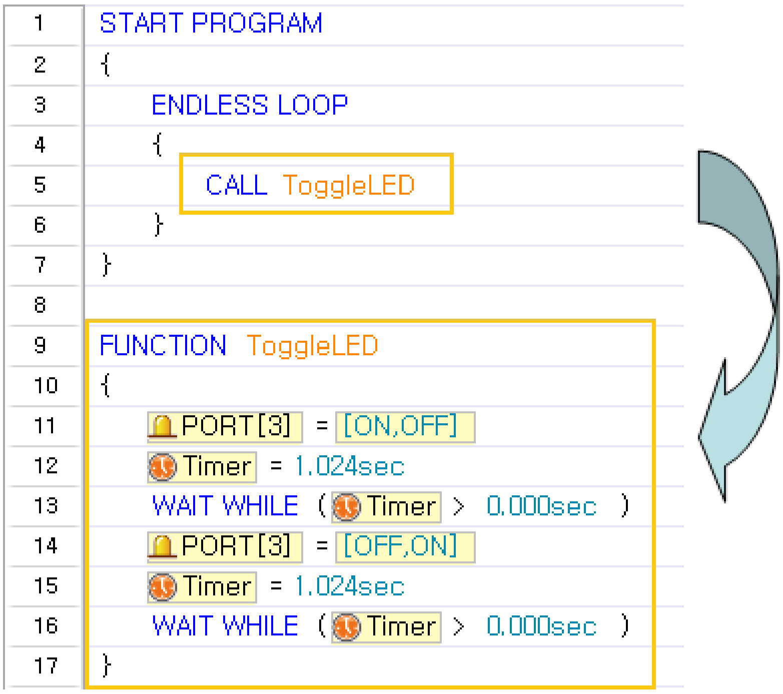
Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Make/Call Function

ROBOTIS Tech Support v1.00

Make/Call Function

Last updated 2010. 1.18 (v1.0 Eng)

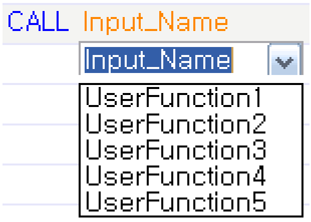
If the same code need to be repeated multiple times, or if the code needs to be distinguished according to its role, the code can be made into a function to be called whenever necessary. This is similar to the concept of a function in C language. The only difference is that there are no return values and input parameters. Used properly, functions can be used to easily determine the flow of the program and to reduce the lines of code. Functions are executed by calling them. After the called function ends, execution resumes from the point of the call.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        CALL  ToggleLED
    }
}

FUNCTION  ToggleLED
{
    PORT[3]  =  [ON,OFF]
    Timer  =  1.024sec
    WAIT WHILE  ( Timer  >  0.000sec  )
    PORT[3]  =  [OFF,ON]
    Timer  =  1.024sec
    WAIT WHILE  ( Timer  >  0.000sec  )
}
```

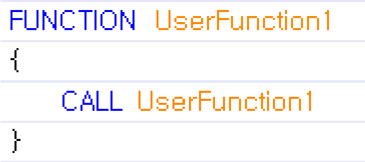
Usage

- The following rules apply when making a function:
  - There cannot be duplicate function names.
  - A function must exist outside another function or program body.
  - Spaces and special characters(!, @, #, $ etc.) are not allowed in function names.
  - Function names cannot start with numbers.
- While inputting function names, press Esc to cancel.
- Otherwise, press Enter to save.
- While selecting the function to call, press ESC to cancel.
- Otherwise, click the appropriate function or press Enter while the function is highlighted to save.



```
CALL  Input_Name
```

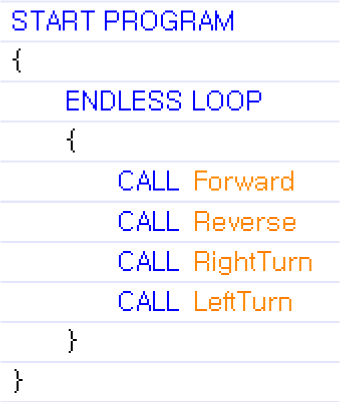
- A function cannot call itself.



```
FUNCTION  UserFunction1
{
      CALL  UserFunction1
}
```

Example

- The program will continuously call the functions to move forward, backward, right and left.



```
START PROGRAM
{
      ENDLESS LOOP
      {
            CALL  Forward
            CALL  Reverse
            CALL  RightTurn
            CALL  LeftTurn
      }
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Exit Function

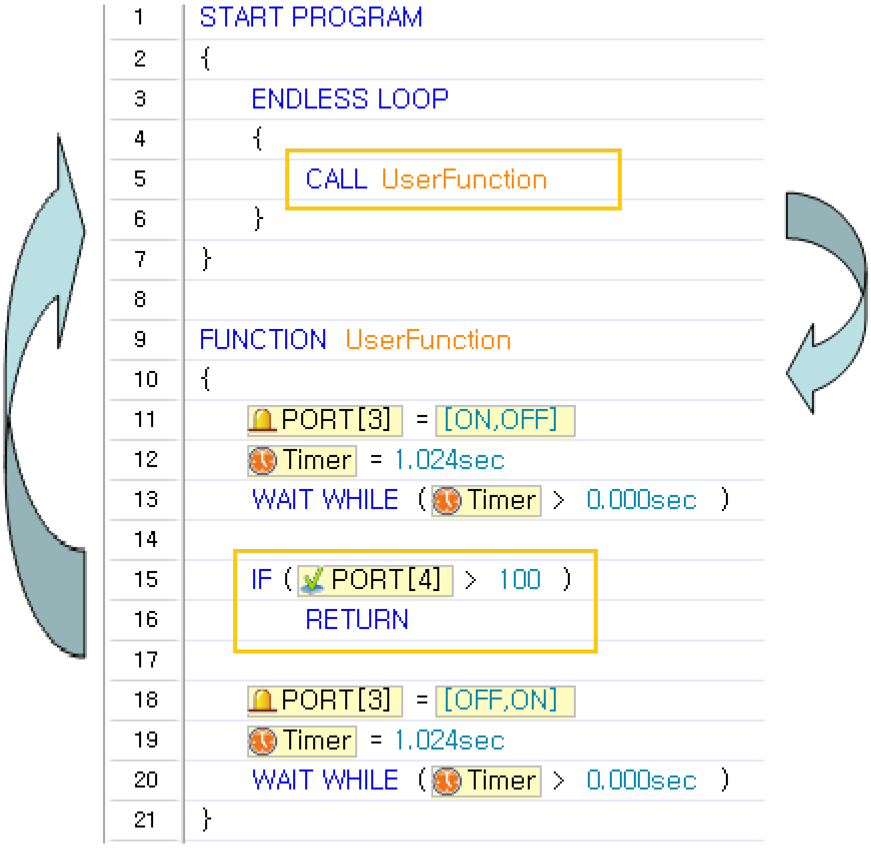
ROBOTIS Tech Support v1.00

Exit Function

Last updated 2010. 1.18 (v1.0 Eng)

When this command is executed, the function will end, even if not every line in the function has been executed.

It is the equivalent of the "return" statement in C language.



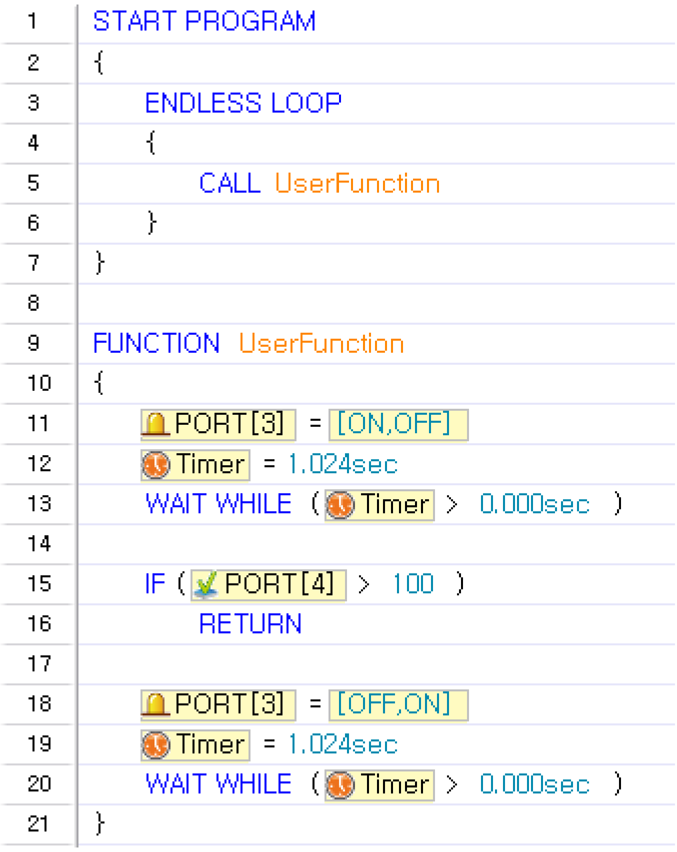
```
START PROGRAM
{
    ENDLESS LOOP
    {
        CALL  UserFunction
    }
}

FUNCTION  UserFunction
{
    PORT[3]  =  [ON,OFF]
    Timer  =  1.024sec
    WAIT WHILE  ( Timer  >  0.000sec  )

    IF ( PORT[4]  >  100  )
        RETURN

    PORT[3]  =  [OFF,ON]
    Timer  =  1.024sec
    WAIT WHILE  ( Timer  >  0.000sec  )
}
```

Usage

- Can be used only within a general function or a callback function.

Example

- In the following example, UserFunction is called repeatedly. Because of the **Return** command in UserFunction, the last 3 lines will never be executed.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        CALL UserFunction
    }
}

FUNCTION UserFunction
{
    PORT[3] = [ON,OFF]
    Timer = 1.024sec
    WAIT WHILE ( Timer > 0.000sec )

    IF ( PORT[4] > 100 )
        RETURN

    PORT[3] = [OFF,ON]
    Timer = 1.024sec
    WAIT WHILE ( Timer > 0.000sec )
}
```

Callback Function

Last updated 2010.1.18 (v1.0 Eng)

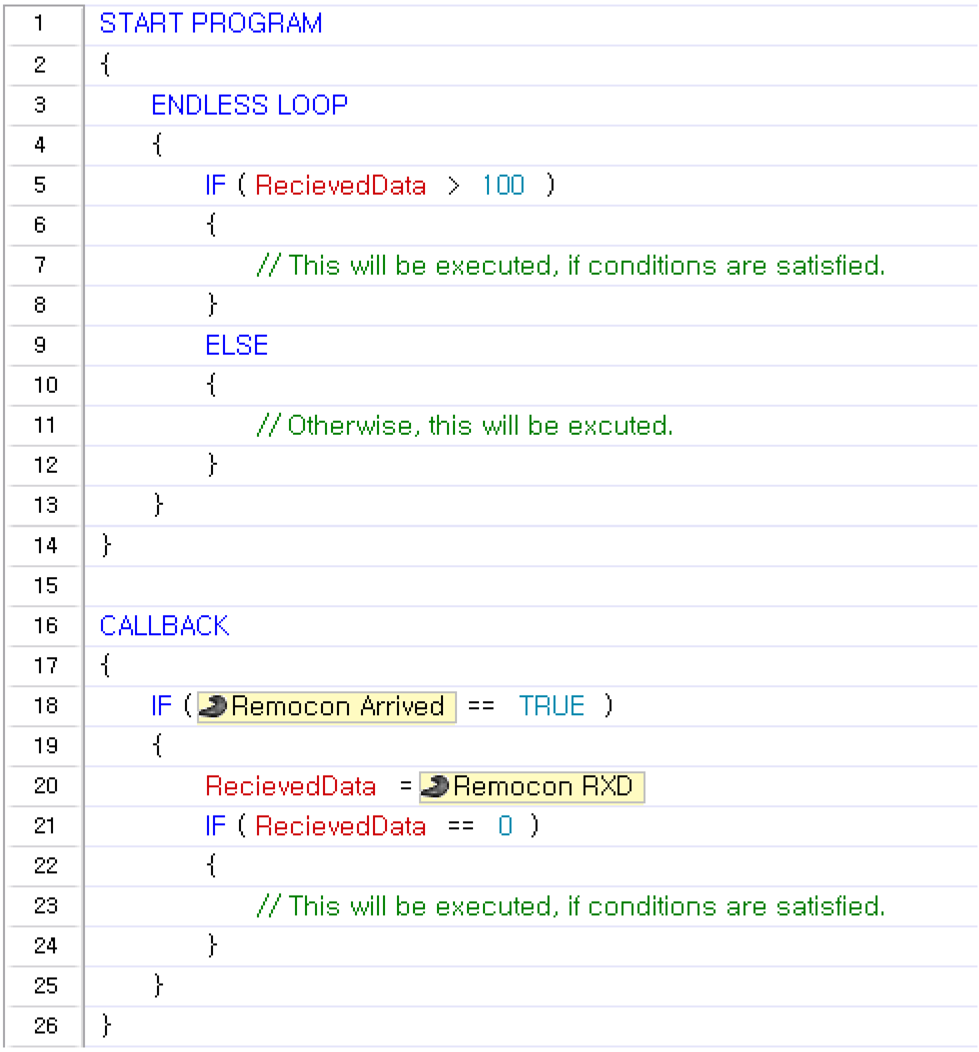
Callback function is a function that runs independently of the main program routine and is automatically executed at fixed intervals. Therefore, a callback function cannot include code that requires much time. Use of loops, variables, and function calls are limited.

Usage

- The callback function cannot exist inside another function or program body.
- There can be only one callback function.
- A callback function does not have a name and cannot be called.

Example

- This example shows how periodically receive wireless data and to save it in a variable.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        IF ( RecievedData > 100 )
        {
            // This will be executed, if conditions are satisfied.
        }
        ELSE
        {
            // Otherwise, this will be excuted.
        }
    }
}

CALLBACK
{
    IF ( Remocon Arrived == TRUE )
    {
        RecievedData = Remocon RXD
        IF ( RecievedData == 0 )
        {
            // This will be executed, if conditions are satisfied.
        }
    }
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > End Program

ROBOTIS Tech Support v1.00

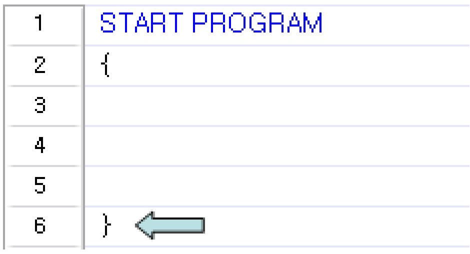
End Program

Last updated 2010. 1.15 (v1.0 Eng)

If this command is called during program execution, the program exits immediately.

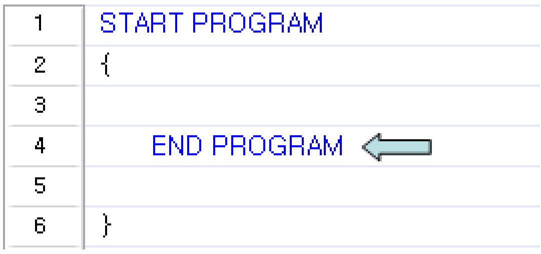
There are 2 ways to end a programme.

- When the end of "Start Program" is reached (Natural Close)



```
START PROGRAM
{


}
```

- When "End Program" is called (Forced Exit)



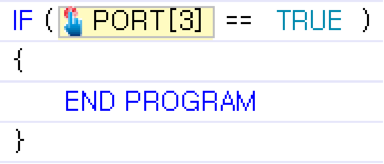
```
START PROGRAM
{

    END PROGRAM

}
```

Usage

- Call the command at the point in the program where you want it to end.

Example

- In this sample code, the program will end when the touch connected to Port3 is pressed.



