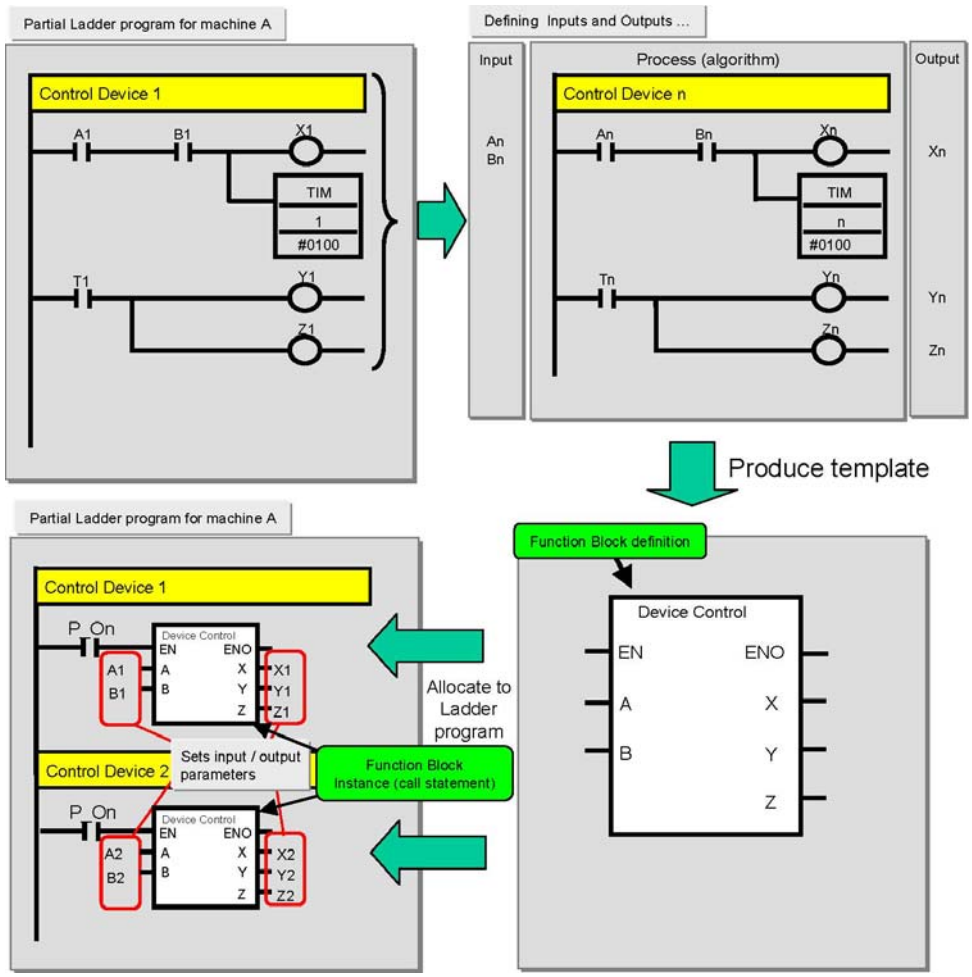
PLC Controlled Elevator

LAB 3 TUTORIAL

Contents:

