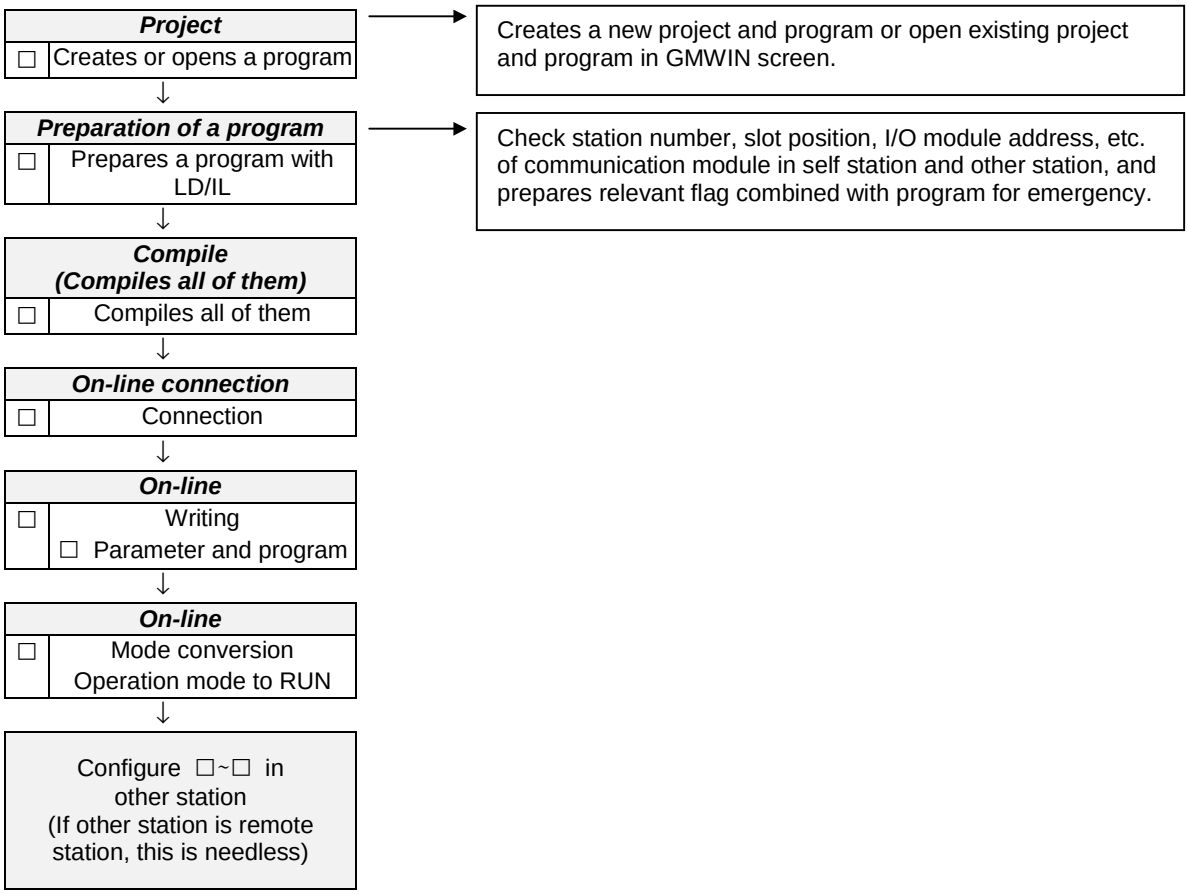
6.3 *Function block* service

6.3.1 Introduction

Function block is the function to read or write specific data or specific variable data to certain area or specific variable area of other station when certain event occurred, and this can be used various according to usage.

6.3.2 Programming procedure of *function block*



6.3.3 Types of *function block*

Function block that is used in preparing program is classified to 4 commands according to usage.

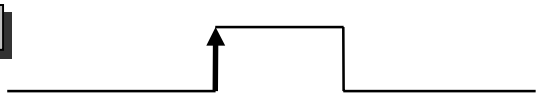
Type	Usage
READ, WRITE	Reads data of other station or writes data to other station
STATUS	Checks current status of PLC in other station
CONNECT	Establishes logical communication channel with other company's PLC (applied to Mnet only)
DA4INI, etc.	Accesses special module in FSM

6.3.4 Input/Output of *function block*

This explains common I/O of *function block*.

Input

REQ :



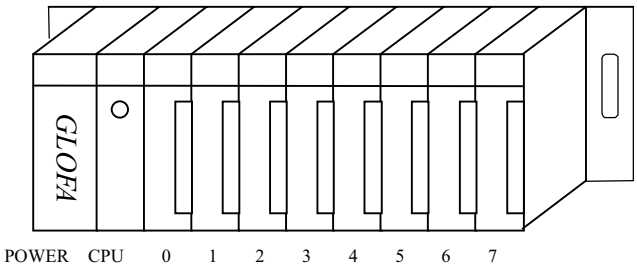
This is used as operation condition except CONNECT *function block*. This is operated at rising edge that is changed '0' to '1'. Once this is operated, this is not affected until receiving response from other station, and restarts from scan after NDR or ERR bit is set.

EN :

Function block is operated if level is '1', and this should maintain state of '1' during service(Applied in CONNECT *function block* only : BOOL type). If EN bit is '1' when ERR bit is set to 'On', *function block* requests establishment service of communication channel again at next scan. If the value is changed from '1' to '0', it requests normal cancellation of channel established.

NET_NO :

This specifies the slot location of communication module to execute this *function block* among communication module that is mounted in basic base of GLOFA PLC. Right side of CPU in basic base is '0' slot.



ST_NOH / ST_NOL :

Specify upper/lower number of other station number for Fnet, SAP number for Mnet.
For details, see RDTYPE *function block*.

Output

NDR :

After *function block* is operated, this is set to 'On' if data is normally received, and set to 'Off' if appropriate scan is finished. If this bit is used for input condition of other *function block*, user can make a reliable communication.

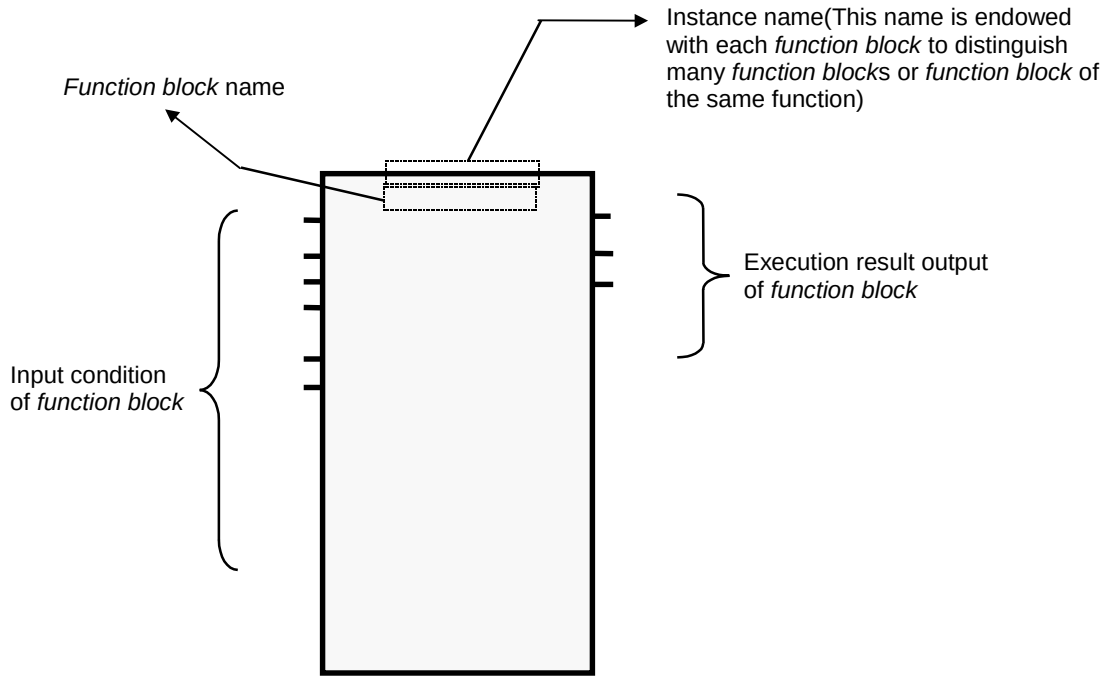
ERR :

This is set to 'On' when error occurred after *function block* is operated, and maintains 'On' until next *function block* is operated and then becomes 'Off'. If error occurred, data is not received(See Received error from communication module of Chap. 6.3.7).