```
IF ( PORT[3] == TRUE )
{
    END PROGRAM
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Start/End of Section

ROBOTIS Tech Support v1.00

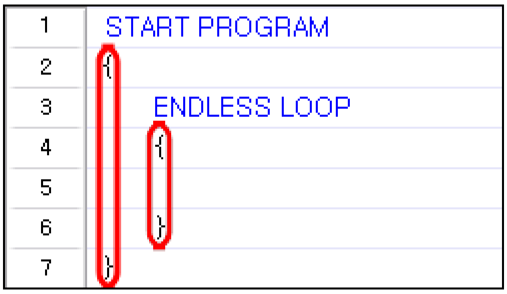
Start/End of Block

Last updated 2010.1.15 (v1.0 Eng)

A block (identified by "{" and "}") is a group of commands. All commands in a block have the same scope. The concept of a block is the same as in the C language.

Usage

- Each block has an opening bracket ({) and a closing bracket (}).
  RoboPlus Task performs automatic indentation to show whether the brackets have been paired properly. If there are missing brackets, they must be added before the program can run indentation was not arranged properly, you have to revise them by yourself,



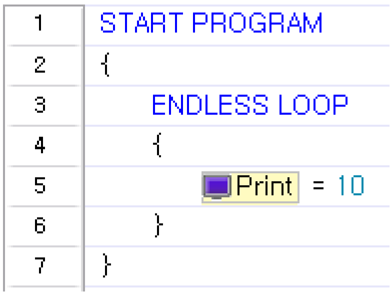
```
START PROGRAM
{
    ENDLESS LOOP
    {

    }
}
```

- Each block must be "owned" by a command. That is, blocks cannot be used independent of commands.
  The following are the most commonly used commands that are followed by a block.

  1 Start Program
  2 If / Else if / Else
  3 Endless Loop
  4 Loop For
  5 Loop While
  6 Callback Function
  7 Function

Example

- The "Start Program" and "Endless Loop" commands must be followed by blocks, as shown in the example below.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        Print = 10
    }
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Comment or Notes

ROBOTIS Tech Support v1.00

Comment or Notes

Last updated 2010. 1. 15 (v1.0 Eng)

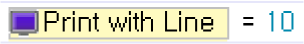
This command is used to insert a comment or a note in the program code. Comments are helpful when interpreting or reviewing the code later. They are mostly used to mark easily forgotten parts or to emphasize important information. Comments and notes do not affect the program in any way. Like in C, comments can be made with two slashes(//). Comments blocks (/*', '*/) are not supported.

Usage

- Insert the commend where you would like to write a comment or note.
- When "//" is added, double click or press Enter to write in the comment or note.
- Pressing ESC while writing will erase what has been written and return the line to its previous state.
- When finished, press Enter.

Example

- This code will print "10" on the screen. The comment explains what the line below it will do.



```
Print with Line = 10
```

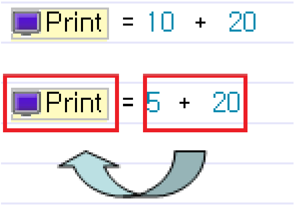
Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Calculate

ROBOTIS Tech Support v1.00

Calculate

Last updated 2010. 1.15 (v1.0 Eng)

"Calculate" is a command to perform an arithmetic operation on two numbers.

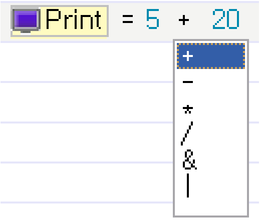


"Calculate" can perform the following operations:

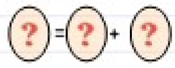
- Basic Operations (supports negative numbers)
  - **Addition(+)**: Add two values.
  - **Subtraction(-)**: Subtract the second number from the first number.
  - **Multiplication(*)**: Multiply two numbers.
  - **Division(/)**: Divide the first number by the second number (Remainders are discarded.)
- Bit Operations (Means 2 decimal operation.)
  - **AND(&)**: Perform a logical AND operation.
  - **OR(|)**: Perform a logical OR operation.

Usage

- You can choose an operator by double clicking a mouse or by pressing Enter.

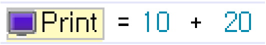


- Choose the appropriate 3 parameters (result, operand1, operand2) necessary for the command.



Example

- This example shows how to add 10 and 20 and to display the result on the screen.



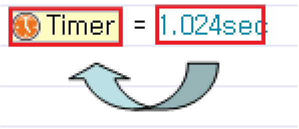
Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Load

ROBOTIS Tech Support v1.00

Load

Last updated 2010.1.15 (v1.0 Eng)

"'Load" is defined as "to place into an appropriate device." In RoboPlus Task, "Load" places a value in a device.



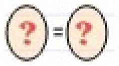
Timer = 1.024sec

"Load" is used to mean the following:

- Execute a device's function.
- Move a value.

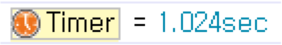
Usage

- Choose the appropriate 2 parameters (destination, source) necessary for the command.



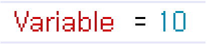
Example

- To execute a device (Set the timer to 1.024 seconds.)



Timer = 1.024sec

- To set a value (Insert 10 into the variable)



Variable = 10

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Label/Jump

ROBOTIS Tech Support v1.00

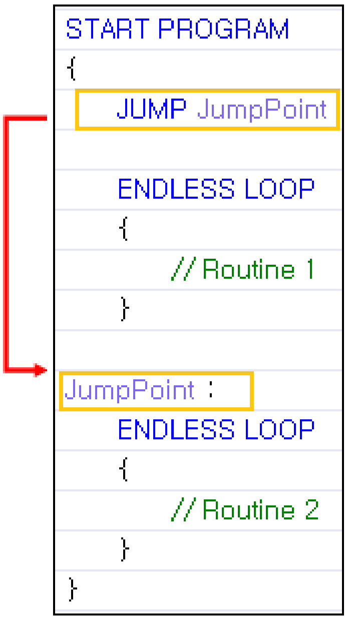
Label/Jump

Last updated 2010.1.15 (v1.0 Eng)

"Label" and "Jump" are used to branch a program. Branching is used to change the order commands in a program are executed.

"Jump" branches the program and "Label" designates where to branch to.

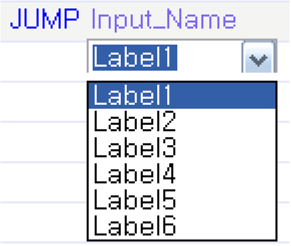
It is the same function as "Label" and "Goto'" in the C language.



```
START PROGRAM
{
    JUMP JumpPoint

    ENDLESS LOOP
    {
        // Routine 1
    }

JumpPoint :
    ENDLESS LOOP
    {
        // Routine 2
    }
}
```

Usage

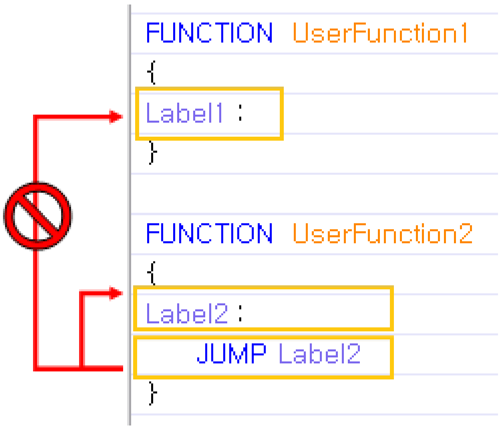
- Label names must abide by the following rules:
  - There cannot be duplicate label names.
  - Label must exist within a program or function body.
  - A jump to a label in another function is not possible.
  - Spaces and Special characters(!, @, #, $, etc.) are not allowed in label names.
  - Labels cannot start with a number.
- While inputting the label name, Press Esc to cancel.
- Otherwise, press Enter to save
- While selecting the label to jump to, Press Esc to cancel.
- Otherwise, click the appropriate label or press Enter while the label is highlighted to save.



```
JUMP Input_Name
```

- A jump can only be made to an existing label.

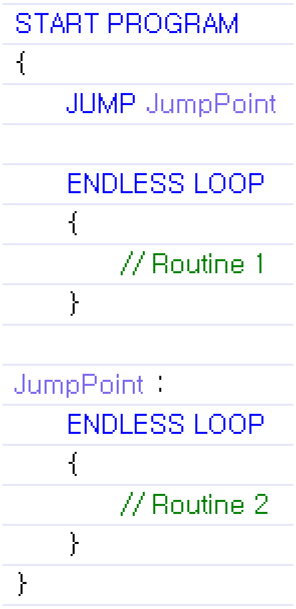
- The label must be in the same function block as the jump command.



```
FUNCTION  UserFunction1
{
Label1 :
}

FUNCTION  UserFunction2
{
Label2 :
        JUMP Label2
}
```

Example

- In this sample code, the program jumps to "JumpPoint" as soon as it starts and executes"'Routine 2."



```
START PROGRAM
{
    JUMP JumpPoint

    ENDLESS LOOP
    {
        // Routine 1
    }

JumpPoint :
    ENDLESS LOOP
    {
        // Routine 2
    }
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > If/Else If/Else

ROBOTIS Tech Support v1.00

If/Else if/Else

Last updated 2010.1.18 (v1.0 Eng)

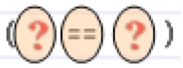
These commands will branch the flow of the program depending on whether the condition is true or false.

- **If :** Execute if f the clause is true. This is the equivalent of the "if" statement in C language.
- **Else If**: Execute if the clause is true and previous clause ("if" or "else if" clause) is false. This is the equivalent of the "else if" statement in C language.
- **Else**: Execute if none of the conditions are true. This is the equivalent of the "else" statement in C language.

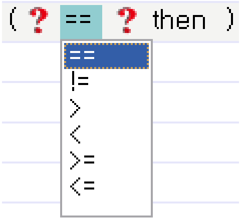
What's a Conditional Clause?

Conditional clause is a feature to perform different actions depending on whether the condition evaluates to true (condition is met) or false (condition is not met).

Conditional clause is composed of the following 3 parts: parameter 1, relational operator, and parameter 2.



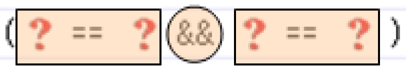
These are 6 types of relational operators.



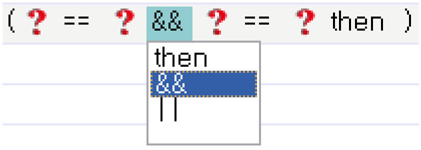
- **==** : True if the two parameters are equal.
- **!=** : True if the two parameters are not equal.
- **>=** : True if parameter 1 is greater than or equal to parameter 2.
- **>** : True if parameter 1 is greater than parameter 2.
- **<=** : True parameter 1 is less than or equal to parameter 2.
- **<** : True if parameter 1 is less than parameter 2.

Conditional clause can be combined into a complex conditional clause using conditional operators.

A complex conditional clause is composed of the following 3 parts: conditional clause 1, conditional operator, conditional clause 2.



There are 3 types of conditional operators.

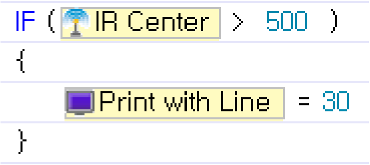


- **then** : Does not link any clauses.
- **AND(&&)**: True if both conditional clauses are true.
- **OR(||)**: True if one of the conditional clauses is true.

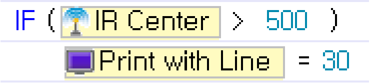
There is no limit to how many conditional clauses can be combined into one complex conditional clause. Each conditional clause is evaluated in order, and the final value will be either "true" or "false."

Usage

- An 'IF' command must always precede an "Else if" or an "Else" command.
- A block, designated by brackets, needs to follow each clause. (However, if the block consists of only one line, the block need not be enclosed with brackets.)



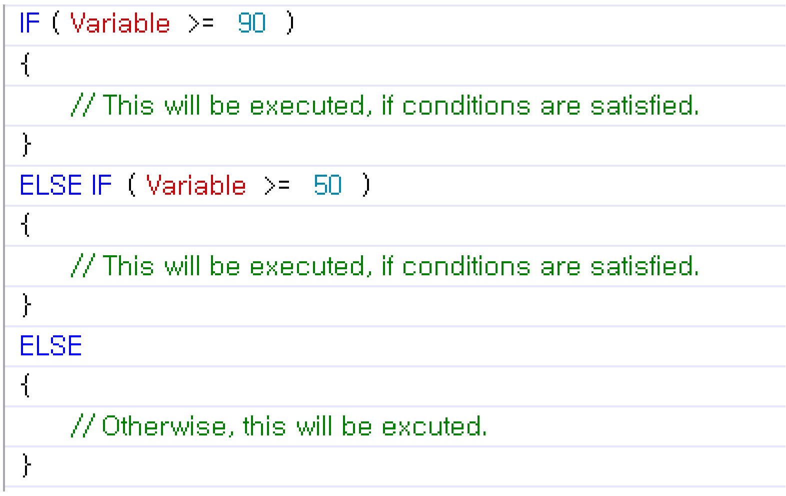
```
IF ( IR Center > 500 )
{
      Print with Line = 30
}
```



