Chapter 10 Communication function block libraries

10. Communication function block libraries

10.1. Communication function block libraries

Each communication function block library is described in this section.

Point Below function block is supported according to PLC type. (O : Available, X : Not available)

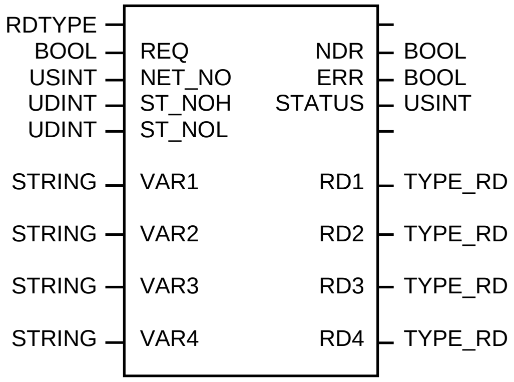
Product Data type(Bit size)		GM1 GM2	GM3	GM4	GM5	Function block name (RD / WR)
Basic type (Max. 4)	BOOL(1)	O	O	O	O	RD(WR) BOOL
	BYTE(8)	O	O	O	O	RD(WR)BYTE
	WORD(16)	O	O	O	O	RD(WR)WORD
	DWORD(32)	O	O	O	O	RD(WR)DWORD
	LWORD(64)	O	X	X	X	RD(WR)LWORD
	USINT(8)	O	O	O	O	RD(WR)USINT
	UINT(16)	O	O	O	O	RD(WR)UINT
	UDINT(32)	O	O	O	O	RD(WR)UDINT
	ULINT(64)	O	X	X	X	RD(WR)ULINT
	SINT(8)	O	O	O	O	RD(WR)SINT
	INT(16)	O	O	O	O	RD(WR)INT
	DINT(32)	O	O	O	O	RD(WR)DINT
	LINT(64)	O	X	X	X	RD(WR)LINT
	REAL(32)	O	X	X	X	RD(WR)REAL
	LREAL(64)	O	X	X	X	RD(WR)LREAL
	TIME(16)	O	O	O	O	RD(WR)TIME
	DATE(48)	O	O	O	O	RD(WR)DATE
	TIME of DAY(48)	O	O	O	O	RD(WR)TOD
	DATE and TIME (48)	O	O	O	O	RD(WR)DT
Block (Max Fnet:120, Mnet:400 byte)		O	O	O	O	RD(WR) block

Product Data type(bit size)		GM1 GM2	GM3	GM4	GM5	Function block name (RD / WR)
Array Type (Within Max. 100 byte)	BOOL	O	O	O	O	RD(WR) Array
	BYTE	O	O	O	O	RD(WR)Array
	WORD	O	O	O	O	RD(WR) Array
	DWORD	O	O	O	O	RD(WR) Array
	LWORD	O	X	X	X	RD(WR) Array
	USINT	O	O	O	O	RD(WR) Array
	UINT	O	O	O	O	RD(WR) Array
	UDINT	O	O	O	O	RD(WR) Array
	ULINT	O	X	X	X	RD(WR) Array
	SINT	O	O	O	O	RD(WR) Array
	INT	O	O	O	O	RD(WR) Array
	DINT	O	O	O	O	RD(WR) Array
	LINT	O	X	X	X	RD(WR) Array
	REAL	O	X	X	X	RD(WR) Array
	LREAL	O	X	X	X	RD(WR) Array
	TIME	O	O	O	O	RD(WR) Array
	DATE	O	O	O	O	RD(WR) Array
	TIME of DAY	O	O	O	O	RD(WR) Array
	DATE and TIME	O	O	O	O	RD(WR) Array

RDTYPE(BOOL...DT)

Read data from another station

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

Function	Description
 RDTYPE BOOL USINT UDINT UDINT STRING STRING STRING STRING REQ NET_NO ST_NOH ST_NOL VAR1 VAR2 VAR3 VAR4 NDR ERR STATUS RD1 RD2 RD3 RD4 BOOL BOOL USINT TYPE_RD TYPE_RD TYPE_RD TYPE_RD	**Input** REQ : Function block execution request at rising edge(0→1) NET_NO : Slot number installing the communication module for F/B transfer(0 ~ 7) ST_NOH : Fix to 0 during using Fnet (SAP with upper number of station communication module in remote station during using Mnet) ST_NOL : Station number of communication module installed in another station(Lower station number of another station during using Mnet) VAR1 – 4 : Direct address or variable identifier to read the data **Output** NDR : On during receiving the data without error ERR : On when the error occurs after executing function block STATUS : Detailed code value for the error RD1 – 4 : Station area to store the received data from another station

■ Function and description

Read the data of another station through the communication module and store the data to specified location.
Use respective function block according to the data type to be processed.
Ex) Select "RDWORD" in function block list to process WORD(16Bit) type data.

■ ST_NOH/ST_NOL

Assign the station number of communication module.

- Fnet : ST_NOH=0(Fix), ST_NOL=Station number of another station(For example, Station 10 is 10 by decimal and 16#A by hexadecimal)
- Mnet(For Mini-MAP) : ST_NOH=SSAP(Station SAP)+DSAP(Another station SAP)+Upper station number of another communication module ST_NOL=Lower station number of another communication module

* SAP(Service Access Point) : Factor to define the service characteristics and connect the service according to upper application layer. 16#54,16#58, 16#5C,16#60 and 16#64 are supplied for the communication with GLOFA Mnet and 16#10 and 16#14 are for the communication between other company's Mini-Map module and Mnet.

Ex 1) Communication with GLOFA Mnet

When A station reads the data of B station(remote station),

MAC address of A station : 16#00E091000000 , MAC address of B station : 16#00E091000003(Upper station No.: 00E0, Lower station No.: 91000003)

ST_NOH : 16#54 (GLOFA SAP) 58 (Another station SAP) 00E0 (Another station communication module's upper station number)

Therefore, ST_NOH=16#545800E0

ST_NOL=16#91000003 (Another station communication module's upper station number)

* MAC address is marked at the side of product.