STATUS :

This shows detailed code value of error when error occurred after *function block* is operated, and maintains 'values' until next *function block* is operated and then becomes 'Off'(See Received error from communication module of Chap. 6.3.7).

6.3.5 How to use *function block*



(In some *function block*, output result is outputted to left side)

6.3.6 *Function block* library of link

This explains each *function block*.

Table 6.3.6 Usable PLC type of *function block* and classification according to type

(O : Available, X : Not available)

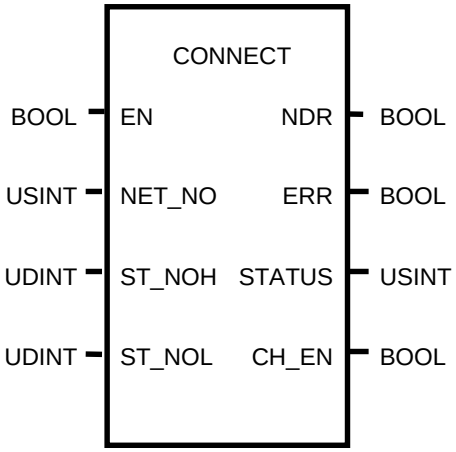
Data type(bit size) \ PLC type		GM1 GM2	GM3	GM4	GM5	GM6	Remote	Name of *function block*(RD/WR)
Basic type (Max. setting No. 4)	BOOL(1)	O	O	O	O	O	O	RD(WR)BOOL
	BYTE(8)	O	O	O	O	O	O	RD(WR)BYTE
	WORD(16)	O	O	O	O	O	O	RD(WR)WORD
	DWORD(32)	O	O	O	O	O	O	RD(WR)DWORD
	LWORD(64)	O	X	X	X	X	X	RD(WR)LWORD
	USINT(8)	O	O	O	O	O	X	RD(WR)USINT
	UINT(16)	O	O	O	O	O	X	RD(WR)UINT
	UDINT(32)	O	O	O	O	O	X	RD(WR)UDINT
	ULINT(64)	O	X	X	X	X	X	RD(WR)ULINT
	SINT(8)	O	O	O	O	O	X	RD(WR)SINT
	INT(16)	O	O	O	O	O	X	RD(WR)INT
	DINT(32)	O	O	O	O	O	X	RD(WR)DINT
	LINT(64)	O	X	X	X	X	X	RD(WR)LINT
	REAL(32)	O	X	X	X	X	X	RD(WR)REAL
	LREAL(64)	O	X	X	X	X	X	RD(WR)LREAL
	TIME(16)	O	O	O	O	O	X	RD(WR)TIME
	DATE(48)	O	O	O	O	O	X	RD(WR)DATE
	TIME of DAY(48)	O	O	O	O	O	X	RD(WR)TOD
	DATE and TIME(48)	O	O	O	O	O	X	RD(WR)DT
Block(Max. Fnet:120, Mnet:400 byte)		O	O	O	O	O	O	RD(WR)Block
Array type (Max. 100 byte)	BOOL	O	O	O	O	O	X	RD(WR)Array
	BYTE	O	O	O	O	O	X	RD(WR)Array
	WORD	O	O	O	O	O	O	RD(WR)Array
	DWORD	O	O	O	O	O	X	RD(WR)Array
	LWORD	O	X	X	X	X	X	RD(WR)Array
	USINT	O	O	O	O	O	X	RD(WR)Array
	UINT	O	O	O	O	O	X	RD(WR)Array
	UDINT	O	O	O	O	O	X	RD(WR)Array
	ULINT	O	X	X	X	X	X	RD(WR)Array
	SINT	O	O	O	O	O	X	RD(WR)Array
	INT	O	O	O	O	O	X	RD(WR)Array

PLC type / Data type(bit size)		GM1 GM2	GM3	GM4	GM5	GM6	Remote	Name of *function block*(RD/WR)
Array type (Max. 100 byte)	DINT	O	O	O	O	O	X	RD(WR)Array
	LINT	O	X	X	X	X	X	RD(WR)Array
	REAL	O	X	X	X	X	X	RD(WR)Array
	LREAL	O	X	X	X	X	X	RD(WR)Array
	TIME	O	O	O	O	O	X	RD(WR)Array
	DATE	O	O	O	O	O	X	RD(WR)Array
	TIME of DAY	O	O	O	O	O	X	RD(WR)Array
	DATE and TIME	O	O	O	O	O	X	RD(WR)Array

CONNECT

Establishes logical communication channel with other station(For connection with Mnet of other company)

Product name	GM1	GM2	GM3	GM4	GM5	GM6
Applicable	●	●	●			

Function block	Description
	**Input** REQ : Request execution of *function block*(FB) at rising edge(0→1). NET_NO: Slot No.(0~7) that communication module of self station that this FB is to be transmitted is mounted. ST_NOH: Upper station number and SAP of communication module mounted in other station to establish channel. ST_NOL : Lower station number of communication module mounted in other station to establish channel. **Output** NDR : Set to 'On', when data is received without error. ERR : Set to 'On', when error occurred after *function block* is executed. STATUS: Detailed code value of error when error occurred. CH_EN : Result of channel establishment.

■ Function and description

Communication with other station in Mnet is made after communication channel is established, and SAP is needed for this connection. SAP is classified to SAP(SSAP) of self station and SAP(DSAP) of other station, and manufacturer who supplies Mini-MAP communication module supplies user with SAP used in self station. Types of SAP include Association SAP, Associationless SAP, and Unspec. SAP.

- Association SAP
  Channel(Initiate) service should be used during communication service, after channel is established.
- Associationless SAP
  Not using channel(Initiate) service, and execute communication assuming that communication channel is established internally.
- Unspec. SAP
  Both Association SAP and Associationless SAP function are contented.

If the SAP supplied from other company is Association SAP or if communication channel(Initiate) service should be used to communicate in other company's product, communication channel(Initiate) should be established using CONNECT *function block*. However, in communication with product of self-company, communication not using this *function block* is possible.
Operation condition of *function block* is operated not with edge but with level. If channel is established once, channel is maintained continuously as long as EN input level is '1', and CH_EN bit is set to '1' when channel is established, and CH_EN bit is set to '0' when channel is cancelled by the request of self station or other station. Therefore, user can use CH_EN bit for the operation condition of other *function block*.
To communicate continuously, EN should be maintained to '1', and channel is cancelled normally if En bit is set to '0'.

■ **EN**

This is operated when level is '1', and should be maintained to '1' during service (BOOL).

■ **NET_NO**

Slot location(0~7) that communication module that data is transmitted using this *function block* is mounted, among communication modules mounted in main base of self station PLC.

■ **ST_NOH**

Upper station number and SAP of communication module mounted in other station to establish channel.
ST_NOH = SSAP(self station SAP) + DSAP(other station SAP) + Upper station number of communication module in other station.

■ **ST_NOL**

Lower station number of communication module mounted in other station to establish channel.
ST_NOL = Lower station number of other station

Ex.) Connection with Mnet of other company.
When communication channel is established with station B(other company's) in station A(self company's)
(Suppose that SAP = 4E of Mini-MAP module in other company's).
MAC address of A station : 16#00E091000000 (self company's),
MAC address of B station : 16#080070221C9A (other company's).
ST_NOH : 16#10(SAP of self station) 4E(SAP of other station) 0800(upper station number of communication module in other station).
Namely, ST_NOH = 16#104E0800.
ST_NOL = 16#70221C9A(lower station number of communication module in other station).

* SAP(SSAP) supplied with GLOFA Mini-MAP module for connection with Mnet of other company's has two types, 16#10 and 16#14.