```
IF ( IR Center > 500 )
      Print with Line = 30
```

Example

- The examples below shows how to program the following conditions.
    - When the variable is greater than or equal to 90.
    - When the variable is greater than or equal to 50 and less than 90.
    - Other cases



```
IF ( Variable >= 90 )
{
      // This will be executed, if conditions are satisfied.
}
ELSE IF ( Variable >= 50 )
{
      // This will be executed, if conditions are satisfied.
}
ELSE
{
      // Otherwise, this will be excuted.
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Type of Commands > Endless Loop

ROBOTIS Tech Support v1.00

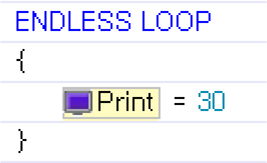
Endless Loop

Last updated 2010.1.18 (v1.0 Eng)

This command is used to repeat the command lines in the block without end.

Usage

- A block is always required. (However, if the block consists of only one, the block need not be enclosed with brackets.)

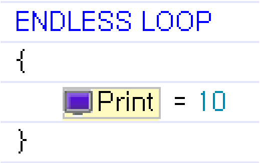


```
ENDLESS LOOP
{
        Print  = 30
}
```

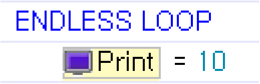
- Use the "Break Loop" command to exit the loop.

Example

- Continuously prints "10" on the Program Output Monitor.



```
ENDLESS LOOP
{
        Print  = 10
}
```



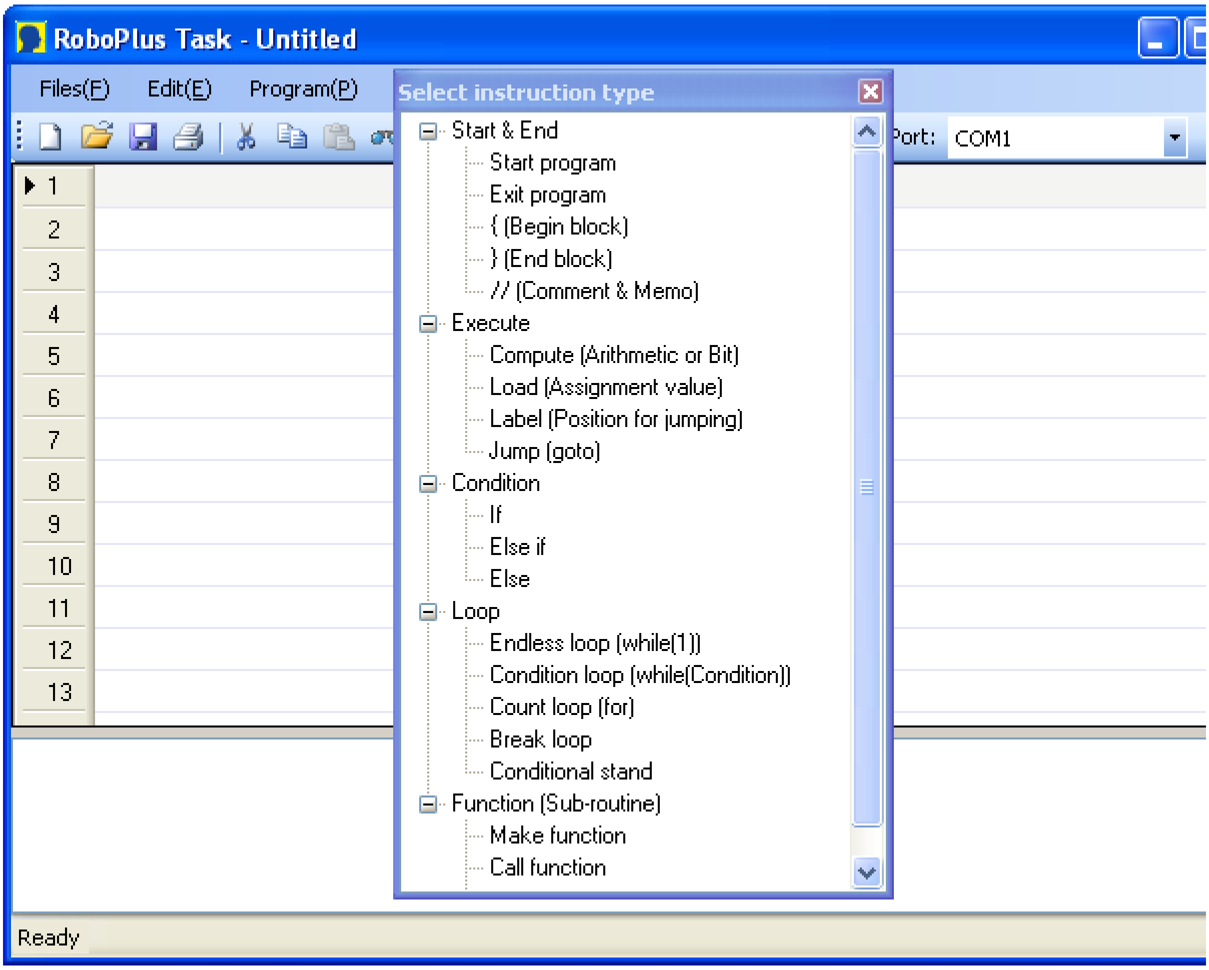
```
ENDLESS LOOP
        Print  = 10
```

ROBOTIS Tech Support v1.00

Select Command Line

Last updated 2010.1.15 (v1.0 Eng)

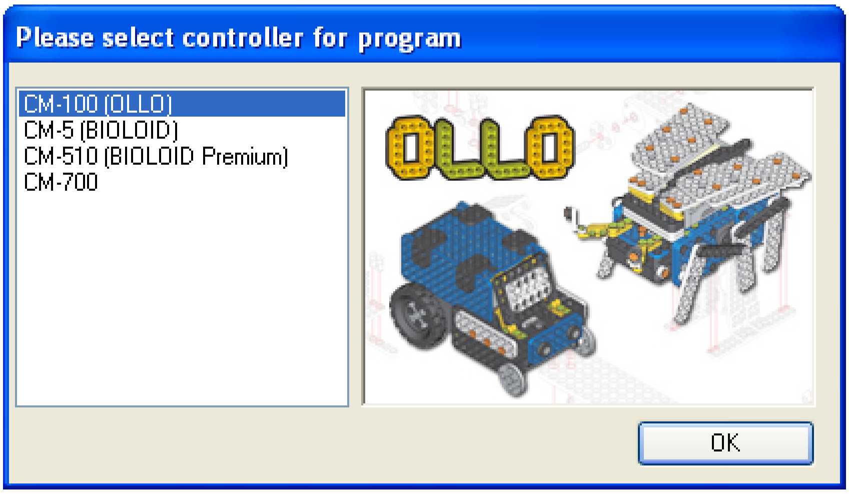
- Double click on a blank line or click on the line and press enter. Choose a command from the list of commands supported by the selected controller.



Select instruction type

- Start & End
  - Start program
  - Exit program
  - { (Begin block)
  - } (End block)
  - // (Comment & Memo)
- Execute
  - Compute (Arithmetic or Bit)
  - Load (Assignment value)
  - Label (Position for jumping)
  - Jump (goto)
- Condition
  - If
  - Else if
  - Else
- Loop
  - Endless loop (while(1))
  - Condition loop (while(Condition))
  - Count loop (for)
  - Break loop
  - Conditional stand
- Function (Sub-routine)
  - Make function
  - Call function

- If the controller has not been selected yet, the program will ask you to choose the type of controller that will be used by the current program.



Please select controller for program

- CM-100 (OLLO)
- CM-5 (BIOLOID)
- CM-510 (BIOLOID Premium)
- CM-700

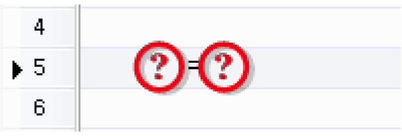
Home > Software Help > RoboPlus > RoboPlus Task > Getting It Started > Select Parameter

ROBOTIS Tech Support v1.00

Select Parameter

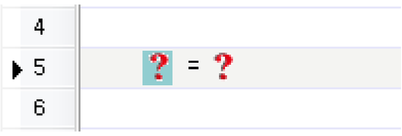
Last updated 2010.1.15 (v1.0 Eng)

"Parameter" refers to the device required to execute commands. A question mark(?) indicates that a parameter has not been set.

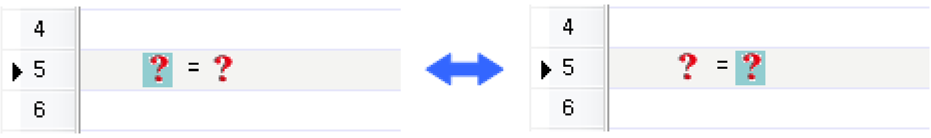


After selecting a command, you must designate a parameter to complete the command line.

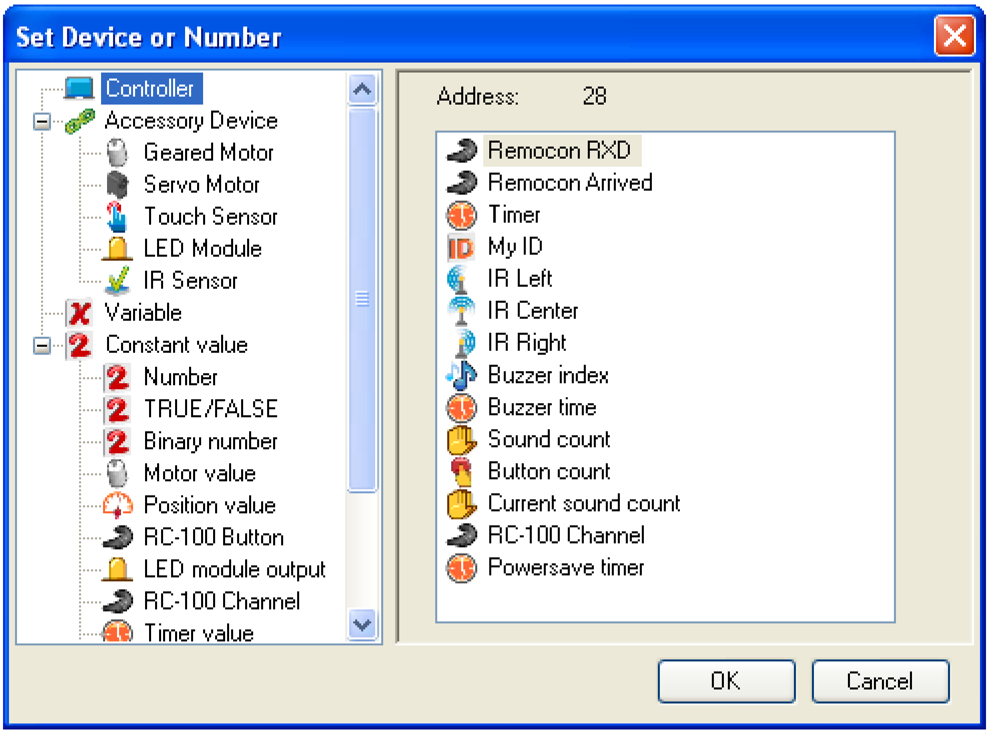
1. Go to 'edit mode' by double clicking the mouse or pressing the enter key.



2. Choose the parameter to create by pressing left/right arrow keys or by clicking on the question mark.



3. Press enter or double click to see the parameter selection window.



Set Device or Number

Address: 28

4. Choose the appropriate parameter. It is very important to learn how to use each parameter.

ROBOTIS Tech Support v1.00

Download

Last updated 2010.1.15 (v1.0 Eng)

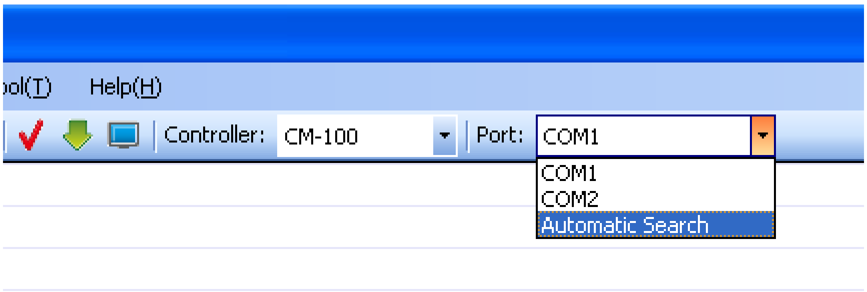
Download the task codes into your controller. You only have to download once, as the task codes will be saved inside of controller.