Ex 2) Communication with other Mnet
When A station reads the data of B station(remote station)(providing Mini-Map module SAP=4E)
MAC address of A station : 16#00E091000000, Mac address of B station : 16#080070221C9A
ST_NOH : 16#10 (GLOFA SAP) 4E (Another station SAP) 0800 (Another station communication module's upper station number)
Therefore, ST_NOH=16#104E0800
ST_NOL=16#70221C9A (Another station communication module's upper station number)

■ VAR1 - VAR 4

Direct address or variable identifier of remote station to read the data.(Marked by STRING)
The data type shall be in accordance with the function block data type.(For example, VAR1 - VAR 4 data type uses WORD for "RDWORD" function block)

- Fnet :
  Direct address : Read the area of another station directly.
  BOOL, BYTE, WORD, DWORD and LWORD(GM1/2 only) are available.
  Ex 1) When read 100th bit area of remote station memory : '%MX100'
  Ex 2) When read input 16 point of second slot(2) of another station main base(0) : '%IW0.2.0'
  Variable identifier : The variable defined by another station(Access and resource global variable registration defining what data type process the variable identifier and where the data is assigned to) shall be used as the variable identifier to read the data of another station.
  Ex) 'PLC_1'

- Mnet
  Communication with GLOFA Mnet:
  The operation method of direct address and variable identifier is same as that of Fnet.
  Communication with other Mnet:
  Assign direct address used or supplied by other Mnet.
  (Variable identifier is not supplied.)

* Input string mark(") to the position that does not used in VAR1 ~VAR4.

■ RD1 - RD4

Assign the area to store received data from another station.
VAR1 input data is stored to RD1 and VAR2,3,4 are stored to respective RD2,3,4.
Data type shall be in accordance with that of function block.

■ NDR

Switch ON when the data are received normally after operating the function block and hold ON till the block is restarted.

■ ERR

Switch ON when the error occurs after operating the function block and hold ON till the block is restarted at next scanning. When the error occurs, the data is not received.

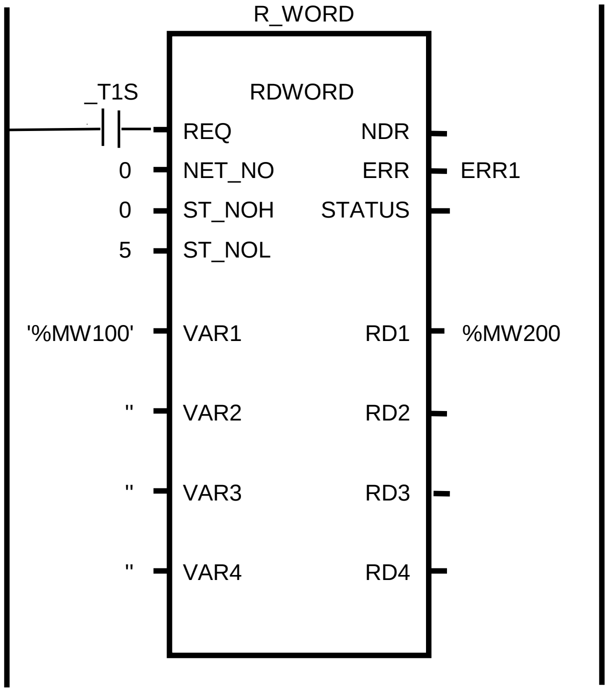
■ STATUS

Indicate the detailed code value on the error ON when the error occurs after operating the function block and hold ON till the block is restarted at next scanning. (Refer to 10-20 for code value)

■ Program example: **Providing that Fnet module is installed at 0 slot)**

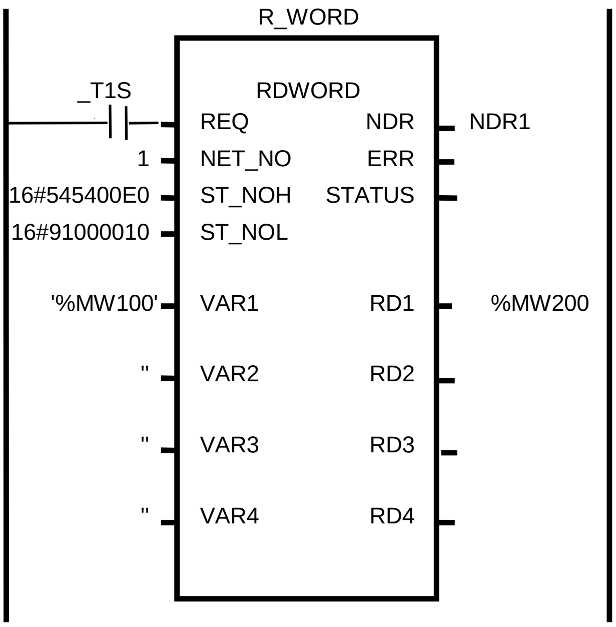
In case that prefix number of another station is 5 and read and store %MW100 of another station area to %MW200 of local station.
(Use RDWORD function block and preset REQ condition every second)

LD	IL

 R_WORD
_T1S
RDWORD
REQ
NDR
0
NET_NO
ERR
ERR1
0
ST_NOH
STATUS
5
ST_NOL
'%MW100'
VAR1
RD1
%MW200
''
VAR2
RD2
''
VAR3
RD3
''
VAR4
RD4
	CAL RDWORD R_WORD
	REQ := _T1S
	NET_NO := 0
	ST_NOH := 0
	ST_NOL := 5
	VAR1 := '%MW100'
	VAR2 := ''
	VAR3 := ''
	VAR4 := ''
	LD R_WORD.ERR
	ST ERR1
	LD R_WORD.RD1
	ST %MW200

■ **Program example : Providing that GLOFA Mini-MAP module is installed at No.1 slot.**
If MAC address of remote station is 16#00E091000010(marked at the side of Mini-MAP module) and read and store %MW100 of remote station area to %MW200 of local station.
(Use RDWORD function block and preset REQ condition every second)

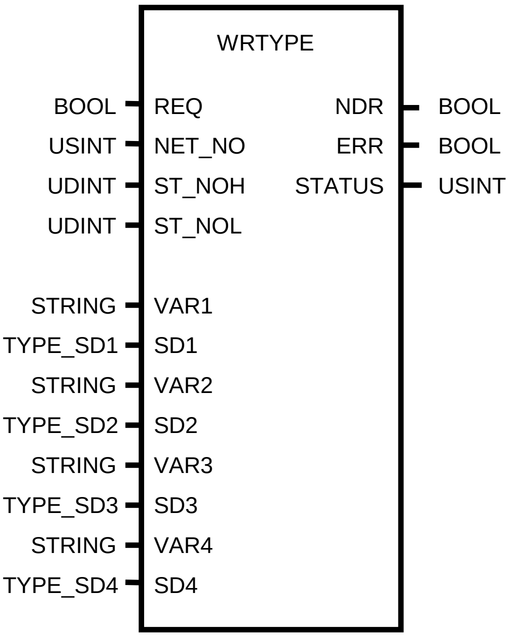
LD	IL

 R_WORD
_T1S
RDWORD
REQ
NDR
NDR1
1
NET_NO
ERR
16#545400E0
ST_NOH
STATUS
16#91000010
ST_NOL
'%MW100'
VAR1
RD1
%MW200
''
VAR2
RD2
''
VAR3
RD3
''
VAR4
RD4
	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)