■ **NDR**

This is set to 'On' when *function block* is operated and normally finished, and this is set to 'Off' when appropriate scan is finished.

■ **ERR**

This is set to 'On', if error occurred after *function block* is operated and the request for channel cancellation is received from other station with communication channel maintained, and this maintains 'On' until this *function block* is operated again in next scan.

■ **STATUS**

This shows detailed code value of error, and this maintains the value until this *function block* is operated again in next scan.

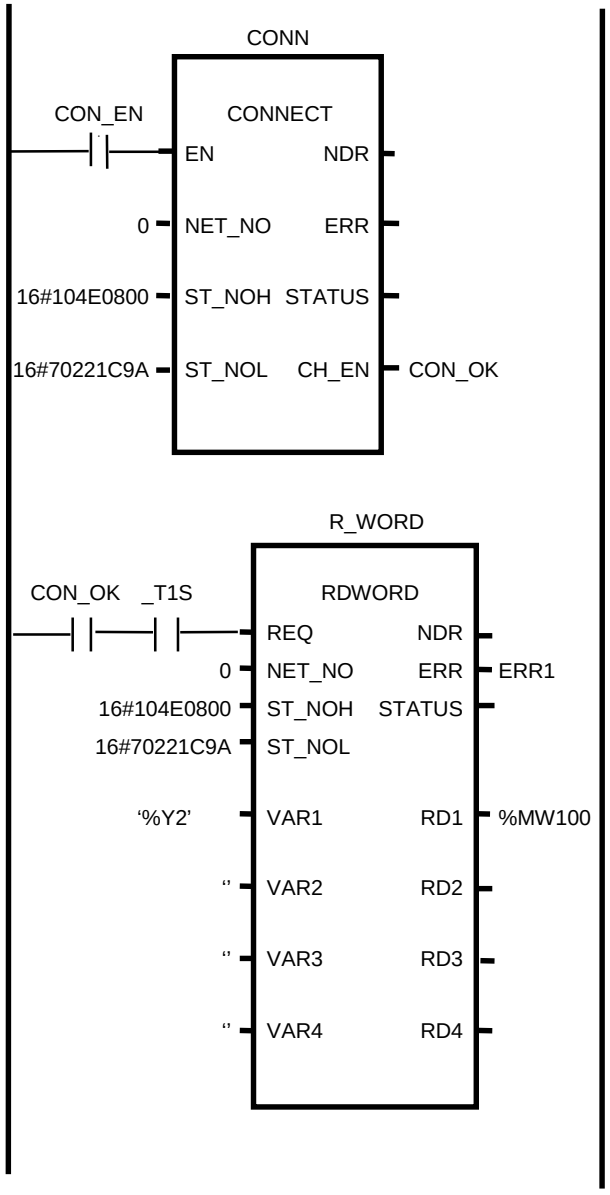
■ **CH_EN**

This is set to '1' when channel is established, and this is set to '0' when channel is cancelled. CH_EN is set to '1' when NDR is set to '1'. NDR bit is cleared at next time but CH_EN bit maintains '1' until channel is cancelled. CH_EN bit is set to '0' when ERR bit is set to '1', and maintains '0' until channel is established.

■ Example of program

Suppose that Mnet is used with Mnet of other company product and Mini-MAP module of self station is mounted in slot 0. If MAC address of other station is 16#080070221C9A and 1 word(16 bit) of Y2 in other station area is stored in %MW100 area in self station, after channel is established with Mini-MAP of other company(Suppose that other company's SAP = 4E).

LD



IL

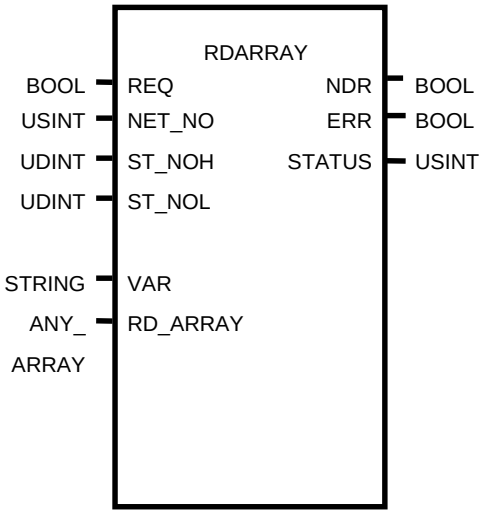
```
CAL CONNECT      CONN
     EN :=       TRUE(1)
     NET_NO :=   0
     ST_NOH :=   16#104E0800
     ST_NOL :=   16#70221C9A
 LD  CONN.CH_EN
 ST  CON_OK

 LD  CON_OK

CAL RDWORD R_WORD
     REQ :=    _T1S
     NET_NO :=   0
     ST_NOH :=   16#104E0800
     ST_NOL :=   16#70221C9A
     VAR1 :=   '%Y2'
     VAR2 :=   ''
     VAR3 :=   ''
     VAR4 :=   ''
 LD R_WORD.RD1
 ST  %MW100
 LD  R_WORD.ERR
 ST  ERR1
```

- CON_EN is a variable which has initialization value, 1.
  In above example, instance name of *function block* CONNECT is CONN. If ST_NOH = 16#104E0800 and ST_NOL = 16#70221C9A, *function block* operates all the time, and service is transmitted and received through communication module mounted in slot 0 of main base. If service is in proceeding and CH_EN bit is set, this means that channel is established between communication module of self station and communication module of other station to be communicated. Y2 of other station is read per 1 sec. with RDWORD function and it is stored in %MW100 area of self station.

RDARRAY

Reads data of array type from other station (Max. 100 byte)

Product name	GM1	GM2	GM3	GM4	GM5	GM6
Applicable	●	●	●	●	●	●

Function block	Description
	**Input** REQ : Request execution of *function block*(FB) at rising edge(0→1). NET_NO : Slot No.(0~7) that communication module of self station that this FB is to be transmitted is mounted. ST_NOH : Fixed to 0 in Fnet (In Mnet, upper station number and SAP of communication module mounted in other station). ST_NOL : Station number of communication module mounted in other station (In Mnet, lower station number). VAR : Variable identifier(direct address not available) to read data(variable defined in other station). RDARRAY: Self station area that ARRAY data received from other station is to be stored. **Output** NDR : Set to 'On', when data is received without error. ERR : Set to 'On', when error occurred after *function block* is executed. STATUS: Detailed code value of error when error occurred.

■ Function and description

This is the *function block*, which is used to read data of other station defined as array type.
Data can be read not using direct variable of other station(Ex. : %I, %Q, and %M area) but using variable name used in other station. Variable name used should be specified in access variable list of other station, and the variable name, used for path name when access variable is specified, should be defined again as array type in global variable list(definition is not necessary in self station). The data type, which is the same as the array defined in other station, should be used.

■ ST_NOH / ST_NOL

Station number of communication module in other station(For details, see RDTYPE *function block*).

■ VAR :

Variable identifier to be read from other station.

■ NDR / ERR / STATUS

Displays execution result of *function block*(For details, see RDTYPE *function block*).

■ RDARRAY

Self station area that array data received from other station is to be stored.
(The data type, which is the same as the array defined in other station, should be used)

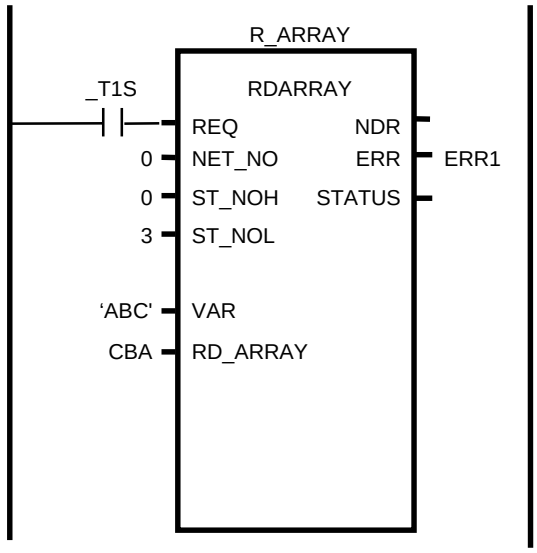
■ Example of program

Suppose that Fnet is used and Fnet module is mounted in slot 0.
If station number of other station is station 3, and variable of ABC defined as array in other station is read and it is stored in array variable of CBA in self station. (Access for ABC variable in station 3 and register global variable and set REQ condition with a period of 1 second)

- Variable registration for station 3 (For registration method of access variable, see 6.3.8 Access variable registration).