1. **The controller should be connected to the PC.**

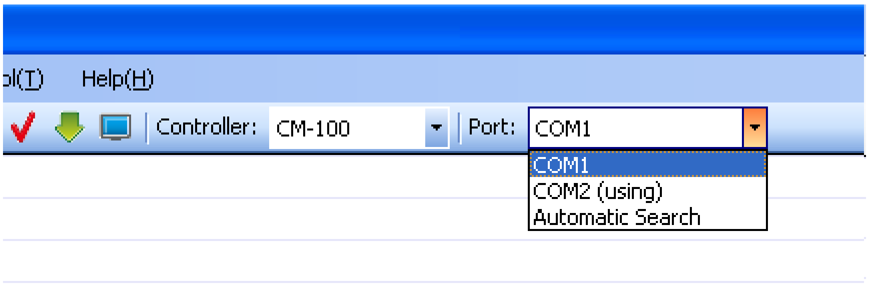
   To download the task code, the controller must be connected to the PC. (Please refer to controller information for information on how to connect the controller to the PC.)

2. **Select the communication port to use.**

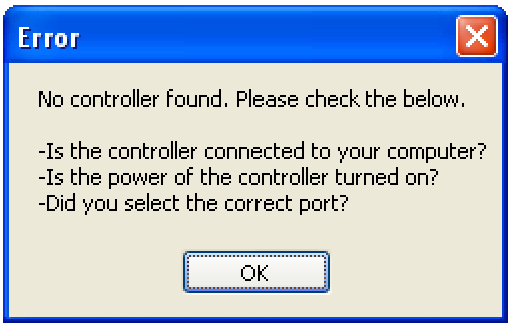
   Use the "Automatic Search" function to easily select the appropriate port.



If the chosen communication port is being used by another program, you must first find and stop the program.



If RoboPlus Task is unable to find a controller, the following error message will be shown.



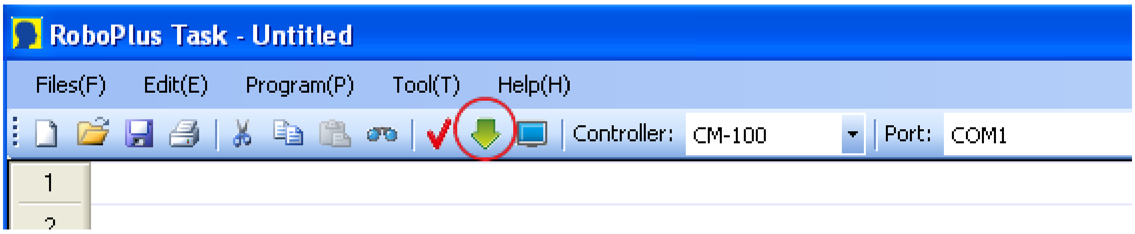
Error

No controller found. Please check the below.

-Is the controller connected to your computer?
-Is the power of the controller turned on?
-Did you select the correct port?

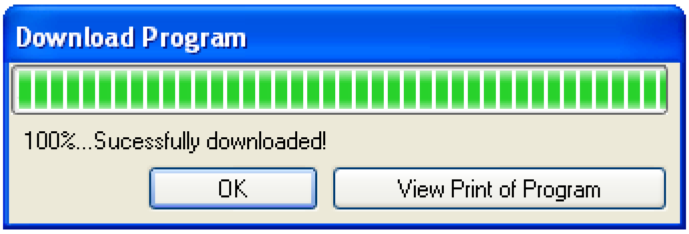
- Check if the controller is connected to the PC. (See controller information on how to connect the controller.)
- Check if the controller is turned on.
- Check if the correct communication port was chosen.

3. **Select the download menu.**



If the program has an error, you must find the error and correct it. (See "rule check error messages")

4. **Download the program.**



100%...Sucessfully downloaded!

If the downloading fails, it will automatically try again from the beginning.

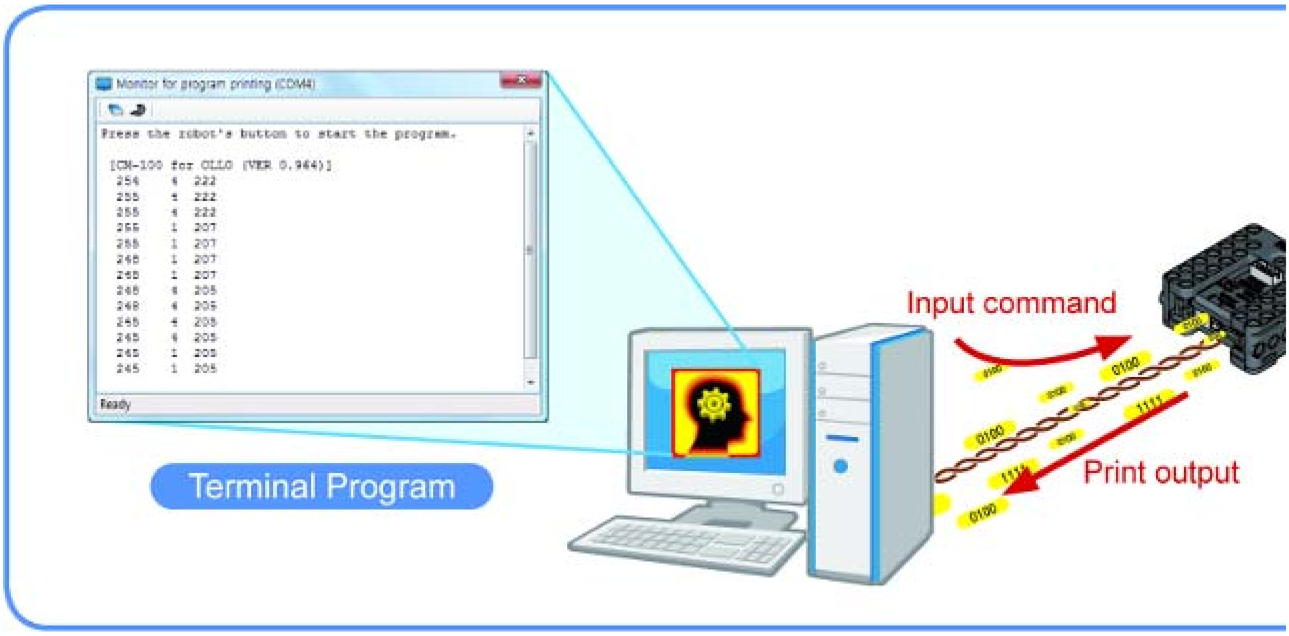
5. **Execute the task code -> Your robot will move.**

Turn on the controller and execute the downloaded task code. (Please refer to controller information to learn how to execute the task code.)

Print Program Output

Last updated 2010.1.15 (v 1.0 Eng)

Controllers usually do not have display devices the way a PC has a monitor, so it is hard to keep track of what goes on in a controller. Therefore, a "terminal" is used to "borrow" the PC's monitor.



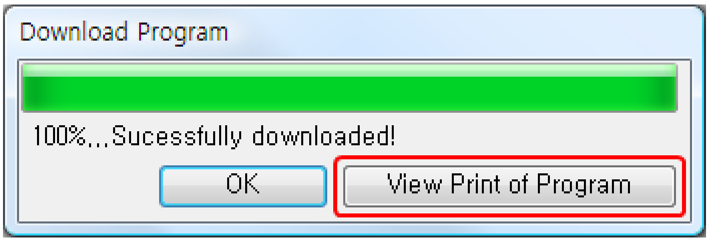
```
Press the robot's button to start the program.

 [CM-100 for OLLO (VER 0.944)]
  254    4  222
  255    5  222
  255    4  222
  256    1  207
  255    1  207
  248    1  207
  248    1  207
  248    4  205
  248    4  205
  245    4  205
  245    4  205
  245    1  205
  245    1  205
```

RoboPlus Task includes a terminal program to display the status of the controller.

Open the Program Output Monitor

To see the output of the program, you must open the **Program Output Monitor** BEFORE executing the program. There are three ways to open the Program Output Monitor.

1. Click the '**View Print of Program**' on the **Download Program** window**.**

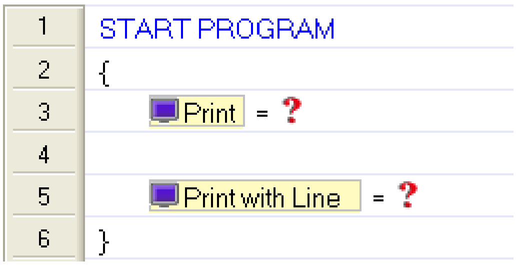


100%...Sucessfully downloaded!

2. Click the **"View Print of Program"** button in the **toolbar.**
3. Press **F5** or Click on **"View Print of Program (V)"** menu under **Program (P)**.

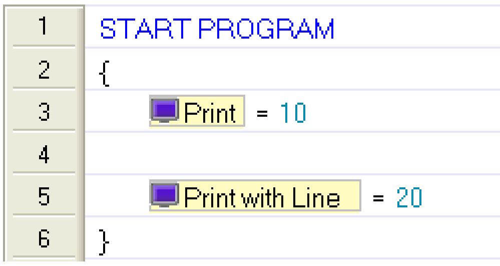
Print / Print with carriage return

- Use the "Print" parameter in your task code to see desired values on the screen.

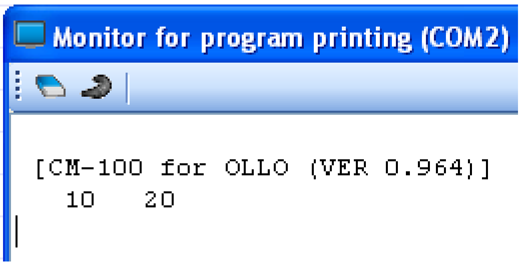


```
START PROGRAM
{
    Print = ?

    Print with Line = ?
}
```

- Print : Print the value, then move the cursor to next line.

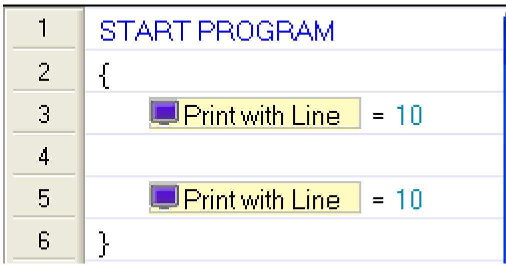


```
START PROGRAM
{
    Print = 10

    Print with Line = 20
}
```

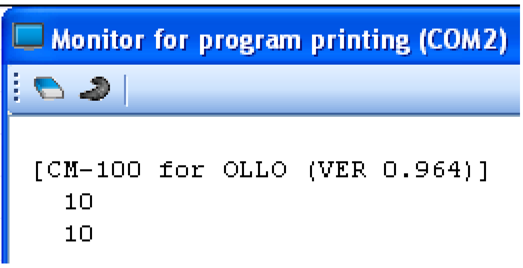


```
[CM-100 for OLLO (VER 0.964)]
  10    20
```

- Print with Line: Print out the value, and move the cursor to next line.



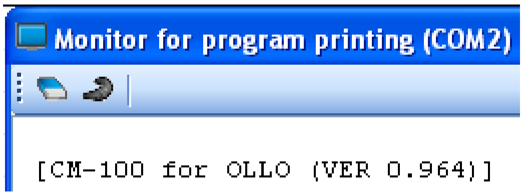
```
START PROGRAM
{
    Print with Line = 10

    Print with Line = 10
}
```



```
[CM-100 for OLLO (VER 0.964)]
  10
  10
```

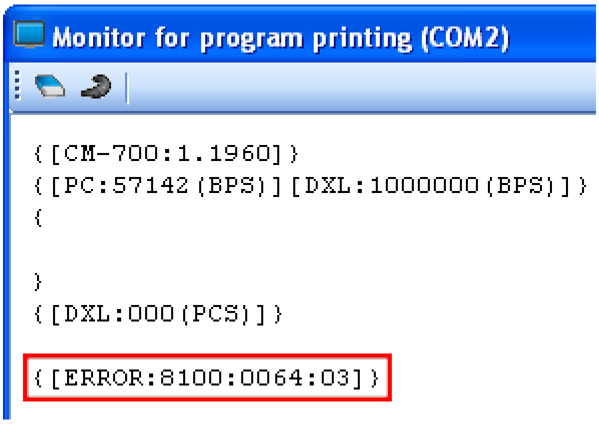
What can be printed on the screen

- Values from the controller
  - A welcome screen is shown when the program starts



```
[CM-100 for OLLO (VER 0.964)]
```

  - Error messages during program operation (See types of error messages)

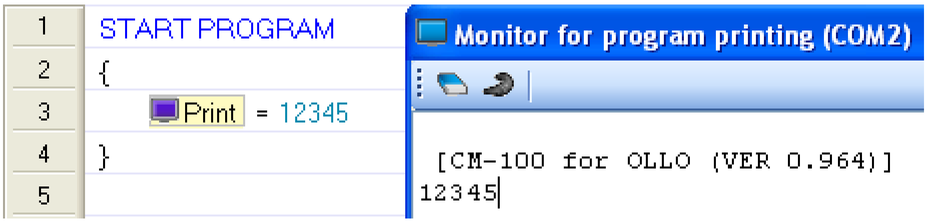


```
{[CM-700:1.1960]}
{[PC:57142(BPS)][DXL:1000000(BPS)]}
{

}
{[DXL:000(PCS)]}

{[ERROR:8100:0064:03]}
```

- Values from task code

  All decimal numbers between -32768 and +32765 can be displayed. (Characters cannot be printed.)