1. What is Function Block?

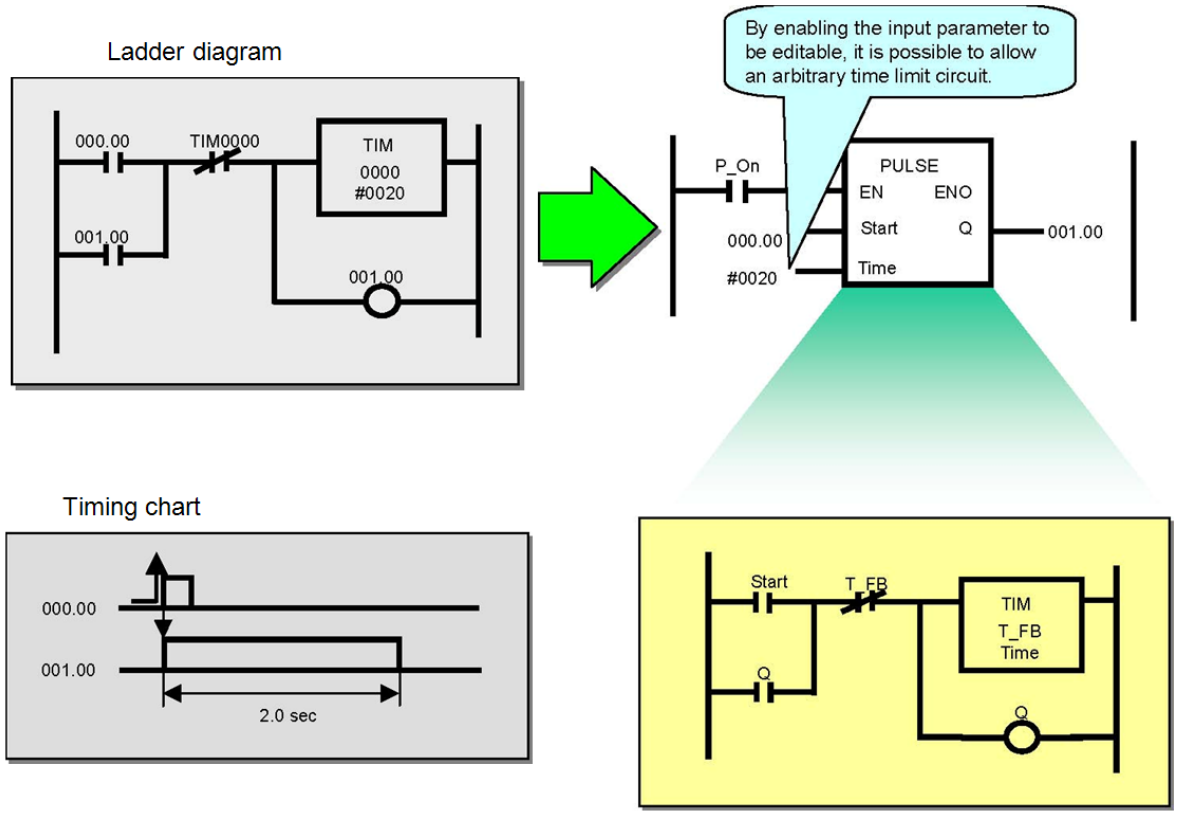
Function Blocks are predefined programs (or functions) contained within a single program element that may be used in the ladder diagram. A contact element is required to start the function, but inputs and outputs are editable through parameters used in the ladder arrangement. The functions can be reused as the same element (same memory) or occur as a new element with its own memory assigned.



Function Block definition **…** This contains the defined logic (algorithm) and I/O interface. The memory addresses are not allocated in the Function Block Definition Function Block instance (call statement) **…** This is the statement that will call the function block instance when used by the ladder program, using the memory allocated to the instance

2. An example of Function Block

The following figures describe an example of a function block for a time limit circuit, to be used in the ladder. It is possible to edit the set point of the TIM instruction to reallocate the set time for turning off the output in the ladder rung. Using the function block as shown below, it is possible to make the time limit of the circuit arbitrary by only changing one specific parameter.



3. Function Block library

The Function Block Library is a collection of predefined Function Block files provided by the PLC manufacturer. These files are intended to be used as an aid to simplify programs, containing standard functionality for programming PLCs and Omron FA component functions. The Function Block Part file is prepared using the ladder diagram function block, for defining each function of the PLC unit and the FA component. The files contain a program written in ladder diagram and have the extension .CXF. The file name of the Function Block Part file begins with '_' (under score).

Here is a list of the benefits to be gained from using the FB Library:

(1) No need to create ladder diagrams using basic functions of the PLC units and FA components: More time can be spent on bespoke programs for the external devices, rather than creating basic ladder diagrams, as these are already available.