Variable registration	Variable name	Access path	Description
Access variable registration	ABC	DEF(Ex.)	Set the path name of access variable ABC to DEF
Resource global variable registration	RES1.DEF	-	Register again DEF to global variable

LD



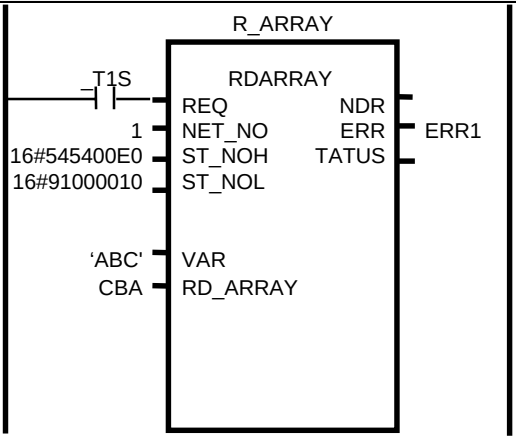
IL

```
CAL     RDARRAY     R_ARRAY
    REQ :=   _T1S
    NET_NO :=  0
    ST_NOH :=  0
    ST_NOL :=  3
    VAR :=   'ABC'
    RD_ARRAY :=  CBA

LD      R_ARRAY.ERR
ST      ERR1
```

■ Example of program

Suppose that Mnet is used and communication is made with Mnet of self company product and Mini-MAP module mounted in slot 1.
If MAC address of other station is 16#00E091000010 and variable of ABC defined as array in other station is read, and it is stored in array variable of CBA in self station(Access for ABC variable of other station. Register global variable as the Ex. of Fnet shown above, and set REQ condition with a period of 1 second)

LD



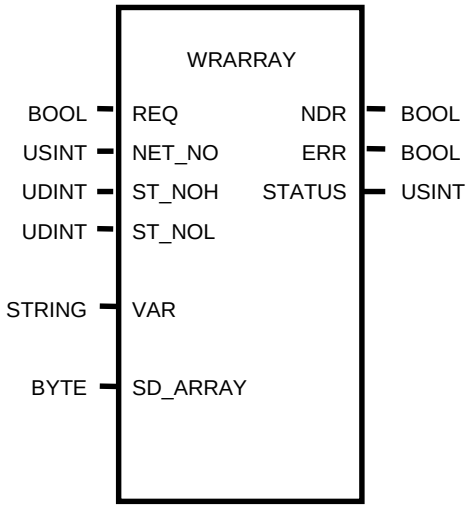
IL

```
CAL   RDARRAY    R_ARRAY
      REQ :=     _T1S
      NET_NO :=  1
      ST_NOH :=  16#545400E0
      ST_NOL :=  16#91000010
      VAR :=     'ABC'
      RD_ARRAY :=   CBA

LD      R_ARRAY.ERR
ST      ERR1
```

WRARRAY

Writes data of array type to other station (Max. 100 byte)

Product name	GM1	GM2	GM3	GM4	GM5	GM6
Applicable	●	●	●	●	●	●

Function block	Description		
	**Input**	REQ :	Request execution of *function block*(FB) at rising edge(0→1).
		NET_NO :	Slot No.(0~7) that communication module of self station that this FB is to be transmitted is mounted.
		ST_NOH :	Fixed to 0 in Fnet (In Mnet, upper station number and SAP of communication module mounted in other station).
		ST_NOL :	Station number of communication module mounted in other station (In Mnet, lower station number).
		VAR :	Variable identifier(direct address not available) to read data(variable defined in other station).
		SD_ARRAY :	Area for array data of self station to be transmitted to other station.
	Output	NDR :	Set to 'On', when data is transmitted without error.
		ERR :	Set to 'On', when error occurred after *function block* is executed.
		STATUS :	Detailed code value of error when error occurred.

■ Function and description

This is the *function block*, which is used to transmit array data of self station to variable defined as array type in other station. Data can be transmitted not to direct variable(Ex. : %I, %Q, and %M area) but to variable name used in other station. Variable name used should be specified in access variable list of other station, and the variable name, used for path name when access variable is specified, should be defined again as array type in global variable list(definition is not necessary in self station). The data type, which is the same as the array defined in other station, should be used(Max. 100 byte).

■ ST_NOH / ST_NOL

Station number of communication module in other station(For details, see RDTYPE *function block*).

■ VAR :

Variable identifier to transmit to other station(variable defined in other station).

■ SD_ARRAY

Array area of self station, which has the data to be transmitted to other station.
(The data type, which is the same as the array defined in other station, should be used)

■ NDR / ERR / STATUS

Displays execution result of *function block*(For details, see RDTYPE *function block*).

■ Example of program

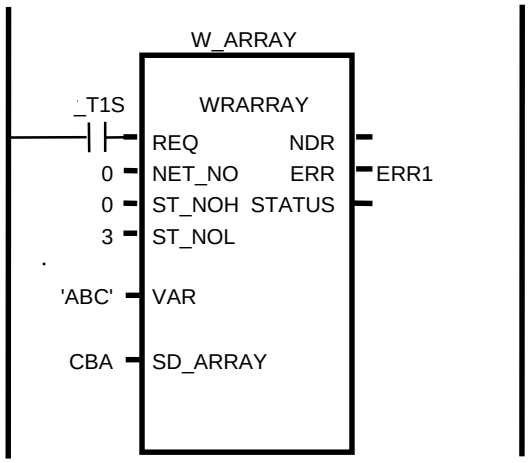
Suppose that Fnet is used and Fnet module is mounted in slot 0.

If station number of other station is station 3, and data of CBA array variable in self station is transmitted to variable of ABC defined as array in other station(Access for ABC variable in station 3 and register global variable and set REQ condition with a period of 1 second).

- Variable registration for station 3 (For registration method of access variable, see 6.3.8 Access variable registration).

Variable registration	Variable name	Access path	Description
Access variable registration	ABC	DEF(Ex.)	Set the path name of access variable ABC to DEF
Resource global variable registration	RES1.DEF	-	Register again DEF to global variable

LD



IL

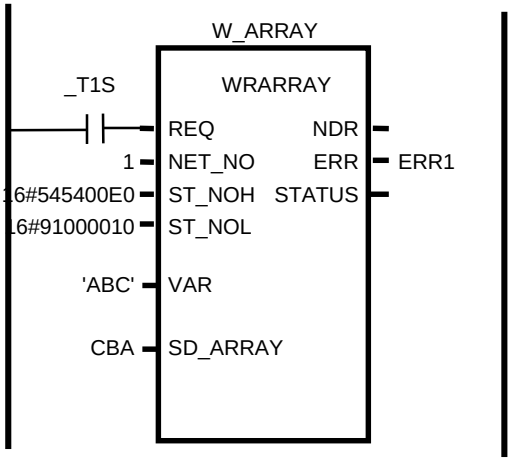
```
CAL   WRARRAY   W_ARRAY
      REQ :=      _T1S
      NET_NO :=   0
      ST_NOH :=   0
      ST_NOL :=   3
      VAR :=      'ABC'
      RD_ARRAY := CBA

LD    W_ARRAY.ERR
ST    ERR1
```

■ Example of program

Suppose that Mnet is used and communication is made with Mnet of self company product and Mini-MAP module mounted in slot 1.

If MAC address of other station is 16#00E091000010 and data of CBA array variable in self station is transmitted to variable of ABC defined as array in other station(Access for ABC variable of other station. Register global variable as the example of Fnet shown above, and set REQ condition with a period of 1 second).