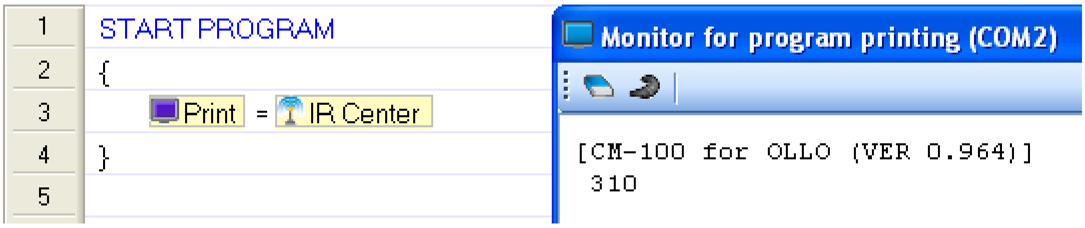
  - To print numbers



```
START PROGRAM
{
    Print = 12345
}
```

```
 [CM-100 for OLLO (VER 0.964)]
12345
```

  - To print values from sensors

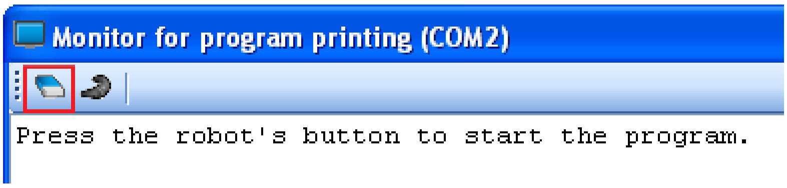


```
START PROGRAM
{
    Print = IR Center
}
```

```
[CM-100 for OLLO (VER 0.964)]
 310
```

Clear Screen

- You may erase everything on the screen.



```
Press the robot's button to start the program.
```

Home > Software Help > RoboPlus > RoboPlus Task > Getting It Started > Virtual Robot Control

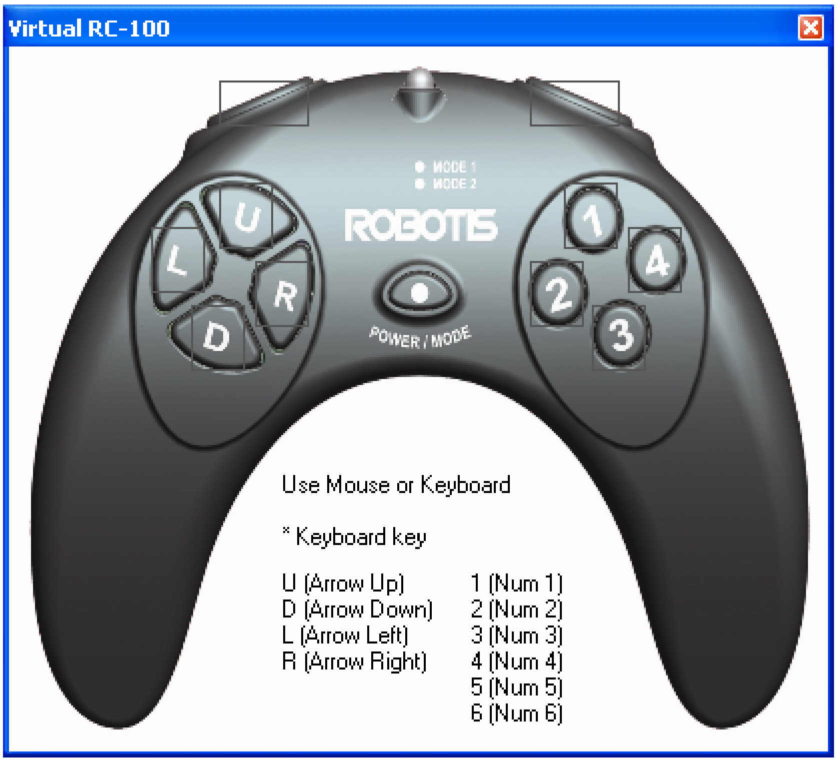
ROBOTIS Tech Support v1.00

Virtual Robot Control

Last updated 2010.1.15 (v1.0 Eng)

RoboPlus Task supports virtual robot control, which makes controlling of robot possible without a remote controller devices like RC-100.

Click the appropriate button with the mouse or press the appropriate key.



Use Mouse or Keyboard

* Keyboard key

U (Arrow Up)
D (Arrow Down)
L (Arrow Left)
R (Arrow Right)

1 (Num 1)
2 (Num 2)
3 (Num 3)
4 (Num 4)
5 (Num 5)
6 (Num 6)

Please refer to the graph below to remote control with keyboard.

Real RC-100 Key	Virtual RC-100 Key on the Keyboard
U	Arrow Key(↑)
D	Arrow Key(↓)
L	Arrow Key(←)
R	Arrow Key(→)
1	Number Key(1)
2	Number Key(2)
3	Number Key(3)
4	Number Key(4)
5	Number Key(5)
6	NumberKey(6)

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Parameter > Controller > Remocon TXD

ROBOTIS Tech Support v1.00

Remocon TXD

Last updated 2010.1.18 (v1.0 Eng)

This parameter is used to transmit data via a wireless communication module (IR, zigbee module).

Compatible Controller

- CM-100
- CM-5
- CM-510
- CM-700

Characteristics

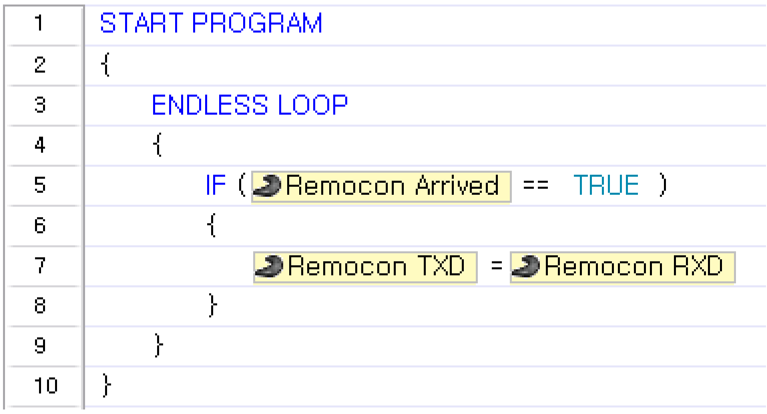
- The data to send must be a number between 0 and value in between 0 and 65535.
- When the "Remocon TXD" parameter is set, the data is immediately sent wirelessly.

Example

- In the example below, the program waits for data, and when the data arrives, the received data is transmitted wirelessly.

Tip

- It is commonly used to send a response to the control program on a PC connected using ZIG2Serial.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        IF ( Remocon Arrived  ==  TRUE  )
        {
            Remocon TXD  =  Remocon RXD
        }
    }
}
```

Remocon RDX

This parameter is used to read the received data received via the wireless communication module (IR, zigbee module).

Compatible Controller

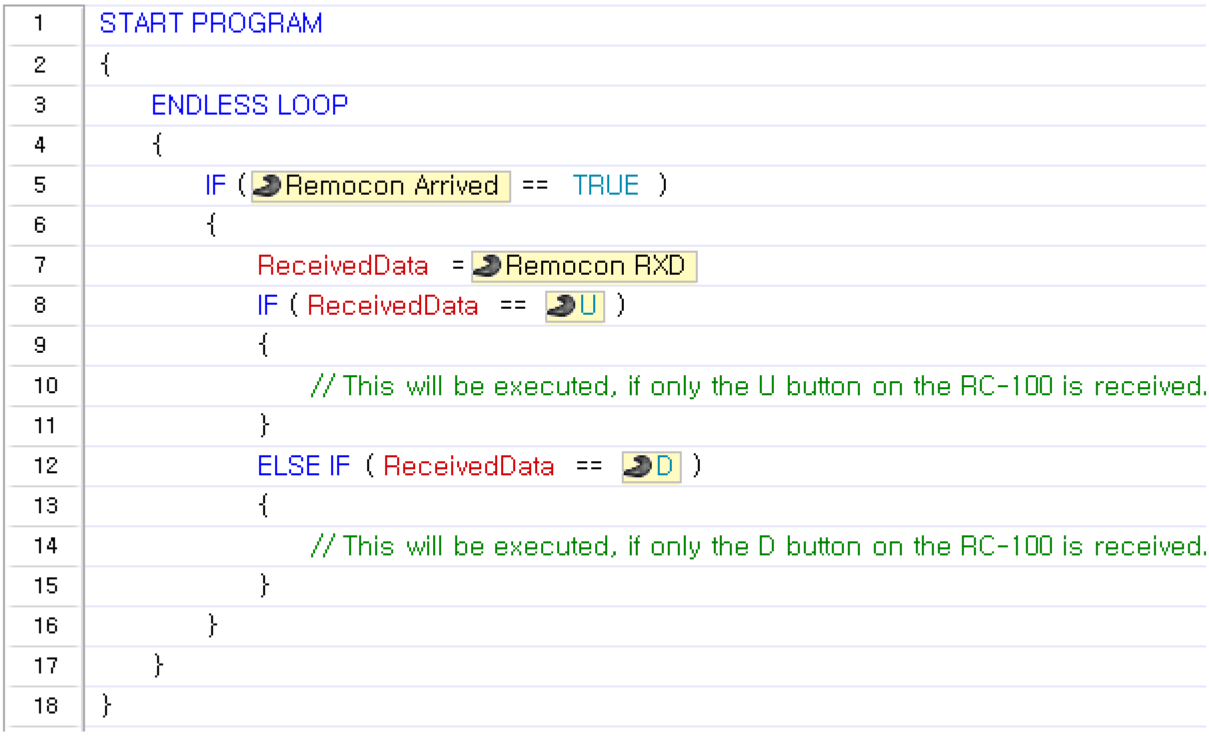
- CM-100
- CM-5
- CM-510
- CM-700

Characteristic

- The data is a number between 0 and 65535.
- The "Remocon Arrived" parameter can be used check for new data.
- Because there is an input buffer, a maximum of 2 data value using the receiving buffer.
- If the input buffer is filled with 2 data values when READ is executed, the first data will be read and removed from the buffer. If there is only 1 data value in the input buffer when READ is executed, the latest data will be retrieved.

Example

- The code below shows how to control movement direction using the RC-100.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        IF ( Remocon Arrived  ==  TRUE  )
        {
            ReceivedData  = Remocon RXD
            IF ( ReceivedData  ==  U )
            {
                // This will be executed, if only the U button on the RC-100 is received.
            }
            ELSE IF ( ReceivedData  ==  D )
            {
                // This will be executed, if only the D button on the RC-100 is received.
            }
        }
    }
}
```

Remocon Arrived

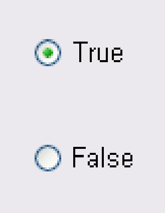
This parameter is used to check whether there are any new data received via the wireless communication module(IR, zigbee module).

Compatible Controller

- CM-100
- CM-5
- CM-510
- CM-700

Characteristics

- Is either TRUE or FALSE



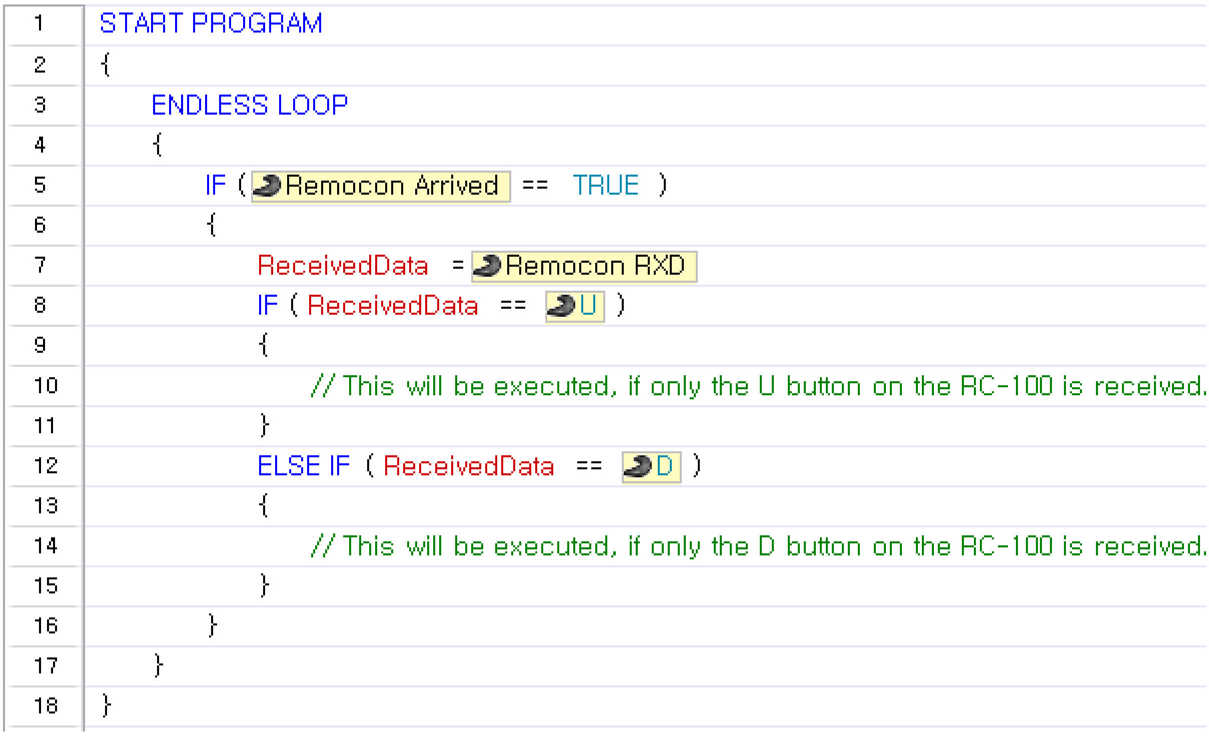
● True

○ False

- **TRUE (No. 1)** : There is new data in the input buffer.
- **FALSE, (No. 0)** : All data in the input buffer have been retrieved.