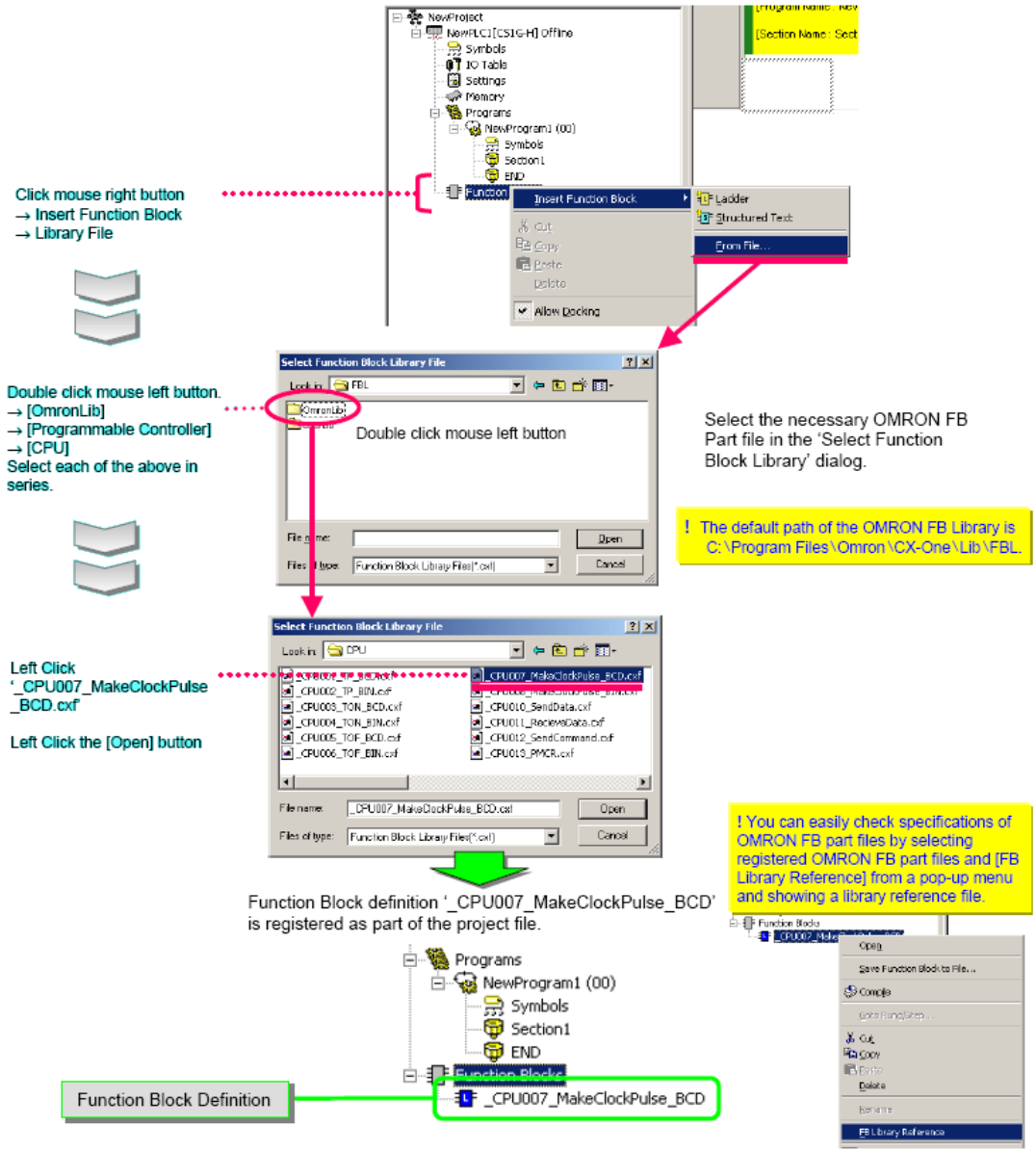
(2) Easy to use: A functioning program is achieved by loading the function block file to perform the target functionality, then by inputting an instance (function block call statement) to the ladder diagram program and setting addresses (parameters) for the inputs and outputs.

(3) Testing of program operation is unnecessary: Omron has tested the Function Block library. Debugging the programs for operating the unit and FA components for the PLCs is unnecessary for the user.

(4) Easy to understand: The function block has a clearly displayed name for its body and instances. A fixed name can be applied to the process. The instance (function block call statement) has input and output parameters. As the temporary relay and processing data is not displayed, the values of the inputs and outputs are more visible. Furthermore, as the modification of the parameters is localised, fine control during debugging etc. is easier. Finally, as the internal processing of the function block is not displayed, when the instance is used in the ladder diagram, the ladder diagram program looks simpler to the end user.

3.1. Import Function Block part file

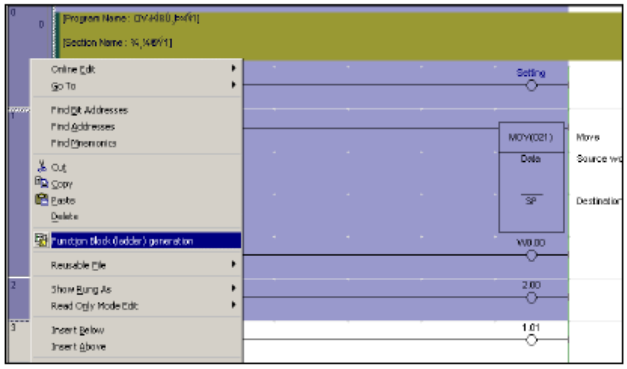
Select Function Block definition icon from the project tree using the mouse cursor, right click. Select Insert Function Block, then select a Library file using mouse to navigate.