LD



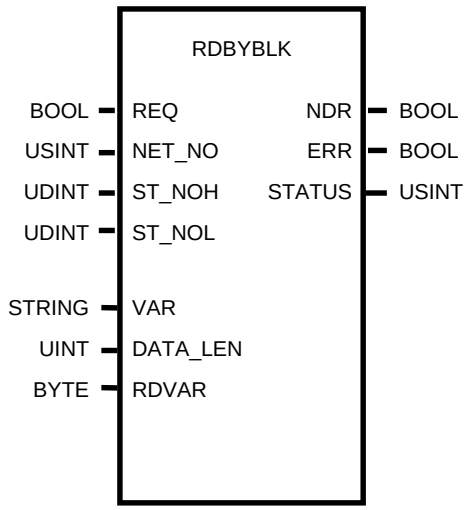
IL

```
CAL   WRARRAY   W_ARRAY
      REQ :=      _T1S
      NET_NO :=   1
      ST_NOH :=   16#545400E0
      ST_NOL :=   16#91000010
      VAR :=      'ABC'
      SD_ARRAY :=     CBA

LD    W_ARRAY.ERR
ST    ERR1
```

RDBLOCK

Reads continuous data from other station (Max. Fnet : 120 byte, Mnet : 400 byte)

Product name	GM1	GM2	GM3	GM4	GM5	GM6
Applicable	●	●	●	●	●	●

Function block	Description
	**Input** REQ : Request execution of *function block*(FB) at rising edge(0→1). NET_NO : Slot No.(0~7) that communication module of self station that this FB is to be transmitted is mounted. ST_NOH : Fixed to 0 in Fnet (In Mnet, upper station number and SAP of communication module mounted in other station). ST_NOL : Station number of communication module mounted in other station (In Mnet, lower station number). VAR : Start address of other station to read data(variable identifier is not available). RDVAR : Area that the data received from other station is to be stored. DATA_LEN : Number of data to be read. **Output** NDR : Set to 'On', when data is received without error. ERR : Set to 'On', when error occurred after *function block* is executed. STATUS : Detailed code value of error when error occurred.

■ Function and description

This is the *function block*, which is used to read massive data continuously from certain address of other station. Only byte should be used for data type, and direct address(%I, %Q, and %M) should be used for variable name.

■ ST_NOH / ST_NOL

Station number of communication module in other station(For details, see RDTYPE *function block*).

■ VAR

This is start address to read data from other station. Only direct address can be used, and only byte can be used for data type.

Ex.) %MB100 – From 100th byte area.
%IB0.2.1 – From first byte area among input area allocated in second slot(2) of main base(0).
%QB0.3.1 – From first byte area among input area allocated in third slot(3) of main base(0).

■ RDVAR

Area of self station that the data read from other station is to be stored(specified as byte).

■ DATA_LEN

Byte number of data to be read from other station.

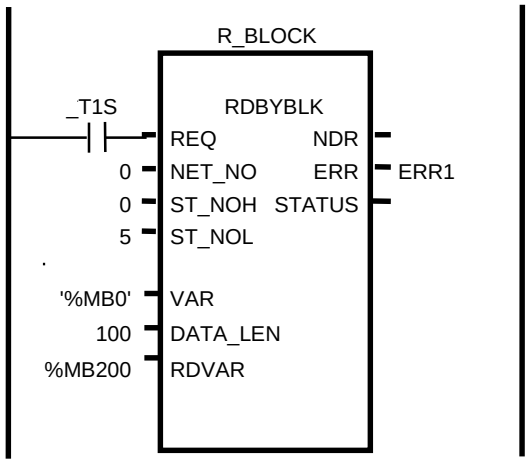
■ NDR / ERR / STATUS

This shows the execution result of *function block*(For details, see RDTYPE *function block*).

■ Example of program

Suppose that Fnet is used and Fnet module is mounted in slot 0.
If station number of other station is station 5, and 100 byte from area %MB0 of other station is read and it is stored in area from %MB200 to %MB299 of self station(REQ condition is set with a period of 1 second).

LD



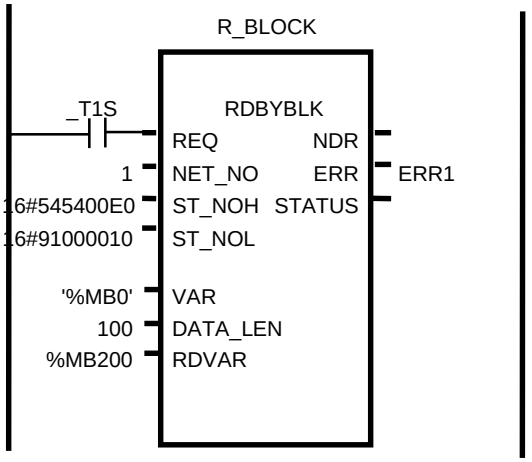
IL

```
CAL   RDBLOCK  R_BLOCK
      REQ :=       _T1S
      NET_NO :=   0
      ST_NOH :=   0
      ST_NOL :=   5
      VAR1 :=      '%MB0'
      RDVAR :=     %MB200
      DATA_LEN :=  100

LD        R_BLOCK.ERR
ST        ERR1
```

■ Example of program

Suppose that Mnet is used and communication is made with Mnet of self company's product and Mini-MAP module mounted in slot 1.
If MAC address of other station is 16#00E091000010, and 100 byte from area %MB0 of other station is read and it is stored to area from %MB200 to %MB299 of self station(REQ condition is set with a period of 1 second).

LD



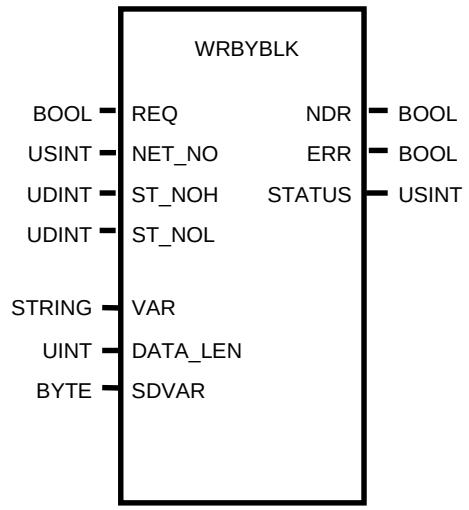
IL

```
CAL   RDBLOCK  R_BLOCK
        REQ :=       _T1S
        NET_NO :=  1
        ST_NOH :=  16#545400E0
        ST_NOL :=  16#91000010
        VAR1 :=      '%MB0'
        RDVAR :=   %MB200
        DATA_LEN :=  100

LD        R_BLOCK.ERR
ST        ERR1
```

WRBLOCK

Writes continuous data to other station (Max. Fnet : 120 byte, Mnet : 400 byte)

Product name	GM1	GM2	GM3	GM4	GM5	GM6
Applicable	●	●	●	●	●	●

Function block	Description
	**Input** REQ : Request execution of *function block*(FB) at rising edge(0→1). NET_NO : Slot No.(0~7) that communication module of self station that this FB is to be transmitted is mounted. ST_NOH : Fixed to 0 in Fnet (In Mnet, upper station number and SAP of communication module mounted in other station). ST_NOL : Station number of communication module mounted in other station (In Mnet, lower station number). VAR : Start address of other station to write data(variable identifier is not available). SDVAR : Area that the data to be transmitted to other station is stored. DATA_LEN : Number of data to transmit. **Output** NDR : Set to 'On', when data is transmitted without error. ERR : Set to 'On', when error occurred after *function block* is executed. STATUS : Detailed code value of error when error occurred.

■ Function and description

This is the *function block*, which is used to write massive data continuously to certain address of other station. Only byte should be used for data type, and direct address(%I, %Q, and %M) should be used for variable name.

■ ST_NOH / ST_NOL

Station number of communication module in other station(For details, see RDTYPE *function block*).

■ VAR

This is start address to write data to other station. Only direct address can be used, and only byte can be used for data type.

Ex.) %MB100 – From 100th byte area.
%IB0.2.1 – From first byte area among input area allocated in second slot(2) of main base(0).
%QB0.3.1 – From first byte area among input area allocated in third slot(3) of main base(0).

■ SDVAR

Area of self station that the data to be transmitted to other station is stored(specified as byte).

■ DATA_LEN

Byte number of data to transmit to other station.