Example

- The code below shows how to control movement direction using RC-100.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        IF ( Remocon Arrived  ==  TRUE  )
        {
            ReceivedData  = Remocon RXD
            IF ( ReceivedData  ==  U )
            {
                // This will be executed, if only the U button on the RC-100 is received.
            }
            ELSE IF ( ReceivedData  ==  D )
            {
                // This will be executed, if only the D button on the RC-100 is received.
            }
        }
    }
}
```

Tip

- Normally used to check whether new data has been received to process.

ROBOTIS Tech Support v1.00

Aux LED

Last updated 2010. 1.18 (v1.0 Eng)

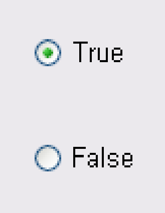
This parameter is used to read and set the controller's Aux LED status.

Compatible Controller

- CM-5
- CM-510
- CM-700

Characteristics

- Is either TRUE or FALSE.



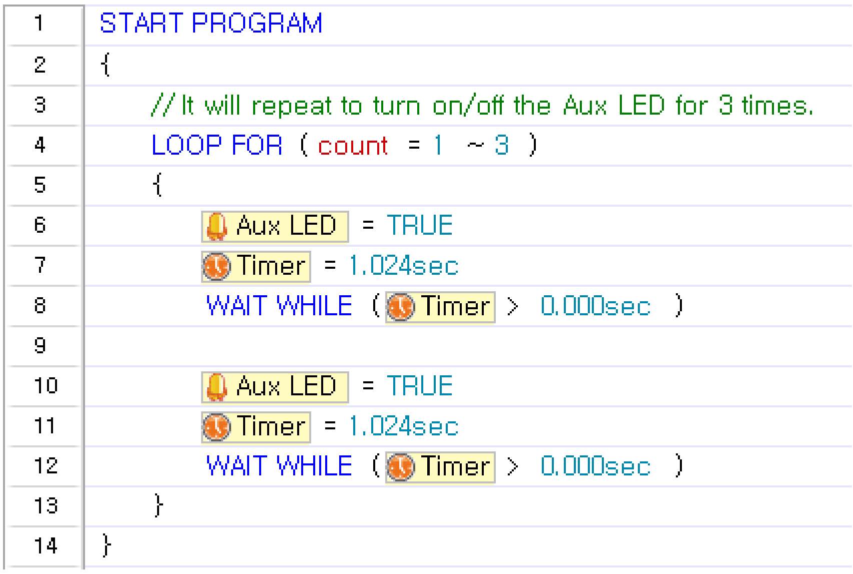
◉ True

○ False

- **TRUE (1)** : When the Aux LED parameter is set to TRUE, the LED will turn on. When the Aux LED parameter is read, a value of TRUE signifies that the LED is on.
- **FALSE (0)** : When the Aux LED parameter is set to FALSE, the LED will turn off. When the Aux LED parameter is read, a value of FALSE signifies that the LED is off.

Example

- In this example, the Aux LED is turned on and off for 1 second 3 times.



```
START PROGRAM
{
    // It will repeat to turn on/off the Aux LED for 3 times.
    LOOP FOR ( count = 1 ~ 3 )
    {
        Aux LED = TRUE
        Timer = 1.024sec
        WAIT WHILE ( Timer > 0.000sec )

        Aux LED = TRUE
        Timer = 1.024sec
        WAIT WHILE ( Timer > 0.000sec )
    }
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Parameter > Controller > Button

ROBOTIS Tech Support v1.00

Button

Last updated 2010.1.18 (v1.0 Eng)

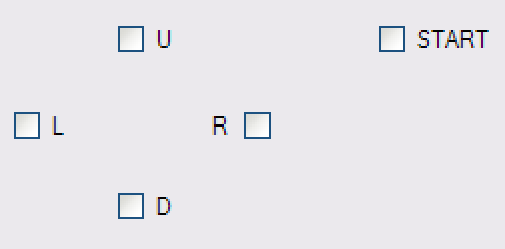
This parameter is used to read the controller's button status.

Compatible Controller

- CM-5
- CM-510

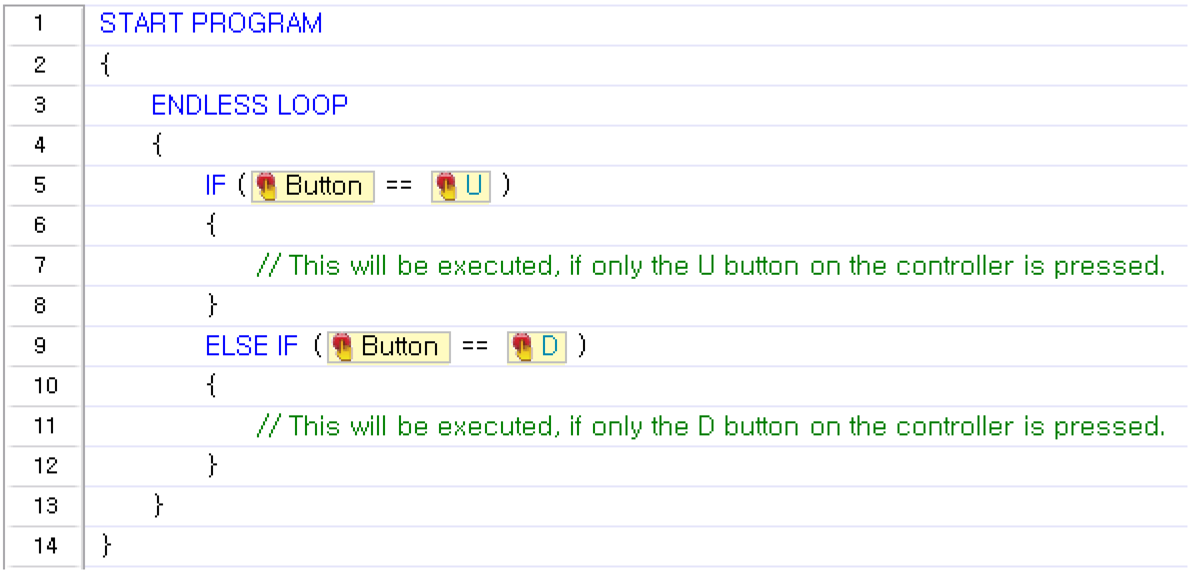
Characteristics

- Each button is assigned a unique value as follows.
- R button : 1, L button : 2, D button : 4, U button : 8, START button : 16
- When several buttons are pressed, the value assigned to the pressed buttons are added and read.
- Even if you do not know the buttons' code values, you can easily determine which buttons have been pressed by using the buttons' constant values.



Example

- This example shows how to perform different motions depending on which button is pressed.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        IF ( Button == U )
        {
            // This will be executed, if only the U button on the controller is pressed.
        }
        ELSE IF ( Button == D )
        {
            // This will be executed, if only the D button on the controller is pressed.
        }
    }
}
```

ROBOTIS Tech Support v1.00

Timer

Last updated 2010. 1. 18 (v1.0 Eng)

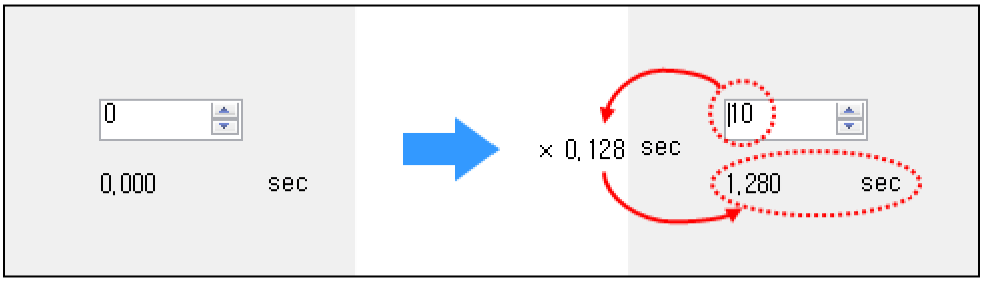
This parameter is used read the timer's current value or to set the timer, which begins to count down automatically. The timer is located in the controller.

Compatible Controller

- CM-100
- CM-5
- CM-510
- CM-700

Characteristics

- You can use constant values to set the timer's value.

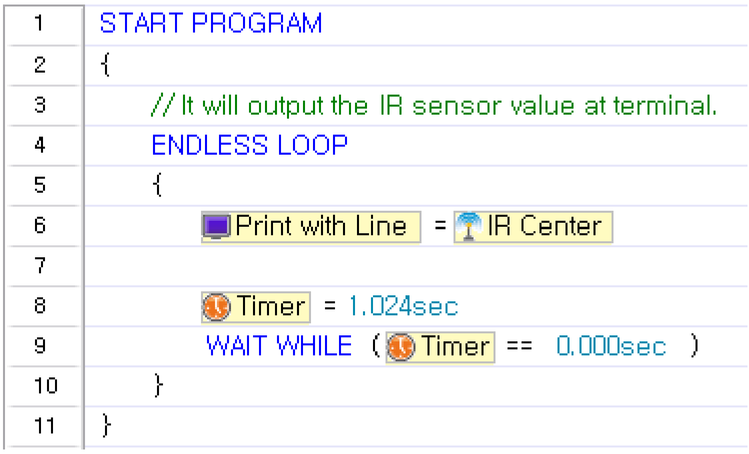


When a decimal number is entered, it will automatically be converted to the corresponding timer value.

- The actual timer value is between 0 and 255. Each timer value is 0.128 seconds.
- If you set a value greater value than 0 in the timer parameter, the timer will start to count down every 0.128 seconds.

Example

- The code below will print the value from the Center IR sensor every second.



```
START PROGRAM
{
    // It will output the IR sensor value at terminal.
    ENDLESS LOOP
    {
        Print with Line = IR Center

        Timer = 1.024sec
        WAIT WHILE ( Timer == 0.000sec )
    }
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Parameter > Controller > Remocon ID

ROBOTIS Tech Support v1.00

Remocon ID

Last updated 2010.1.18 (v1.0 Eng)

This parameter is used to set or read the ID of the opponent's Zigbee communication module.

Compatible Controller

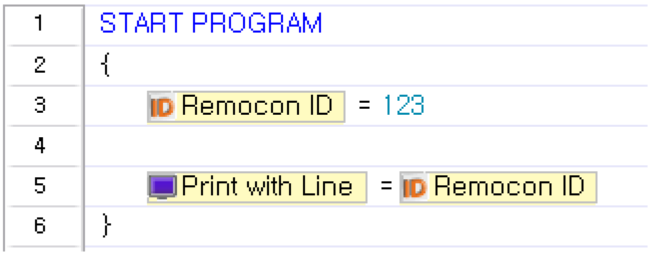
- CM-5
- CM-510
- CM-700

Characteristics

- The ID is a number between 0 and 65535.
- When the opponent's ID is set to 65535(0xFFFF, in hexadecimal), it will send data to all Zigbee modules, regardless of ID.(Broadcasting Mode)

Example

- This example sets the opponent's wireless ID to "123", reads the value, and prints it on the screen.



```
START PROGRAM
{
    Remocon ID = 123

    Print with Line = Remocon ID
}
```

Tip

- For seamless Zigbee communication, the opponent's wireless ID must be set to the correct value.
- Using the broadcasting mode improperly may cause unforeseen problems.

My ID

CM-100 : This parameter is used to determine whether or not a module ZIG-110 module has been installed.

Other Controllers : This parameter is used to read the ID of the Zigbee module installed in the robot.

Compatible Controller

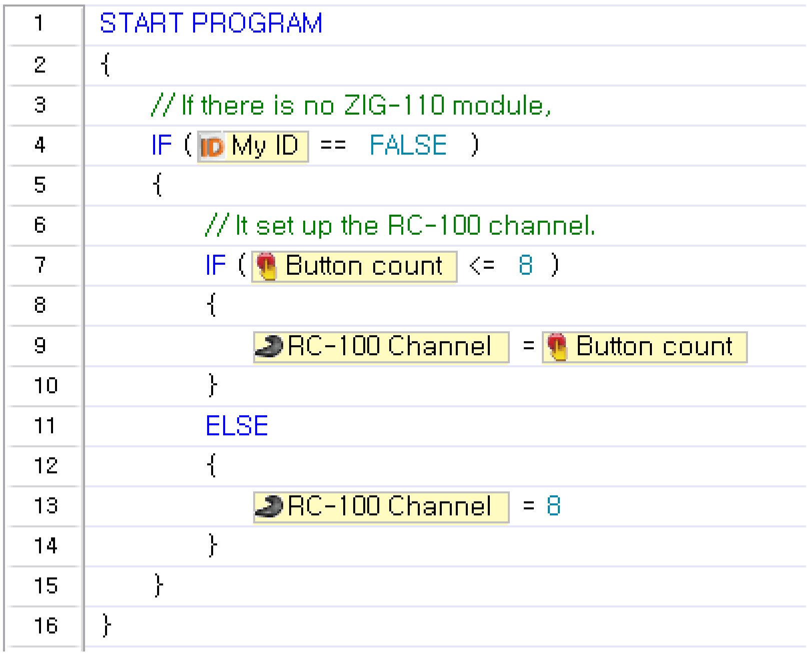
- CM-100
- CM-5
- CM-510
- CM-700

Characteristics

- CM-100 : If a ZIG-110 wireless communication module is installed, TRUE(1) is returned. Otherwise, FALSE(2), is returned.
- Other Controllers : If a ZigBee module is installed, its ID is read (a number between 0 and 65534). If not, 65565 (0xFFFF in hexadecimal) is returned..