Write the data to remote station

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

Function	Description
 WRTYPE BOOL REQ USINT NET_NO UDINT ST_NOH UDINT ST_NOL STRING VAR1 TYPE_SD1 SD1 STRING VAR2 TYPE_SD2 SD2 STRING VAR3 TYPE_SD3 SD3 STRING VAR4 TYPE_SD4 SD4 NDR BOOL ERR BOOL STATUS USINT	**Input** REQ: Function block execution request at rising edge(0→1) NET_NO: Slot number installing the communication module for F/B transfer(0 ~ 7) ST_NOH: Fix to 0 during using Fnet (with upper station number of communication module in remote station during using Mnet) ST_NOL: Station number of communication module installed in another station(Lower prefix module of remote station during using Mnet) VAR1 - 4: Direct address or variable identifier to read the data SD1-4: Data or master area which shall send to another station. **Output** NDR: On during receiving the data without error ERR: On when the error occurs after executing function block STATUS: Detailed code value for the error

■ Function and description

Transfer the data of station or specific data to station through the communication module of station.
Use respective function module according to the data type to be processed.
Ex) Select "WRBYTE" in function block list to process BYTE(8Bit) type data.
(Refer to "RDTYPE" function block for detailed input and output.)

■ ST_NOH/ST_NOL description

Assign upper and lower station number(station number).

■ VAR1 - VAR 4

Direct address or variable identifier of another station to transfer the data.(Marked by STRING)
The data type shall be in accordance with the function block data type.(For example, VAR1 - VAR 4 data type shall use BYTE to use direct address in "WRBYTE" function block.)
When variable identifier is used, data type is automatically set and variable name defined by remote station(Access and resource global variable registration defining what data type process the variable identifier and where the data is assigned to) shall be used.

■ SD1 - SD4

Set the numeric to transfer to the remote station and the area of station. The data set to SD1 is transferred to the area of another station set to VAR1 and SD2,3,4 are transferred to respective VAR2,3,4.
Data type shall be in accordance with that of function block.

■ NDR / ERR / STATUS

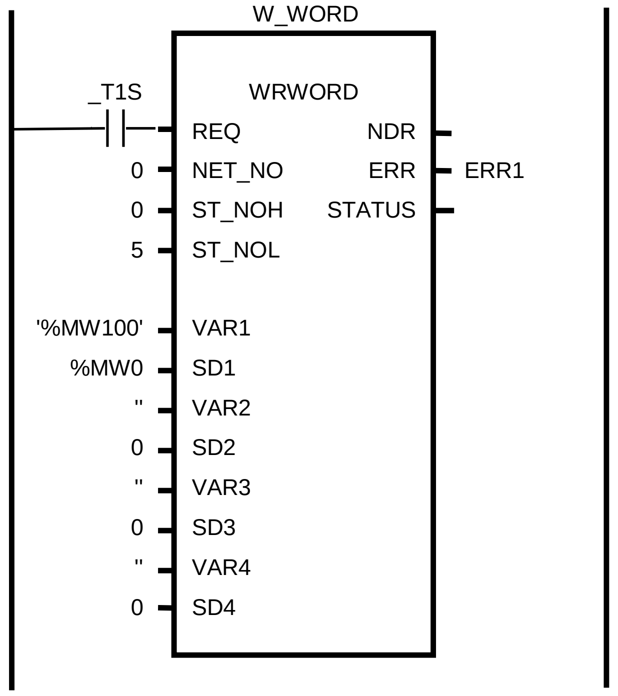
Refer to "RDTYPE" function block.

■ **Program example : Providing that Fnet module is installed at No.0 slot.**

If %MW0 data of master station area is transferred to %MW100 of another 5 station.

(Use WRWORD function block and preset REQ condition every second)

LD



IL

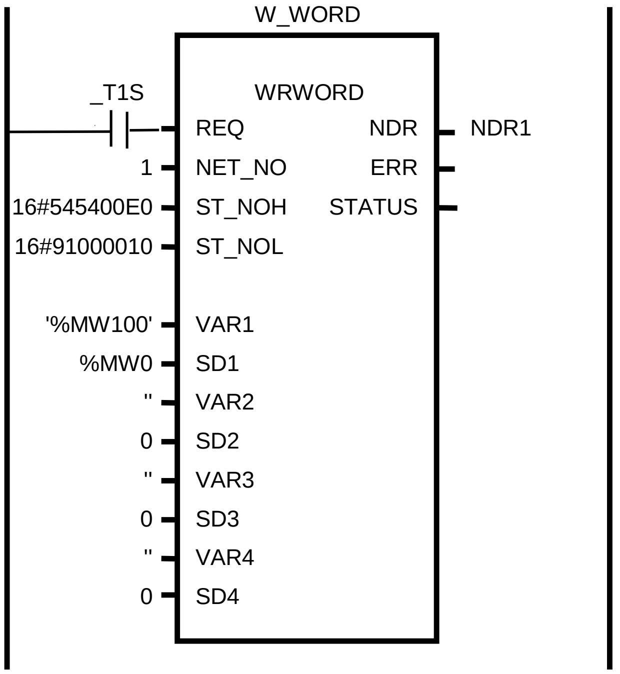
```
CAL       WRWORD       W_WORD
          REQ :=       _T1S
          NET_NO :=    0
          ST_NOH :=    0
          ST_NOL :=    5
          VAR1 :=      '%MW100'
          SD1:=        %MW0
          VAR2 :=      ''
          SD2:=        0
          VAR3 :=      ''
          SD3:=        0
          VAR4 :=      ''
          SD4:=        0
LD                     W_WORD.ERR
ST                     ERR1
```

■ **Program example : Providing that Mnet module is installed to No.1 slot.**

If MAC address of another station is 16#00E091000010 and read and transfer %MW0 of station area to %MW100 of another station.