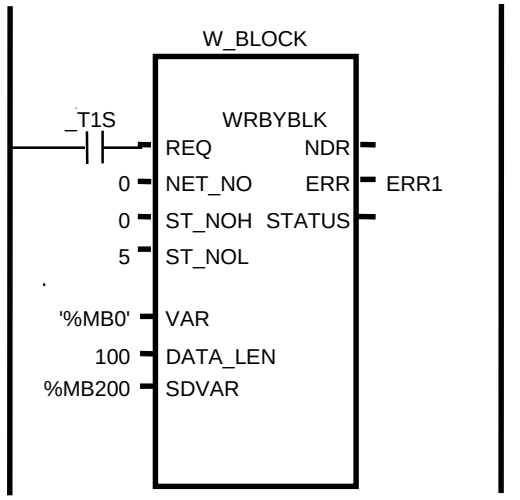
■ NDR / ERR / STATUS

This shows the execution result of *function block*(For details, see RDTYPE *function block*).

■ Example of program

Suppose that Fnet is used and Fnet module is mounted in slot 0.
If station number of other station is station 5, and data in the area from %MB200 to %MB299 of self station is transmitted to area from %MB0 to %MB99 of other station(REQ condition is set with a period of 1 second).

LD



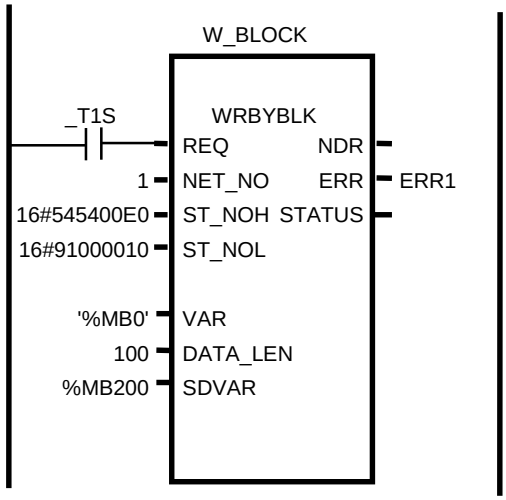
IL

```
CAL   WRBLOCK   W_BLOCK
          REQ :=          _T1S
          NET_NO :=       0
          ST_NOH :=       0
          ST_NOL :=       5
          VAR1 :=         '%MB0'
          SDVAR :=        %MB200
          DATA_LEN :=     100

LD                        W_BLOCK.ERR
ST                        ERR1
```

■ Example of program

Suppose that Mnet is used and communication is made with Mnet of self company's product and Mini-MAP module mounted in slot 1.
If MAC address of other station is 16#00E091000010, and data in the area from %MB200 to %MB299 of self station is transmitted to area from %MB0 to %MB99 of other station(REQ condition is set with a period of 1 second).

LD



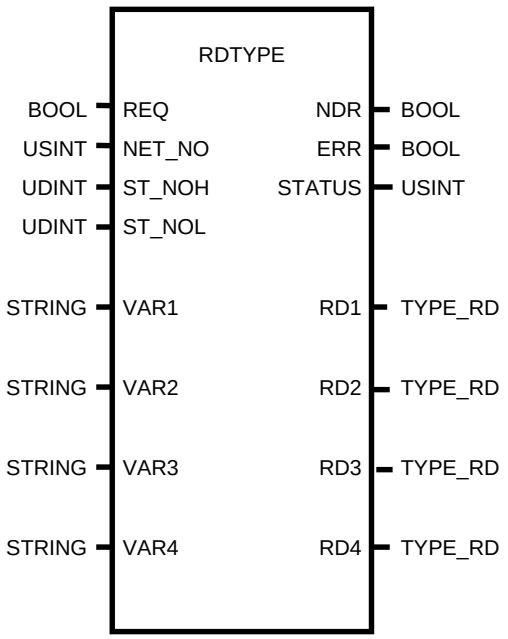
IL

```
CAL   WRBLOCK   W_BLOCK
          REQ :=          _T1S
          NET_NO :=       1
          ST_NOH :=       16#545400E0
          ST_NOL :=       16#91000010
          VAR1 :=         '%MB0'
          SDVAR :=        %MB200
          DATA_LEN :=     100

LD                        W_BLOCK.ERR
ST                        ERR1
```

RDTYPE(BOOL…DT)

Reads data from other station

Product name	GM1	GM2	GM3	GM4	GM5	GM6
Applicable	●	●	●	●	●	●

Function block	Description
	**Input** REQ : Request execution of *function block*(FB) at rising edge(0→1). NET_NO : Slot No.(0~7) that communication module of self station that this FB is to be transmitted is mounted. ST_NOH : Fixed to 0 in Fnet (In Mnet, upper station number and SAP of communication module mounted in other station). ST_NOL : Station number of communication module mounted in other station (In Mnet, lower station number). VAR1~4 : Direct address or variable identifier of other station to read data. **Output** NDR : Set to 'On', when data is received without error. ERR : Set to 'On', when error occurred after *function block* is executed. STATUS : Detailed code value of error when error occurred. RD1~4 : Area of self station that data received from other station is to be stored.

■ Function and description

This is the *function block*, which reads data of other station and stores it to designated location of self station.
Use appropriate *function block* according to the data type to be processed.
Ex.) Choose "RDWORD" in *function block* list for processing of word type data(16 bits).

■ ST_NOH / ST_NOL

Specifies upper/lower station number.

- Fnet : ST_NOH = 0 (Fixed), ST_NOL = other station number (For station 10, decimal uses10 and hexadecimal uses 16#A)
- Mnet(when Mini-MAP is used) : ST_NOH = SSAP(self station SAP) + DSAP(other station SAP) + Upper station number in communication module of other station
  ST_NOL = Lower station number of other station
- SAP(Service Access Point) : This is a factor, which determines the characteristic of service used in communication and connects each other according to the characteristic of upper application layer. 16#54, 16#58, 16#5C, 16#60, and 16#64 are possible for communication with self company's product(Mnet). 16#10 and 16#14 are provided for communication between Mnet of self company and Mini-MAP module of other company.

Ex.1) Communication with Mnet of self company
When data of station B is read at station A.
MAC address of station A : 16#00E091000000, MAC address of station B : 16#00E091000003.
ST_NOH : 16#54(self station SAP) 58(other station SAP) 00E0(upper station number of communication module in other station)
Namely, ST_NOH = 16#545800E0

ST_NOL = 16#91000003(lower station number of communication module in other station)

* MAC address is written at side part of product.

Ex.2) Communication with Mnet of other company

When data of station B(other company) is read at station A(self company) (Suppose that SAP = 4E for Mini-MAP module of other company).

MAC address of station A : 16#00E091000000(self company), MAC address of station B : 16#080070221C9A (other company).

ST_NOH : 16#10(self station SAP) 4E(other station SAP) 0800(upper station number of communication module in other station).

Namely, ST_NOH = 16#104E0800.

ST_NOL = 16#70221C9A(lower station number of communication module in other station)

■ VAR1 ~ VAR4

These are direct address or variable identifier of other station that data is to be read, and string is used for these. Data type should be identical with data type of *function block*(Ex. Data type of VAR1 ~ 4 should use word type when "RDWORD" *function block* is used).

- Fnet :

  Direct address : Directly reads other station area.

  Available data types are BOOL, BYTE, WORD, DWORD, LWORD(GM1/2).

  Ex.) When reading 100th bit area of memory in other station : '%MX100'.

  Ex.) When reading input 16 points of second slot(2) in main base(0) of other station : '%IW0.2.0'.

  Variable identifier : This is used to read the data of other station, and this should be registered in access variable area of other station and variable type should be identical with data type to be received(For registration method of access variable, see 6.3.8 Access variable registration)

- Mnet :

  Communication with self company Mnet :

  Using method of direct address and variable identifier is the same as Fnet.

  Communication with other company Mnet :

  For direct address, specify direct address used or supplied from other company(variable identifier is not available).

* **Enter string indication('') for location, which is not used in VAR1~VAR4.**

■ RD1 ~ RD4

This specifies self station area that received data from other station is to be stored.