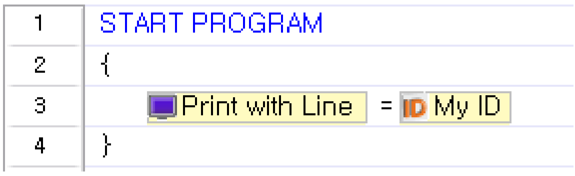
Example

- This example checks whether a ZIG-110 module is installed in the CM-100 controller. If the module is not installed, it sets the RC-100 channel according to the number of times the start button is pressed.



```
START PROGRAM
{
    // If there is no ZIG-110 module,
    IF ( My ID == FALSE )
    {
        // It set up the RC-100 channel.
        IF ( Button count <= 8 )
        {
            RC-100 Channel = Button count
        }
        ELSE
        {
            RC-100 Channel = 8
        }
    }
}
```

- This example prints the ZigBee module's ID. This code can be used with controllers other than CM-100.



```
START PROGRAM
{
    Print with Line = My ID
}
```

IR Left / Center / Right

Last updated 2010. 1.18 (v1.0 Eng)

This parameters are used to read the IR sensors' values.

Compatible Controller

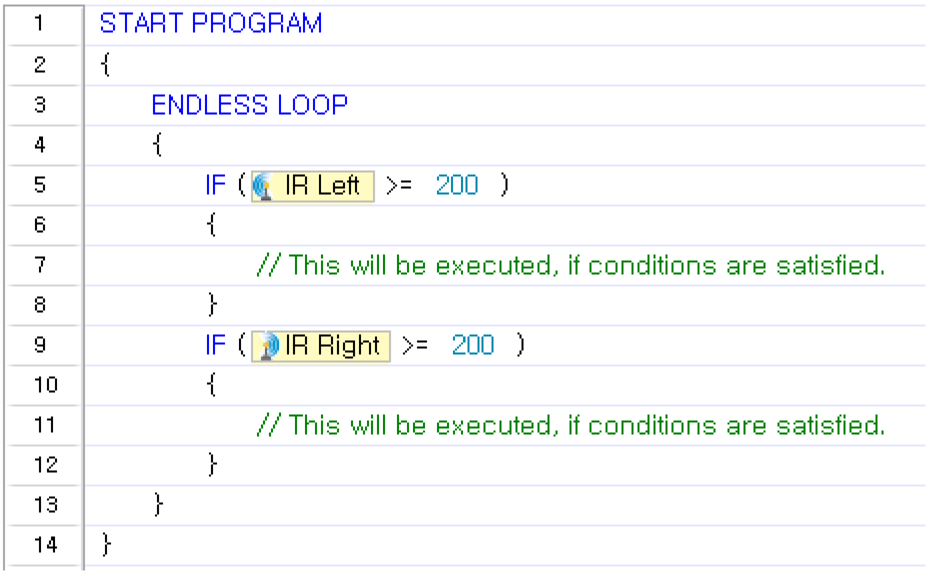
- CM-100

Characteristics

- The sensor value is between 0 and 1023.
- For objects with the same or similar color, the closer it is, the higher the value (closer to 1023), and the farther away it is, the lower the value (closer to 0).
- For objects with the same distance, the lighter (white) the object, the higher the value, and the darker (black) the object, the lower the value.

Example

- In this example, specific motions are performed when only the left IR sensor detects an object or when only the right IR sensor detects an object..



```
START PROGRAM
{
    ENDLESS LOOP
    {
        IF ( IR Left >= 200 )
        {
            // This will be executed, if conditions are satisfied.
        }
        IF ( IR Right >= 200 )
        {
            // This will be executed, if conditions are satisfied.
        }
    }
}
```

Tip

- Sensor values may be affected by external lights which emit infrared rays such as sunlight or a fluorescent lamp.

- Sensor values depend on the object's color or surrounding light, so using the IR sensor to measure the exact distance is not recommended.
- As above mentioned, IR sensor values are different if objects have different colors, even if they are the same distance away.   This characteristic can be utilized to distinguish black from white. (Can be used for tracing line)

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Parameter > Controller > Buzzer Index

ROBOTIS Tech Support v1.00

Buzzer Index (Controller's Buzzers)

Last updated 2010.1.18 (v1.0 Eng)

This parameter is used to set the musical note or melody to be played or to retrieve the note or melody currently being played using the buzzers in the controller.

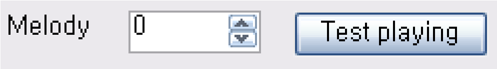
Compatible Controller

- CM-100
- CM-510

Characteristics

- The "Buzzer Time" parameter must always be used with the "Buzzer Index" parameter. "Buzzer Time" must be set before "Buzzer Index" is set. (**The order is important**) (Click here for more information on "Buzzer Time".)
- Depending on what the "Buzzer Time" is set to, "Buzzer Index" can be set to play a musical note or a melody.
  - **When "Buzzer Time" is set to 255 : Melody Mode**

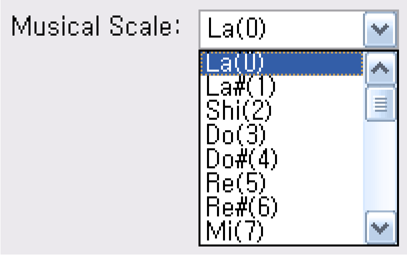
    Choose from 16 different melodies (0~15).



Melody 0

  - **When "Buzzer Time" is between 0 and 254 : Musical Note Mode**

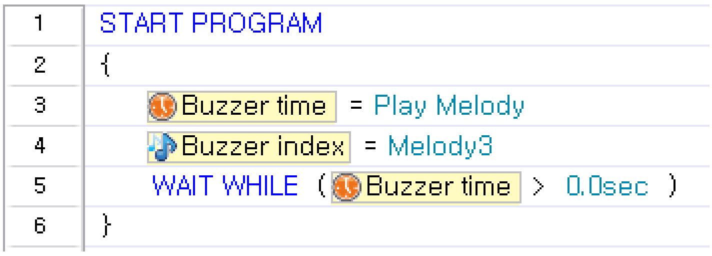
    Choose from 27 notes. The selected notes will play for the length set as " Buzzer Time".



Musical Scale: La(0)

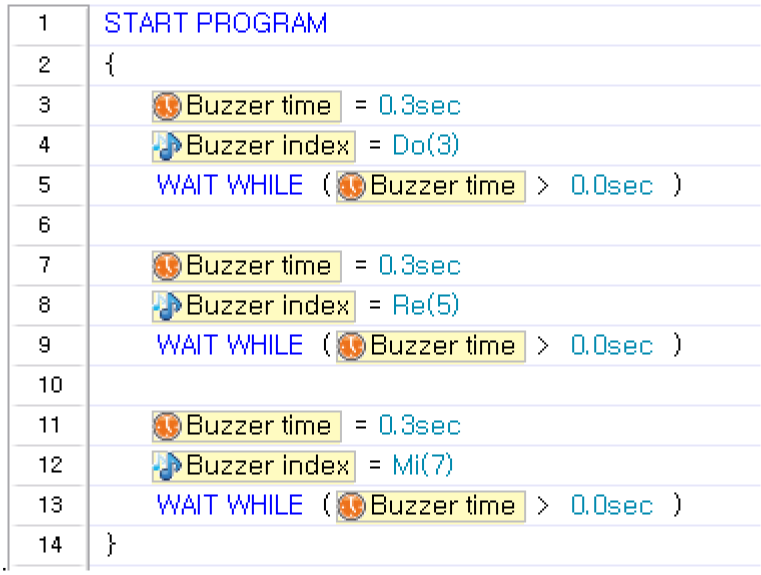
Example

- Plays melody 3.



```
START PROGRAM
{
    Buzzer time = Play Melody
    Buzzer index = Melody3
    WAIT WHILE ( Buzzer time > 0.0sec )
}
```

- Plays Do, Mi and Sol for 0.3 seconds each.



```
START PROGRAM
{
    Buzzer time  = 0.3sec
    Buzzer index  = Do(3)
    WAIT WHILE  ( Buzzer time  >  0.0sec  )

    Buzzer time  = 0.3sec
    Buzzer index  = Re(5)
    WAIT WHILE  ( Buzzer time  >  0.0sec  )

    Buzzer time  = 0.3sec
    Buzzer index  = Mi(7)
    WAIT WHILE  ( Buzzer time  >  0.0sec  )
}
```

Buzzer Time (Controllers' Buzzers)

This parameter is used to set how long the note or melody will be played or to retrieve how much longer it will be played.

Compatible Controller

- CM-100
- CM-510

Characteristics

- The "Buzzer Time" parameter is always used with the "Buzzer Index" parameter. "Buzzer Time" must be set before "Buzzer Index" is set. (**The order is important**) (Click here for more information on "Buzzer Time.")
- "Buzzer Time" can be set to a value between 0 and 255.
- Each value represents 0.1 second. For example, when "Buzzer Time" is set to 1, the note will be played for 0.1 second. The maximum length a note will be played is 5 seconds. Therefore, when values between 50 and 254 are entered, the note will be played for 5 seconds.
    - **When "Buzzer Time" is set to 255 : Melody Mode**

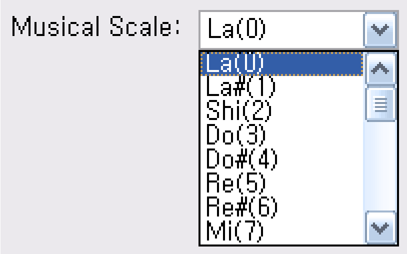
      Choose from 16 different melodies ('0~15) .

      When the melody finishes playing, "Buzzer Time" is reset to 0.

Melody 0

- **When "Buzzer Time" is between 0 and 254 :Musical Note Mode**

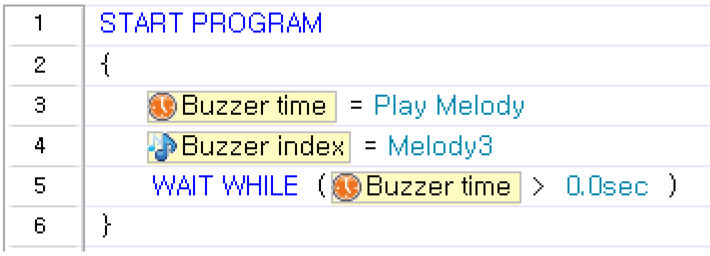
  Choose from 27 notes. The selected note will play for the length set as "Buzzer Time."



Musical Scale: La(0)

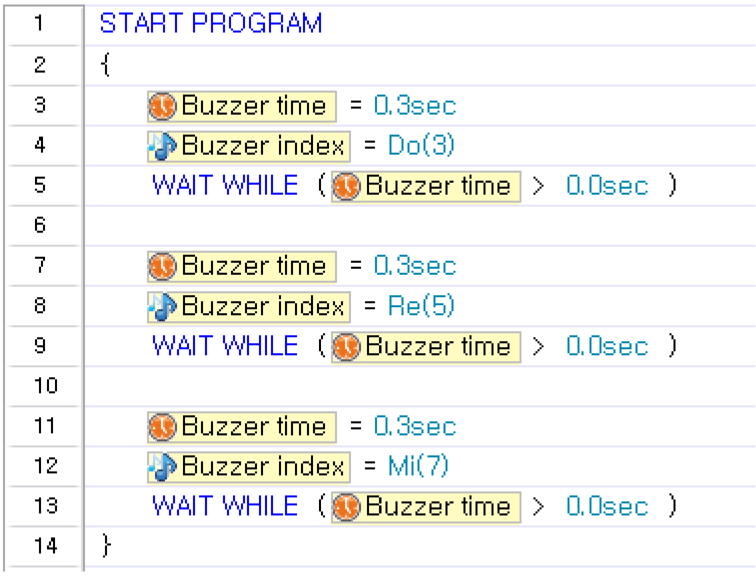
Example

- Plays melody 3. (Same as the example in "Buzzer Index")



```
START PROGRAM
{
    Buzzer time  = Play Melody
    Buzzer index  = Melody3
    WAIT WHILE  ( Buzzer time  >  0.0sec  )
}
```

- Plays Do, Mi and Sol for 0.3 seconds each.(Same as the example in "Buzzer Index")



```
START PROGRAM
{
    Buzzer time  = 0.3sec
    Buzzer index  = Do(3)
    WAIT WHILE  ( Buzzer time  >  0.0sec  )

    Buzzer time  = 0.3sec
    Buzzer index  = Re(5)
    WAIT WHILE  ( Buzzer time  >  0.0sec  )

    Buzzer time  = 0.3sec
    Buzzer index  = Mi(7)
    WAIT WHILE  ( Buzzer time  >  0.0sec  )
}
```

Tip

- "Buzzer Time" cannot be set while a note or melody is being played.

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Parameter > Controller > Sound Count

ROBOTIS Tech Support v1.00

Sound Count (Controller's Mic)

Last updated 2010.1.19 (v1.0 Eng)

A controller equipped with a microphone has a function to count sounds when the sound is louder than a certain threshold. For example, it is possible to count claps. This parameter is used to retrieve the number of detected sounds.