(Use WRWORD function block and preset REQ condition every second)

LD



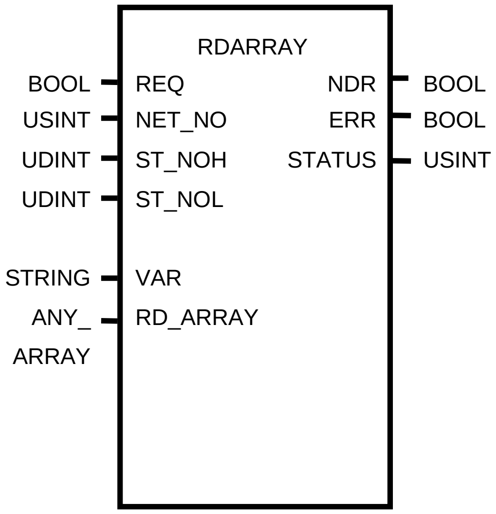
IL

```
CAL       WRWORD       W_WORD
          REQ :=       _T1S
          NET_NO :=    1
          ST_NOH :=    16#545400E0
          ST_NOL :=    16#91000010
          VAR1 :=      '%MW100'
          SD1:=        %MW0
          VAR2 :=      ''
          SD2:=        0
          VAR3 :=      ''
          SD3:=        0
          VAR4 :=      ''
          SD4:=        0
LD                     W_WORD.NDR
ST                     NDR1
```

RDARRAY

Read DATA of array type from another station

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

Function	Description
 RDARRAY BOOL REQ USINT NET_NO UDINT ST_NOH UDINT ST_NOL STRING VAR ANY_ ARRAY RD_ARRAY NDR BOOL ERR BOOL STATUS USINT	**Input** REQ: Function block execution request at rising edge(0→1) NET_NO: Slot number installing the communication module for F/B transfer(0 ~ 7) ST_NOH: Fix to 0 during using Fnet (SAP with upper station number of communication module in remote station during using Mnet) ST_NOL: Station number of communication module installed in remote station(Lower station module of remote station during using Mnet) VAR: Variable identifier(defined by another station) to read the data(can not use direct address) RD_ARRAY: Local station area to store ARRAY data received from another station **Output** NDR: On during receiving the data without error ERR: On when the error occurs after executing function block STATUS: Detailed code value for the error

■ Function and description

Function block to read the data of ARRAY type from another station.

RDARRAY can not use the direct variable of another station but can read the data by the variable name of another station. The variable name shall be assigned in the global variable list as array type. The data type shall be same to the array defined by remote station.

■ ST_NOH/ST_NOL

Communication module station number of another station(Refer to RDTYPE function block for details)

■ VAR :

Variable identifier to read from another station. Use the variable name defined by another station(Access and resource global variable registration defining what data type process the variable identifier and where the data is assigned to).

■ NDR / ERR / STATUS

Display the result of function block(Refer to RDTYPE function block for details).

■ RDARRAY

Station array area to store ARRAY data received from remote station.

The data type shall be same to the array defined by remote station.

■ **Program example : Providing that Fnet module is installed to No.0 slot.**

If the station number of another station is 3 and read ABC variable defined to ARRAY by another station and store it to CBA of local station array variable.

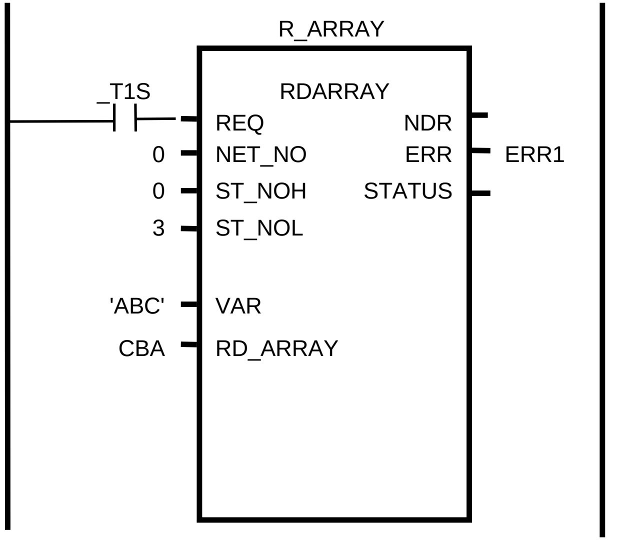
(Register the access and global variable for ABC variable at 3rd station and preset REQ condition every second)

- Variable registration example at 3rd station

Variable registration	Variable name	Access route	Description
Access variable registration	ABC	DEF(Example)	Set the access variable ABC to DEF
Resource global variable registration	RES1.DEF	-	Register DEF to global variable again(use resource 1).

* When the route name is assigned to direct variable(%I,%Q,%M) during access variable registration, resource global variable registration is not required.

LD



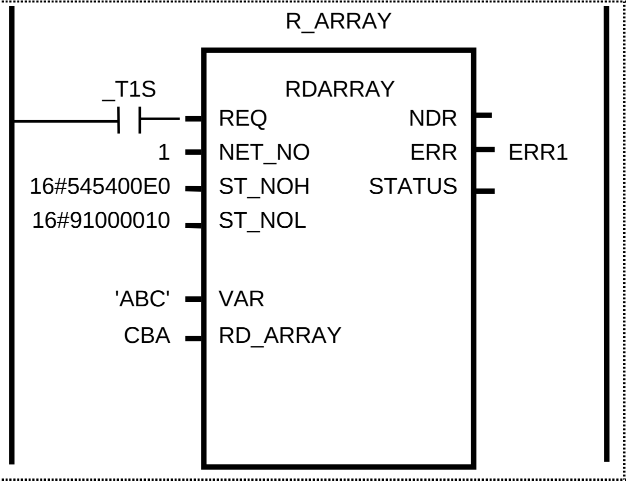
IL

```
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
```

■ **Program example : Providing that Mnet module is installed to No.1 slot.**

If MAC address of another station is 16#00E091000010 and read ABC variable defined as array by remote station and store it to CBA variable of station array.(Register the access and global variables for remote station ABC variable as Fnet example and preset REQ condition every second)