VAR1 input data is stored in RD1, and VAR2, 3 and 4 are stored in each of RD2, 3, and 4.

Data type should be identical with data type of *function block*.

■ NDR

This is set to 'On' if *function block* is operated and normally finished, and set to 'Off' if appropriate scan is finished.

■ ERR

This is set to 'On' when error occurred after *function block* is operated, and maintains 'On' until next *function block* is operated. If error occurred, data is not received.

■ STATUS

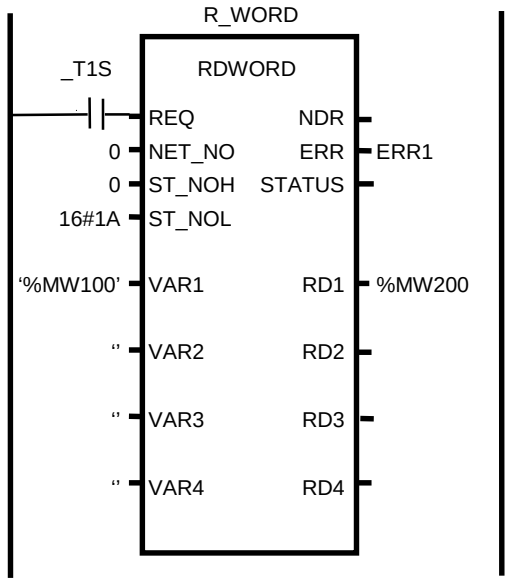
This shows detailed code value of error when error occurred after *function block* is operated, and maintains 'On' until next *function block* is operated(For code value, see page 6-55).

■ **Example of program**

Suppose that Fnet is used and Fnet module is mounted in slot 0.

If station number of other station is 16#1A, and area %MW100 of other station is read and it is stored in area %MW200 of self station(Use RDWORD *function block* and set REQ condition with a period of 1 second).

LD



IL

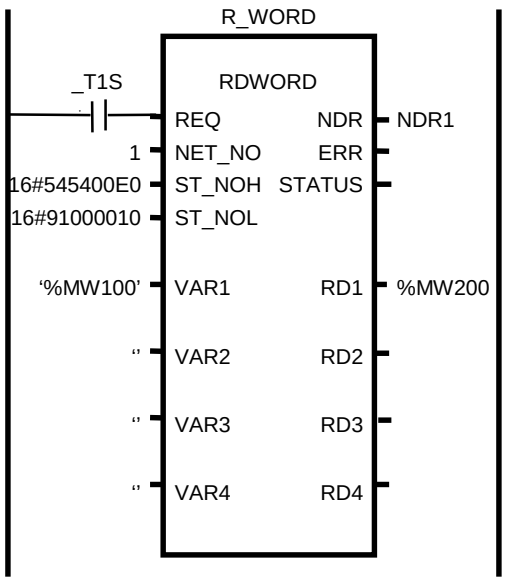
```
CAL   RDWORD     R_WORD
        REQ :=       _T1S
        NET_NO :=    0
        ST_NOH :=    0
        ST_NOL :=    16#1A
        VAR1 :=      '%MW100'
        VAR2 :=      ''
        VAR3 :=      ''
        VAR4 :=      ''

LD              R_WORD.RD1
ST              %MW200
LD              R_WORD.ERR
ST              ERR1
```

■ **Example of program**

Suppose that Mnet is used and communication is made with Mnet of self company product and Mini-MAP module mounted in slot 1.

If MAC address of other station is 16#00E091000010(written at side part of Mini-MAP module in other station), and area %MW100 of other station is read and it is stored in area %MW200 of self station(Use RDWORD *function block* and set REQ condition with a period of 1 second).

LD



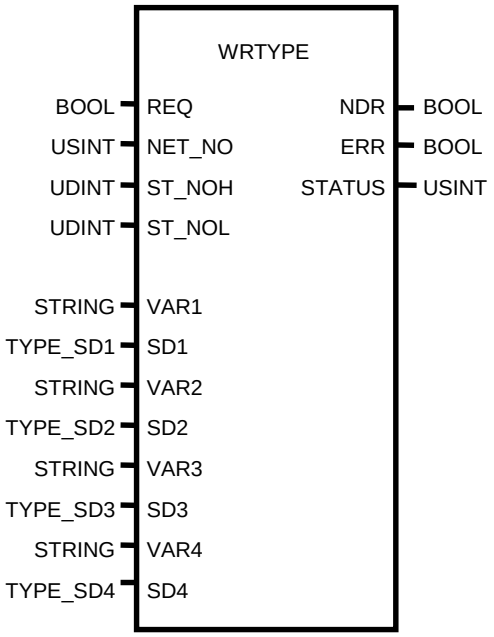
IL

```
CAL   RDWORD     R_WORD
        REQ :=       _T1S
        NET_NO :=    1
        ST_NOH :=    16#545400E0
        ST_NOL :=    16#91000010
        VAR1 :=      '%MW100'
        VAR2 :=      ''
        VAR3 :=      ''
        VAR4 :=      ''

LD              R_WORD.RD1
ST              %MW200
LD              R_WORD.NDR
ST              NDR1
```

WRTYPE(BOOL…DT)

Writes data to other station

Product name	GM1	GM2	GM3	GM4	GM5	GM6
Applicable	●	●	●	●	●	●

Function block	Description
	**Input** REQ : Request execution of *function block*(FB) at rising edge(0→1). NET_NO : Slot No.(0~7) that communication module of self station that this FB is to be transmitted is mounted. ST_NOH : Fixed to 0 in Fnet (In Mnet, upper station number and SAP of communication module mounted in other station). ST_NOL : Station number of communication module mounted in other station (In Mnet, lower station number). VAR1~4 : Direct address or variable identifier of other station to write data. SD1~4 : Data to be transmitted to other station or area of self station. **Output** NDR : Set to 'On', when data is transmitted without error. ERR : Set to 'On', when error occurred after *function block* is executed. STATUS: Detailed code value of error when error occurred.

■ Function and description

This is the *function block*, which transmits area of self station or specific data to other station through communication module of self station.

Ex.) Choose "WRBYTE" in *function block* list for processing of byte type data(8 bits).
For detailed contents about I/O, see "RDTYPE" *function block*.

■ ST_NOH / ST_NOL

Specifies upper/lower station number.

■ VAR1 ~ VAR4

These are direct address or variable identifier of other station that data is transmitted, and string is used for these.

Data type should be identical with data type of *function block*(For example, data type of VAR1 ~ 4 should use byte type when direct address is used in "WRBYTE" *function block*. It is set automatically when variable identifier is used).

*** Enter string indication('') for the location, which is not used in VAR1~VAR4.**

■ SD1 ~ SD4

This specifies the value to be transmitted to other station or self station area.

Data specified in SD1 is transmitted to other station area specified in VAR1. Data of SD2, 3, and 4 are transmitted to each area of R2, 3, and 4.

Data type should be identical with data type of *function block*.

*** Enter '0' for the location, which is not used in SD1~SD4.**

■ NDR / ERR / STATUS :

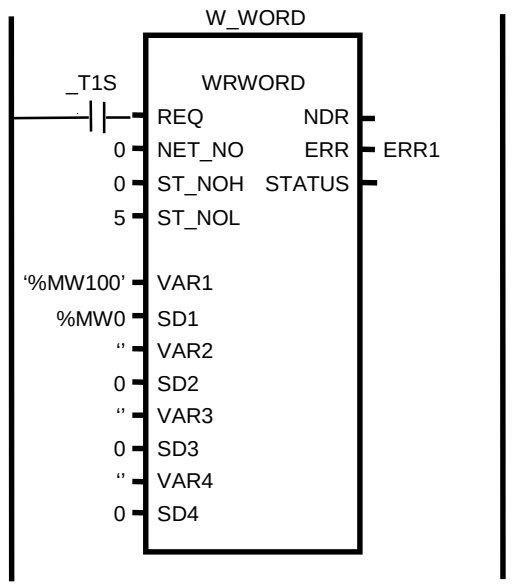
See "RDTYPE" *function block*.

■ **Example of program**

Suppose that Fnet is used and Fnet module is mounted in slot 0.