4. Generating FBs Based on an Existing Ladder Program

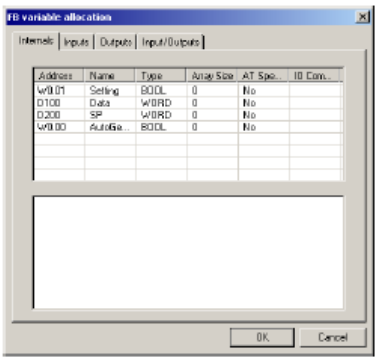
FBs can be generated easily based on programs with proven operating results. This function can accelerate the conversion of program resources to FBs.

Select the program section that you want to convert to an FB and right-click the mouse.

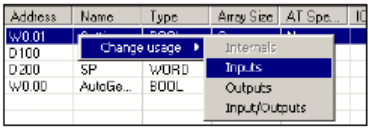


Select ***Function Block (ladder) generation***.

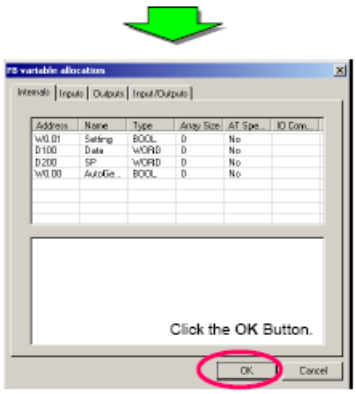
The FB Variable Allocation Dialog Box will be displayed.

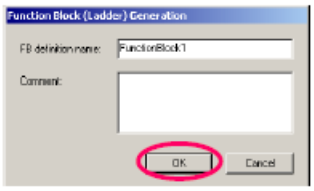


When necessary, change the usage of variables and addresses (internal variable, input variable, output variable, or input-output variable) used in the program section. Select the variable and select ***Change usage*** from the pop-up menu.



Note:
If a variable does not exist in an address being used in the program, a variable starting with "AutoGen" will be added automatically.
When the FB is called in the program, parameters are displayed as variable names, so at a minimum we recommend changing input, output, and input-output variables to easy-to-understand variable names.
To change the names, double-click the address that you want to change in the FB variable allocation Dialog Box to display a dialog box in which the name can be changed.

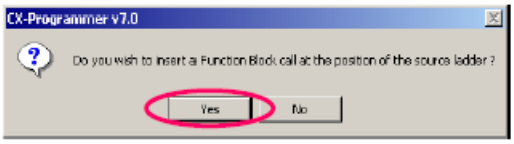


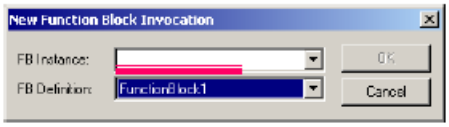


Input the FB definition name and comment, and click the OK Button.

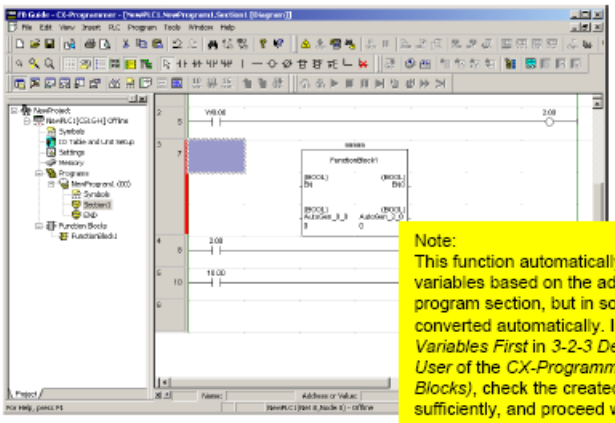
The FB definition will be created.

To insert an FB call instruction created in the ladder program, click the **Yes** Button.





Input the FB instance name and click the **OK** Button.



Note:
This function automatically determines the usage of variables based on the addresses used in the selected program section, but in some cases usage cannot be converted automatically. In these cases, refer to *Registering Variables First* in *3-2-3 Defining Function Blocks Created by User* of the *CX-Programmer Operation Manual (Function Blocks)*, check the created FB definition, verify operation sufficiently, and proceed with actual operation.

The FB call instruction will be inserted in the ladder program.