LD



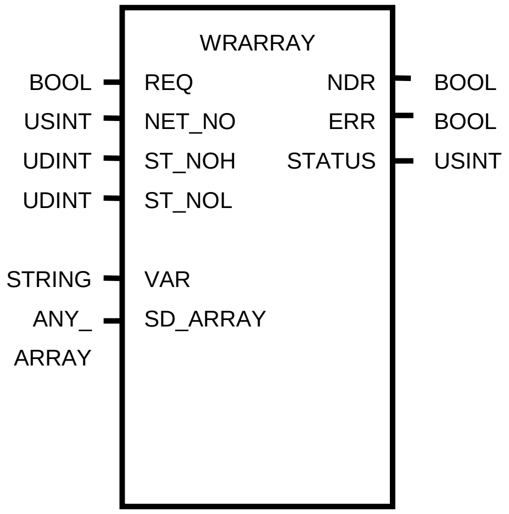
IL

```
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

Write DATA of ARRAY type to another station

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

Function	Description
 WRARRAY BOOL REQ NDR BOOL USINT NET_NO ERR BOOL UDINT ST_NOH STATUS USINT UDINT ST_NOL STRING VAR ANY_ ARRAY SD_ARRAY	**Input** REQ: Function block execution request at rising edge(0→1) NET_NO: Slot number installing the communication module for F/B transfer(0 ~ 7) ST_NOH: Fix to 0 during using Fnet (SAP, upper prefix number of communication module in another station during using Mnet) ST_NOL: Station number of communication module installed in remote station(Lower station module of another station during using Mnet) VAR: Variable identifier(defined by another station) to read the data(can not use direct address) SD_ARRAY: Local station area to store ARRAY data received from remote station **Output** NDR: On during receiving the data without error ERR: On when the error occurs after executing function block STATUS: Detailed code value for the error

■ Function and description

Function block to transfer ARRAY data of station to the variable defined as ARRAY type by remote station. WRARRAY can not use the direct variable(e.g., %I, %Q, %M area) but can transfer the data of variable identifier used in another station. The variable name shall be assigned in the global variable list and defined as array type. The data type shall be same to the array defined by another station.(within total 100 byte)

■ ST_NOH/ST_NOL

Communication module station number of another station(Refer to RDTYPE function block for details)

■ VAR :

Variable identifier to transfer to another station. Use the variable name defined by another station(Access and resource global variable registration defining what data type process the variable identifier and where the data is assigned to).

■ SD_RDARRAY

Station array area having the data to transfer to another station.
(The data type shall be same to the array defined by another station.)

■ NDR / ERR / STATUS

Display the result of function block(Refer to RDTYPE function block for details).

■ **Program example : Providing that Fnet module is installed to No.0 slot.**

If the prefix number of another station is 3 and read ABC variable defined to ARRAY by another station and store it to CBA of station array variable.

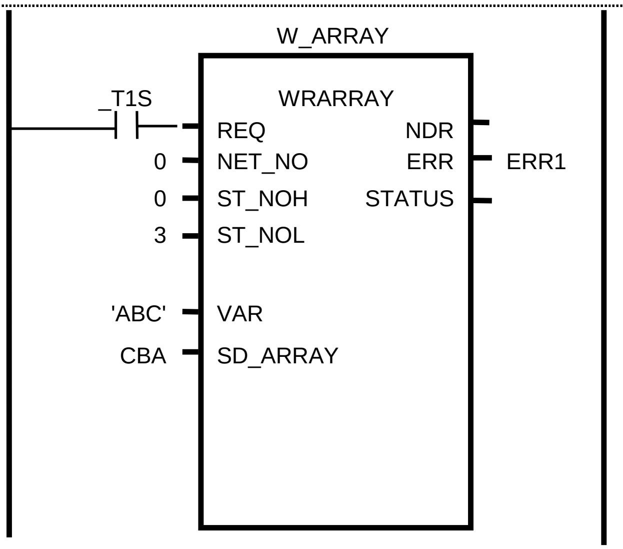
(Register the access and global variable for ABC variable at 3rd station and preset REQ condition every second)

- Variable registration example at 3rd station

Variable registration	Variable name	Access route	Description
Access variable registration	ABC	DEF(Example)	Set the access route name of access variable ABC to DEF
Resource global variable registration	RES1.DEF	-	Register DEF to global variable again (use resource 1).

* When the route name is assigned to direct variable(%I,%Q,%M) during access variable registration, resource global variable registration is not required.

LD



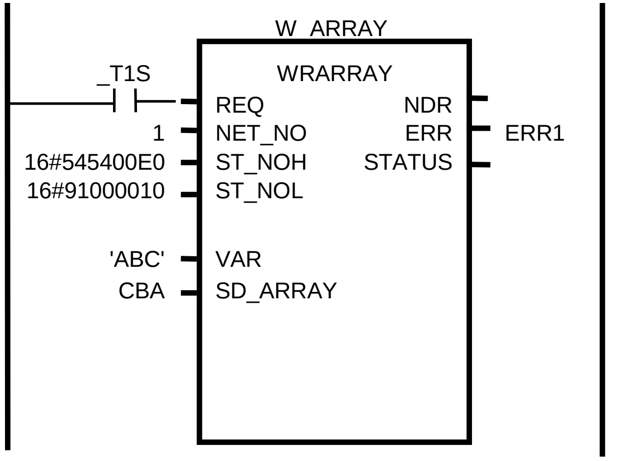
IL

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

LD                    W_ARRAY.ERR
ST                    ERR1
```

■ **Program example : Providing that Mnet module is installed to No.1 slot.**

If MAC address of another station is 16#00E091000010 and transfer CBA variable of local station array to ABC variable defined as array by another station(Register the access and global variables for another station ABC variable as Fnet example and preset REQ condition every second)

LD



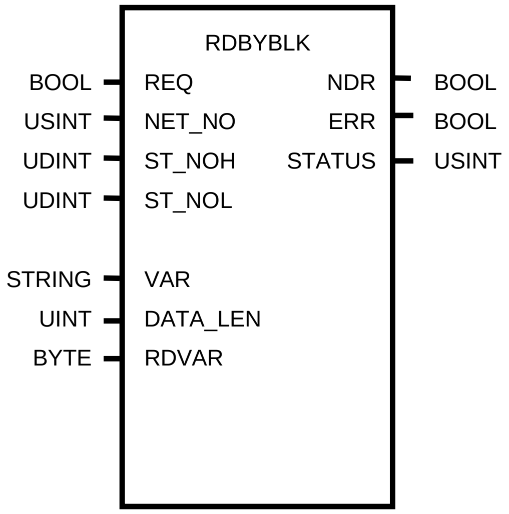
IL

```
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
```