If data in self station area of %MW0 is transmitted to area %MW100 of other station 5 (Use WRWORD *function block* and set REQ condition with a period of 1 second).

LD



IL

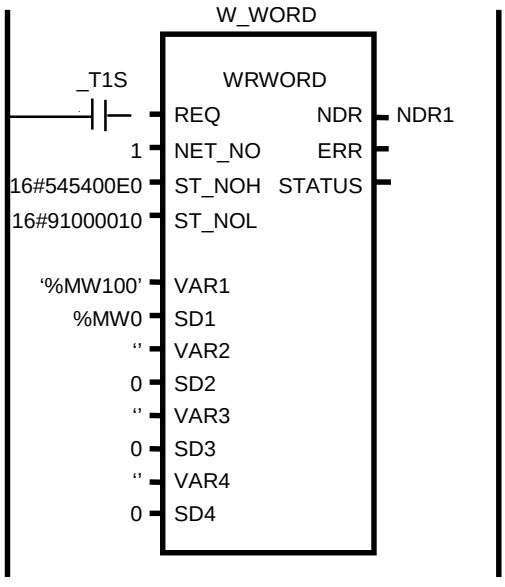
```
CAL   WRWORD    W_WORD
        REQ :=      _T1S
        NET_NO :=   0
        ST_NOH :=   0
        ST_NOL :=   5
        VAR1 :=     '%MW100'
        SD1 :=      %MW0
        VAR2 :=     ''
        SD2 :=      0
        VAR3 :=     ''
        SD 3:=      0
        VAR4 :=     ''
        SD4 :=      0

LD              W_WORD.ERR
ST              ERR1
```

■ **Example of program**

Suppose that Mnet is used and communication is made with Mnet of self company product and Mini-MAP module mounted in slot 1.

If MAC address of other station is 16#00E091000010(written at side part of Mini-MAP module in other station), and area %MW0 of self station is read and it is transmitted in area %MW100 of other station(Use WRWORD *function block* and set REQ condition with a period of 1 second).

LD



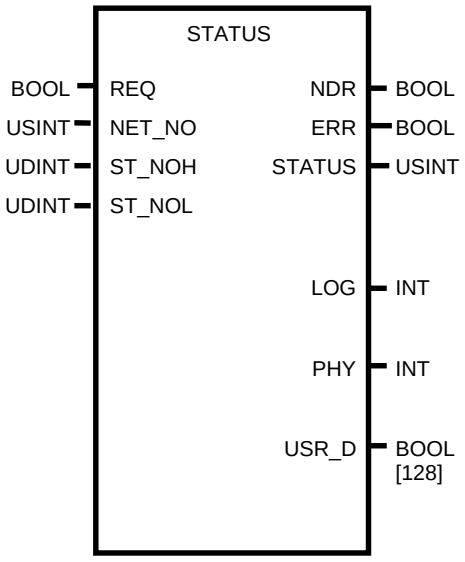
IL

```
CAL   WRWORD    W_WORD
        REQ :=      _T1S
        NET_NO :=   0
        ST_NOH :=   16#545400E0
        ST_NOL :=   16#91000010
        VAR1 :=     '%MW100'
        SD1 :=      %MW0
        VAR2 :=     ''
        SD2 :=      0
        VAR3 :=     ''
        SD 3:=      0
        VAR4 :=     ''
        SD4 :=      0

LD              W_WORD.NDR
ST              NDR1
```

STATUS

Reads status of other station

Product name	GM1	GM2	GM3	GM4	GM5	GM6
Applicable	●	●	●	●	●	●

Function block	Description
	**Input** REQ : Request execution of *function block*(FB) at rising edge(0→1). NET_NO : Slot No.(0~7) that communication module of self station that is mounted. ST_NOH : Fixed to 0 in Fnet (In Mnet, upper station number and SAP of communication module mounted in other station). ST_NOL : Station number of communication module mounted in other station (In Mnet, lower station number). **Output** NDR : Set to 'On', when data is transmitted without error. ERR : Set to 'On', when error occurred after *function block* is executed. STATUS : Detailed code value of error when error occurred. LOG : Level of function, which is usable with communication service. This is always set to '0'. PHY : H/W operation condition of other PLC. USR_D : Integrated data of PLC condition.

■ Function and description

This is the *function block*, which is used to check the status of other station.

■ ST_NOH / ST_NOL

Station number of other station communication module(For details, see RDTYPE *function block*).

■ LOG

This shows the level of function, which is usable with communication service(Logical state).

0 = STATE-CHANGE-ALLOWED.

■ PHY

This is physical state, and shows H/W operation status of PLC.

0 = OPERATIONAL(In operating)

1 = PARTIALLY-OPERATIONAL-H/W(Not all of PLC and peripheral modules are normally operating).

2 = INOPERABLE-H/W(Operation is stopped by error occurrence).

3 = NEED-COMMISSION-H/W(In operating, but data is not reliable).

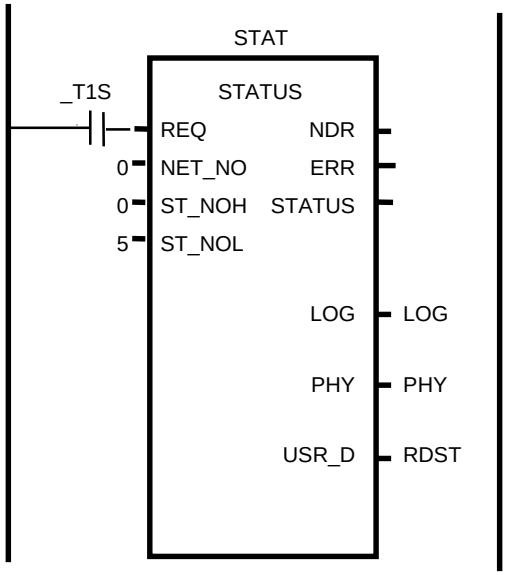
■ USR_D

This shows integrated status of PLC, and supplies 128 bit of bit array to control and monitor the system.

■ **Example of program**

Suppose that Fnet is used and Fnet module is mounted in slot 0.
If integrated information for other station 5 is read with RDST variable defined as array 128 bit.
(REQ condition is set with a period of 1 second).

LD



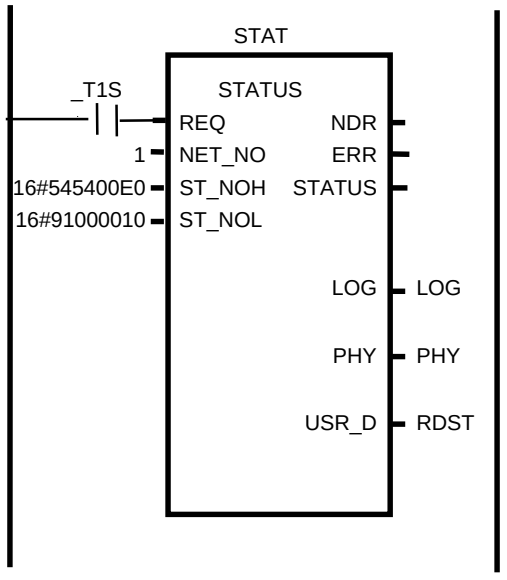
IL

```
CAL   STATUS      STAT
        REQ :=        _T1S
        NET_NO :=     0
        ST_NOH :=     0
        ST_NOL :=     5

LD                STAT.LOG
ST                LOG
LD                STAT.PHY
ST                PHY
LD                STAT.USR_D
ST                RDST
```

■ **Example of program**

Suppose that Mnet is used and communication is made with Mnet of self company's product and Mini-MAP module mounted in slot 1.
If integrated information of Mini-MAP module that MAC address of other station is 16#00E091000010(written at side part of Mini-MAP module in other station) is read with RDST variable defined as array 128 bit.
(REQ condition is set with a period of 1 second).

LD



IL

```
CAL   STATUS      STAT
        REQ :=        _T1S
        NET_NO :=     1
        ST_NOH :=     16#545400E0
        ST_NOL :=     16#91000010
LD                STAT.LOG
ST                LOG
LD                STAT.PHY
ST                PHY
LD                STAT.USR_D
ST                USR_D
```