Compatible Controller

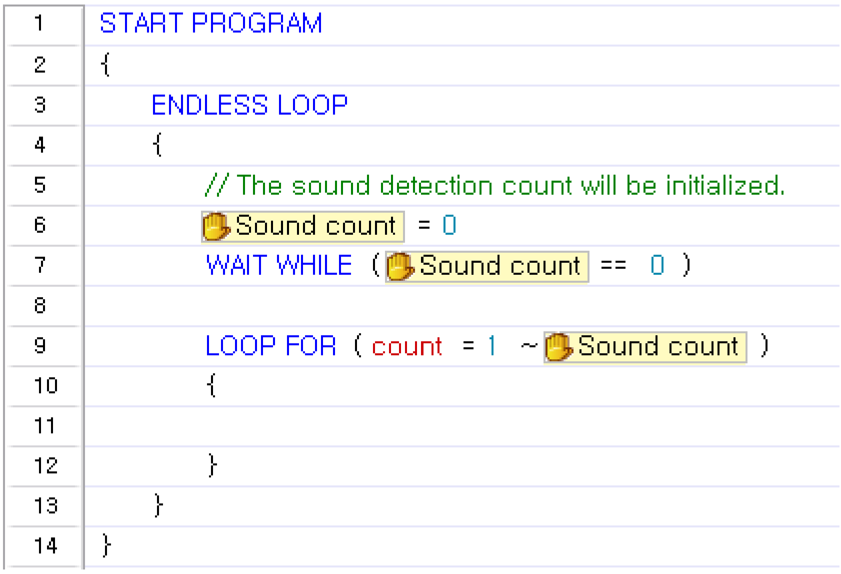
- CM-100
- CM-510

Characteristics

- "Sound Count" uses the numbers between 0 and 255. As a result, the maximum number of sounds counted is 255.
- When the sounds are no longer detected, the number of detected sounds will be input into the "Sound Count" parameter.
- Because "Sound Count" is not initialized automatically, you have to reset it to 0 before use.

Example

- Detects sounds and repeats a specific motion for as many times as it is detected.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        // The sound detection count will be initialized.
        Sound count = 0
        WAIT WHILE ( Sound count == 0 )

        LOOP FOR ( count = 1 ~ Sound count )
        {

        }
    }
}
```

Tip

- The geared motor connected to the controller may make loud noises while moving, which will be detected by the microphone. Please use the sound detection function only when the OLLO has stopped moving completely.

Current Sound Count (Controller's Mic)

A controller equipped with a microphone has a function to count sounds when the sound is louder than a certain threshold. For example, it is possible to count claps. This parameter is used to retrieve the number of detected sounds.

Compatible Controller

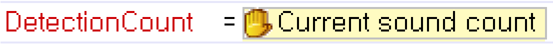
- CM-100
- CM-510

Characteristics

- "Current Sound Count" uses numbers between 0 and 255. As a result, the maximum number of sounds counted is 255.
- The parameter value is increased in real-time whenever a sound is detected.
- If a new sound is not detected for 0.8 seconds, the value of the "Current Sound Count" parameter is passed to the "Sound Count" parameter, and the "Current Sound Count" parameter is reset to 0.

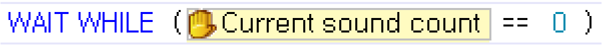
Example

- This code saves the current sound count in the "DetectionCount" variable.



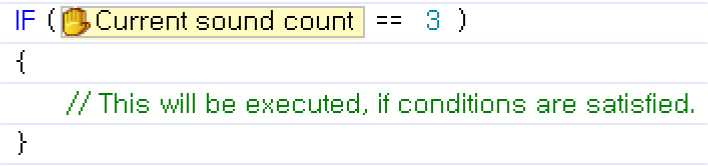
```
DetectionCount   = Current sound count
```

- This code pauses the program when no sounds are detected.



```
WAIT WHILE  ( Current sound count  ==  0  )
```

- This code executes a block of code when 3 sounds are detected.



```
IF ( Current sound count  ==  3  )
{
    // This will be executed, if conditions are satisfied.
}
```

Tip

- When it is connected with controller, sometimes the sounds of geared motor can be too loud to be input in the controller in normal way. Please use the sound detection function only when the OLLO has stopped moving completely.

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Parameter > Controller > Button Count

ROBOTIS Tech Support v1.00

Button Count

Last updated 2010.1.19 (v1.0 Eng)

This parameter is used to read how many times the START button was pressed when the controller was first turned on.

Compatible Controller

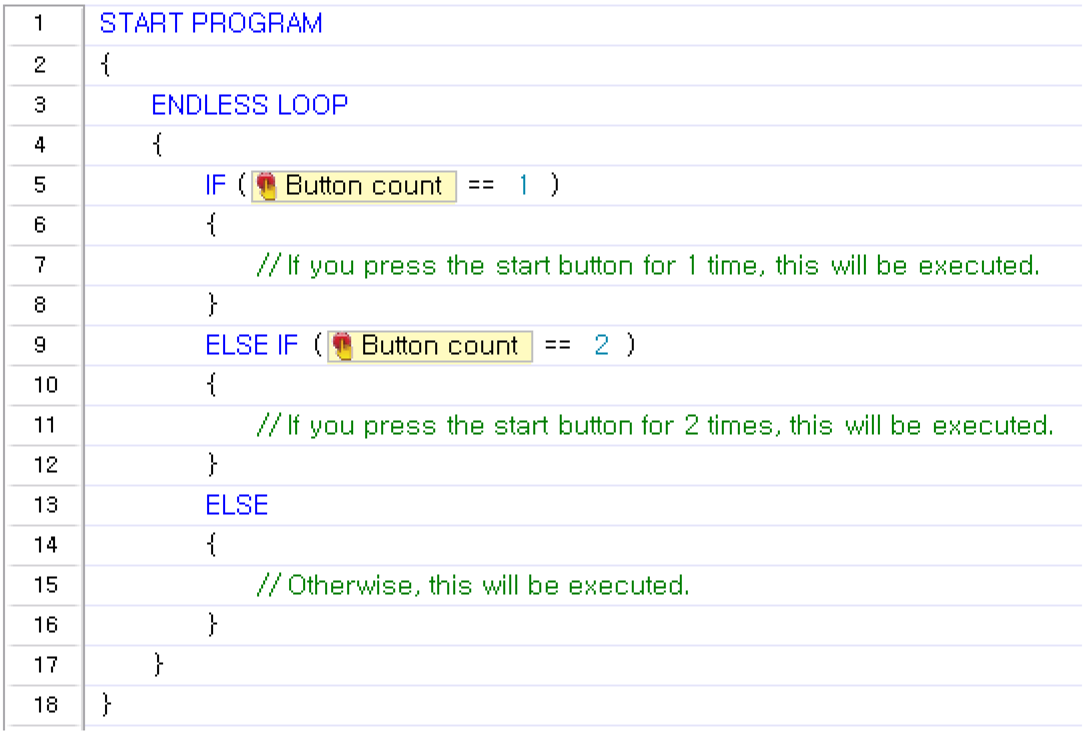
- CM-100

Characteristic

- "Button Counts" uses numbers between 0 and 255. As a result, only up to 255 button presses can be counted .

Example

- The example executes different motions according to how many times the START button was pressed - once, twice, or more.



```
START PROGRAM
{
    ENDLESS LOOP
    {
        IF ( Button count == 1 )
        {
            // If you press the start button for 1 time, this will be executed.
        }
        ELSE IF ( Button count == 2 )
        {
            // If you press the start button for 2 times, this will be executed.
        }
        ELSE
        {
            // Otherwise, this will be executed.
        }
    }
}
```

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Parameter > Controller > PowersaveTimer

ROBOTIS Tech Support v1.00

Powersave Timer

Last updated 2010.1.19 (v1.0 Eng)

The controller has a hibernate function to conserve battery. If no commands are received for a set period, the controller can turn itself off. This parameter is used to set how long the controller will wait or how much time is left.

Compatible Controller

- CM-100

Characteristics

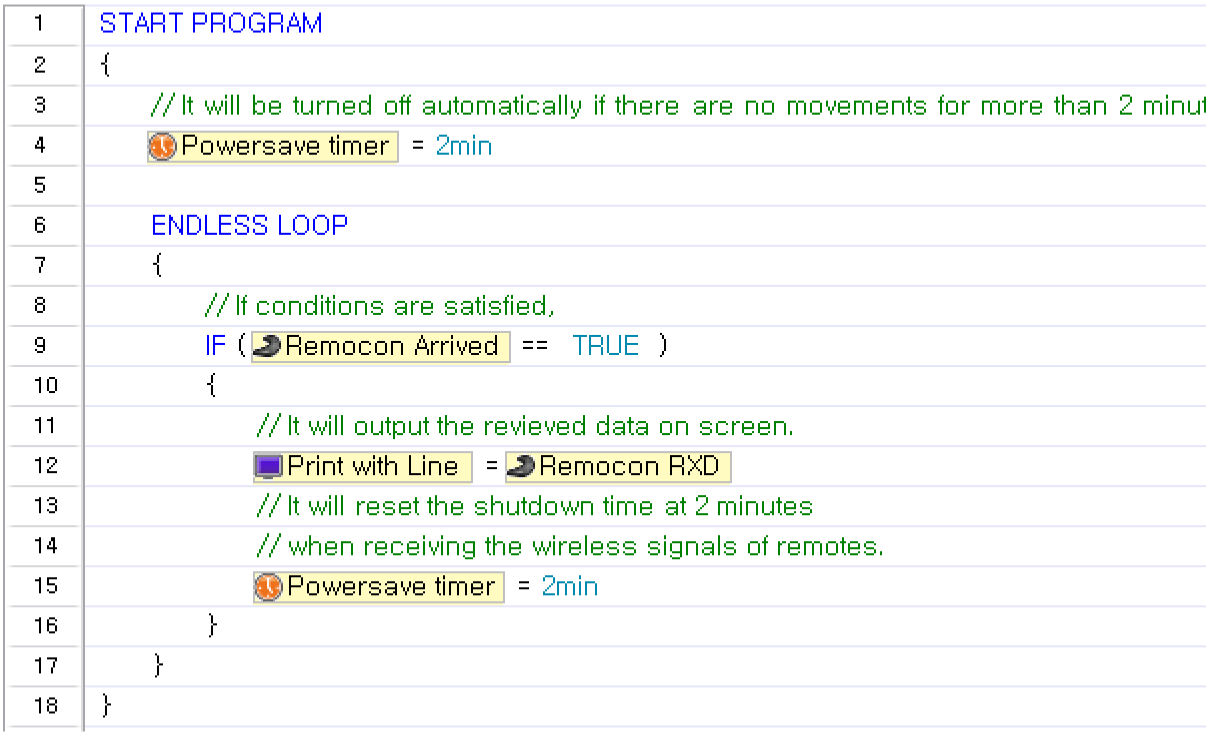
- "Powersave timer" can be set using powersave constants.

0 min

- "Powersave timer" uses numbers between 0 and 255.
- The unit is minutes (i.e., a value of 1 equals 1 minute)
- The default value is 5 minutes.
- Setting the "Powersave timer" to 0 will turn it off.
- The time remaining on the timer is always in minutes. For example, when 50 seconds remain, the timer will say that 1 minute remains .

Example

- The controller will be turned off if no data is received for 2 minutes. If data is received, the timer is reset to 2 minutes.



```
START PROGRAM
{
    // It will be turned off automatically if there are no movements for more than 2 minut
    Powersave timer  = 2min

    ENDLESS LOOP
    {
        // If conditions are satisfied,
        IF ( Remocon Arrived  ==  TRUE  )
        {
            // It will output the revieved data on screen.
            Print with Line  = Remocon RXD
            // It will reset the shutdown time at 2 minutes
            // when receiving the wireless signals of remotes.
            Powersave timer  = 2min
        }
    }
}
```

Tip

- To keep the controller from turning itself off even when certain actions are performed, you must manually reset "Powersave timer."

Home > Software Help > RoboPlus > RoboPlus Task > Programming > Parameter > Controller > RC-100 Channel

ROBOTIS Tech Support v1.00

RC-100 Channel (Controller)

Last updated 2010.1.19 (v1.0 Eng)

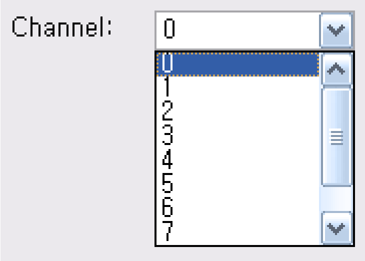
This parameter is used to set up the infrared communication channel or to check the current channel between the controller's IR receiver and RC-100.

Compatible Controller

- CM-100
- CM-510

Characteristics

- The RC-100 channel can be set using constant numbers.

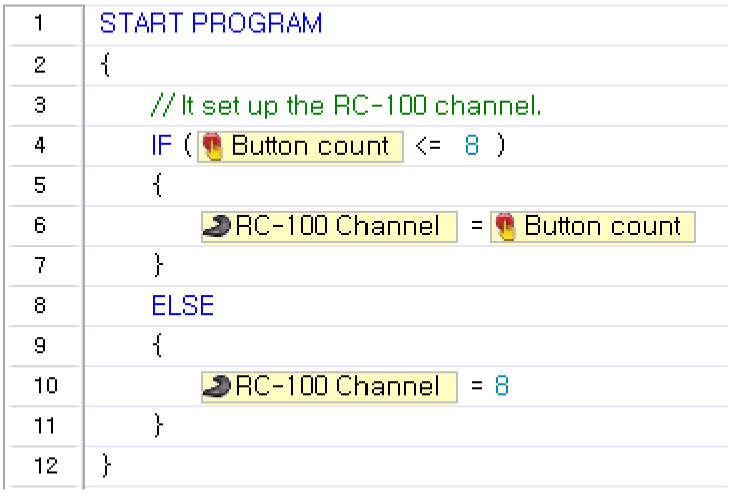


Channel: 0

- "RC-100 Channel" uses numbers between 0 and 8.
- The Channel 0 is the special one that can be communicated with every other channels.

Example

- Sets the RC-100 channel according to how many times the START button was pressed.



```
START PROGRAM
{
    // It set up the RC-100 channel.
    IF ( Button count  <=  8 )
    {
        RC-100 Channel  =  Button count
    }
    ELSE
    {
        RC-100 Channel  = 8
    }
}
```