RDBYBLK

Read continuous data from station
(Max. Fnet:120Byte, Mnet:400Byte)

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

Function	Description
 RDBYBLK BOOL REQ NDR BOOL USINT NET_NO ERR BOOL UDINT ST_NOH STATUS USINT UDINT ST_NOL STRING VAR UINT DATA_LEN BYTE RDVAR	**Input** REQ: Function block execution request at rising edge(0→1) NET_NO: Slot number installing the communication module for F/B transfer(0 ~ 7) ST_NOH: Fix to 0 during using Fnet (SAP, upper station number of communication module in remote station during using Mnet) ST_NOL: Station number of communication module installed in another station(Lower station module of another station during using Mnet) VAR: Variable identifier(defined by another station) to read the data(can not use direct address) RDVAR: Local station block area to store the data received from another station DATA_LEN: Data to be read **Output** NDR: On during receiving the data without error ERR: On when the error occurs after executing function block STATUS: Detailed code value for the error

■ Function and description

Function block to read large data continuously from certain address from another station.
BYTE type can be used and only direct address(%IB, %QB, %MB) can be used for the variable.

■ ST_NOH/ST_NOL

Communication module station number of another station(Refer to RDTYPE function block for details)

■ VAR :

Only direct address can be used for start address and only BYTE type can be used.

Ex) '%MB100' - From 100th byte of memory
'%IB0.2.1' - From 1st byte area among input area assigned to 2nd slot(2) of main base(0)
'%QB0.3.1' - From 1st byte area among output area assigned to 3rd slot(3) of main base(0)

■ DATA_LEN

Indicate data number to read from another station(Max. Fnet:120Byte, Mnet:400Byte).

■ RDVAR

Local station byte area to store the data read from another station.

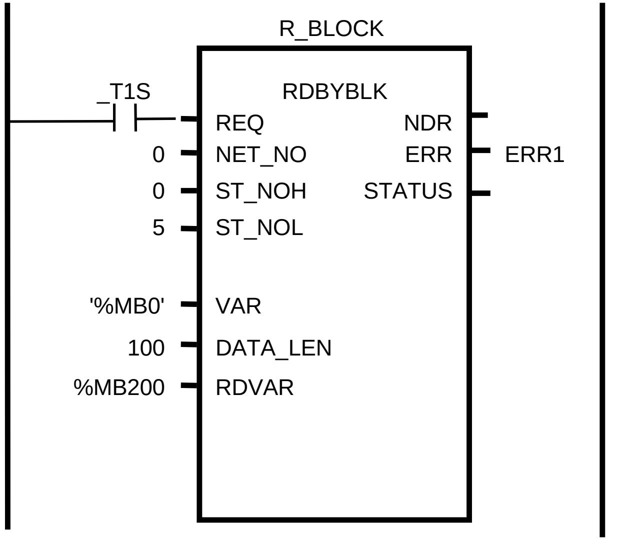
■ NDR / ERR / STATUS

Display the result of function block(Refer to RDTYPE function block for details).

■ **Program example : Providing that Fnet is installed to No.0 slot.**

If the station number of another station is 5 and read 100Byte from %MB0 of another station and store it to %MB200~%MB299 of local station area.
(Preset REQ condition every second)

LD



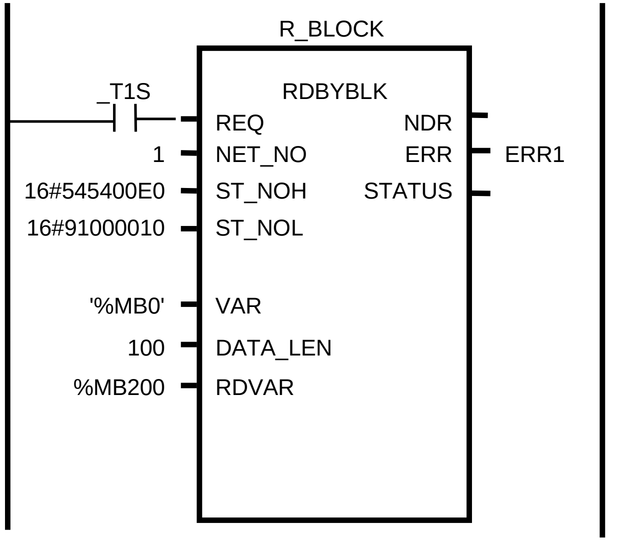
IL

```
CAL     RDBYBLK         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
```

■ **Program example : Providing that Mnet is used and Mini-MAP module is installed to No.1 slot.**

If MAC address of another station is 16#00E091000010 and read 100Byte from %MW0 of another station and store it to %MB200~%MB299 of station area. (Preset REQ condition every second)

LD



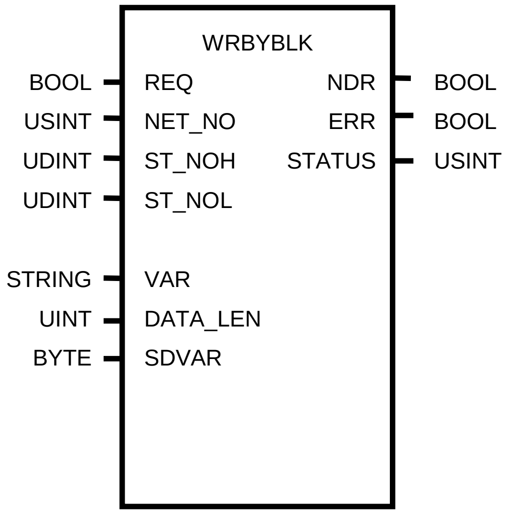
IL

```
CAL     RDBYBLK         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
```

WRBYBLK

Write continuous data to remote station(Byte block)
(Max. Fnet:120Byte, Mnet:400Byte)

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

Function	Description
 WRBYBLK BOOL REQ NDR BOOL USINT NET_NO ERR BOOL UDINT ST_NOH STATUS USINT UDINT ST_NOL STRING VAR UINT DATA_LEN BYTE SDVAR	**Input** REQ: Function block execution request at rising edge(0→1) NET_NO: Slot number installing the communication module for F/B transfer(0 ~ 7) ST_NOH: Fix to 0 during using Fnet (SAP, upper station number of communication module in remote station during using Mnet) ST_NOL: Station number of communication module installed in another station(Lower station module of another station during using Mnet) VAR: Variable identifier(defined by another station) to read the data(can not use direct address) SDVAR: Area that stores the data to be transferred to the another station DATA_LEN: Data number to be transferred **Output** NDR: On during receiving the data without error ERR: On when the error occurs after executing function block STATUS: Detailed code value for the error

■ **Function and description**

Function block to write large data continuously from certain address from another station.
BYTE type can be used and only direct address(%IB, %QB, %MB) can be used for the variable.

■ **ST_NOH/ST_NOL**

Communication module station number of another station(Refer to RDTYPE function block for details)

■ **VAR :**

Only direct address can be used for start address and only BYTE type can be used.
Ex) '%MB100' - From 100th byte of memory
'%IB0.2.1' - From 1st byte area among input area assigned to 2nd slot(2) of main base(0)
'%QB0.3.1' - From 1st byte area among output area assigned to 3rd slot(3) of main base(0)

■ **SDVAR**

Station BYTE area that stores the data to be transferred to another station

■ **DATA_LEN**

Indicate data number to transfer to another station(Max. Fnet:120Byte, Mnet:400Byte).

■ **NDR / ERR / STATUS**

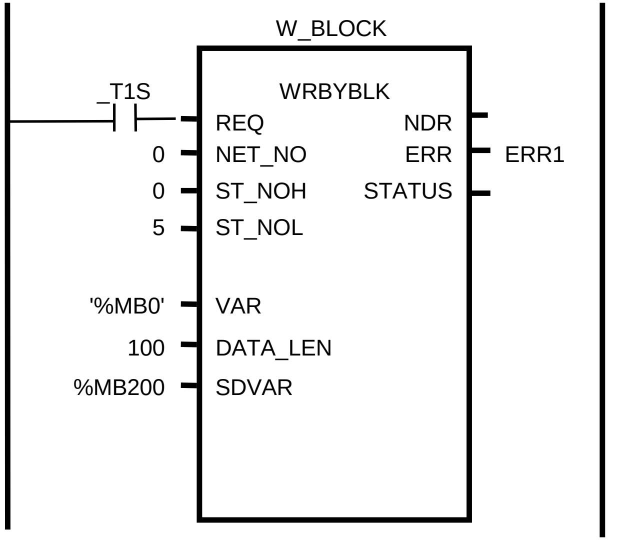
Display the result of function block(Refer to RDTYPE function block for details).

■ **Program example : Providing that Fnet module is installed to No.0 slot.**

If the station number of another station is 5 and transfer the data from %MB200~%MB299 of station area to %MB0~%MB99 of remote station.

(Preset REQ condition every second)

LD



IL

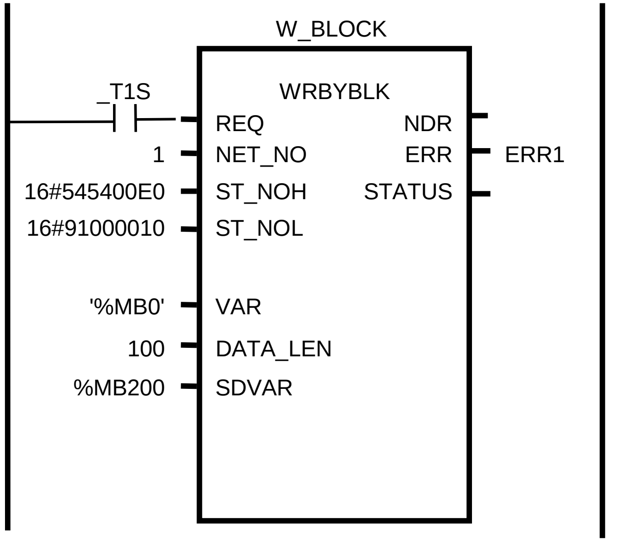
```
CAL    WRBYBLK        W_BLOCK
       REQ :=         _T1S
       NET_NO :=      0
       ST_NOH :=      0
       ST_NOL :=      5
       VAR :=         '%MB0'
       SDVAR :=       %MB200
       DATA_LEN :=    100

LD                    W_BLOCK.ERR
ST                    ERR1
```

■ **Program example : Providing that Mnet module is installed to No.1 slot.**

If MAC address of another station is 16#00E091000010 and transfer the data from %MB200 to %MB299 of station area to %MB0~%MB99 of another station.

(Preset REQ condition every second).

LD



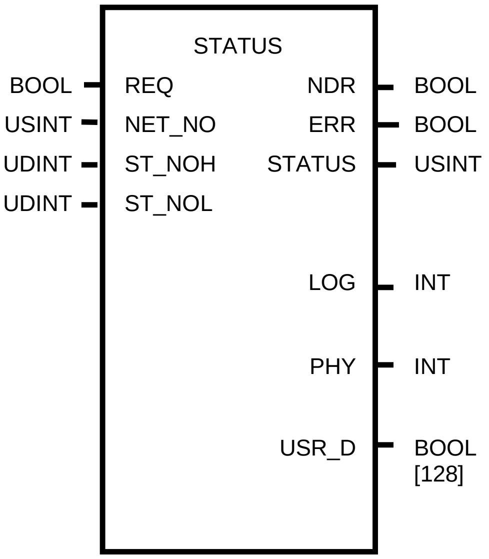
IL

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

LD                    W_BLOCK.ERR
ST                    ERR1
```

STATUS

Read the status of remote station

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

Function	Description
 STATUS BOOL REQ NDR BOOL USINT NET_NO ERR BOOL UDINT ST_NOH STATUS USINT UDINT ST_NOL LOG INT PHY INT USR_D BOOL [128]	**Input** REQ: Function block execution request at rising edge(0→1) NET_NO: Slot number installing the communication module for F/B transfer(0 ~ 7) ST_NOH: Fix to 0 during using Fnet (SAP, upper station number of communication module in another station during using Mnet) ST_NOL: Station number of communication module installed in another station(Lower station module of another station during using Mnet) **Output** NDR: On during receiving the data without error ERR: On when the error occurs after executing function block STATUS: Detailed code value for the error LOG: Function level to use for communication service PHY: H/W operation status of another station PLC USR_D: Total data of another station PLC status

■ Function and description

Function block to check the status of another station.

■ ST_NOH/ST_NOL

Communication module station number of another station(Refer to RDTYPE function block for details)

■ LOG

Indicate the function level to use for communication service(Logical State)

0 = State-Change-Allowed

■ PHY

Indicate H/W operation status of PLC by the physical state.

0 = Operational(in use)
1 = Partially-Operational-H/W(PLC and modules are not operated normally)
2 = Inoperable-H/W(Stop due to the error)
3 = Need-Commission-H/W(Unreliable data though it is used)

■ USR_D

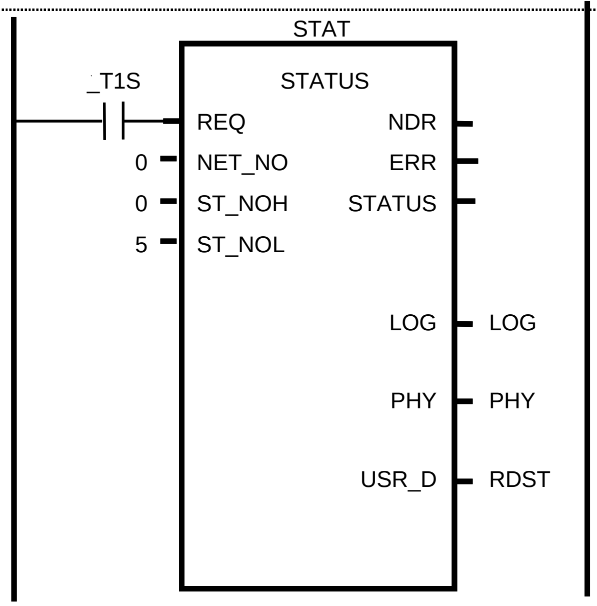
Supply 128Bit of Bit array for general status of another station PLC.
([0]~[64] are used and others are reserved.)

■ **Program example : Providing that Fnet module is installed to No.0 slot.**

In case of reading general information for remote station 5.

(Providing that REQ condition is set every second and RDST variable is declared as BOOL ARRAY[64])

LD



IL

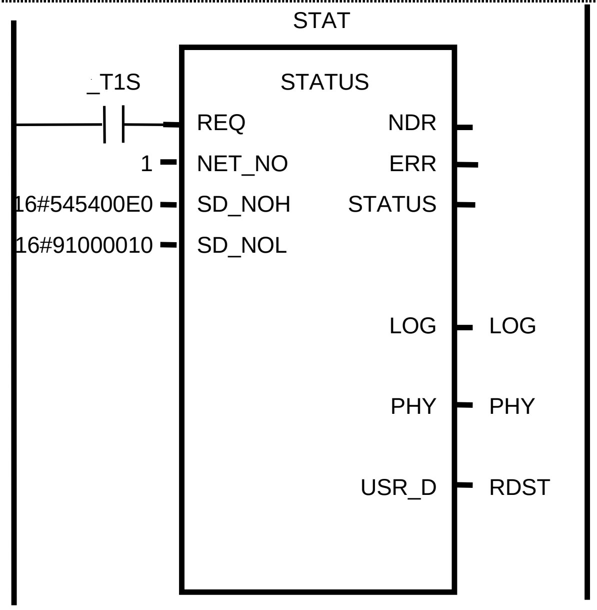
```
CAL     STATUS        STAT
        REQ :=        _T1S
        NET_NO :=     0
        ST_NOH :=     0
        ST_NOL :=     5
LD                    STAT.LOG
ST                    LOG
LD                    STAT.PHY
ST                    PHY
LD                    STAT.USR_D
ST                    RDST
```

■ **Program example : Providing that Mnet module is installed to No.1 slot.**

If MAC address of another station is 16#00E091000010(marked at the side of Mini-MAP module) and read general information of remote station.

(Providing that REQ condition is set every second and RDST variable is declared as BOOL ARRAY[64])

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                    RDST
```

BIT array description stored to USR_D in STATUS function block: Declared from [0] to [64]

Bit number	Representat-ive content	Bit classification	Detailed content	Description
S[0]~S[7]	CPU_TYPE	0x00	GM1	Display CPU type by S[7]~S[0] value.
		0x01	GM2	
		0x02	GM3	
		0x03	GM4	
		0x04	GM5	
		0x05	GM3_FSM	
		0x06	GM4_FSM	
		0x07	SRU	
		0x08	FAM	
		0x09	PMU500	
		0x0A	PADT	
		0x22	GT3	
		0x23	GK4	
		0x24	GK5	
		0x25	GK3_FSM	
		0x26	GK4_FSM	

S[8]~ S[15]	_VERSION_ NO	S[8]~S[11]	_VERSION_NO Lower mark	Ex) If v3.1 is displayed,
		S[12]~S[15]	_VERSION_NO Upper mark	(1: S[11]~S[8] are displayed by decimal, 3: S[15]~S[12] are displayed by decimal)

S[16]	_SYS _STATE		Local control	Display whether operation mode can be changed by mode key or PADT.
S[17]			STOP	Display CPU operation status.
S[18]			RUN	Display CPU operation status.
S[19]			PAUSE	Display CPU operation status.
S[20]			DEBUG	Display CPU operation status.
S[21]			Operation mode change cause	Operation mode change by key
S[22]			Operation mode change cause	Operation mode change by PADT
S[23]			Operation mode change cause	Operation mode change by remote PADT
S[24]			Operation mode change cause	Operation mode change by communication
S[25]			STOP by STOP function	SCAN end and STOP by STOP function under RUN mode operation

Bit number	Representative content	Bit classification	Detailed mark	Description
S[26]	SYS_STATE		Forced input	Display forced ON/OFF to input contact
S[27]			Forced output	Display forced ON/OFF to output contact
S[28]			STOP by ESROP function	Prompt stop by ESTOP function under RUN mode operation
S[29]			No meaning	
S[30]			Monitoring	Display the external monitoring for the program and variable
S[31]			Remote mode ON	Display the operation under remote mode

S[32]	_PADT_CNF		Local PADT connection	Bit indicating status of local PADT connection
S[33]			Remote PADT connection	Bit indicating status of remote PADT connection
S[34]			Remote communication connection	Bit indicating contact status of remote communication

S[35]	_DOMAIN_ST		Abnormal main parameter	Flag that checks and displays the abnormal of main parameter
S[36]			Abnormal I/O parameter	Flag that checks and displays the abnormal of I/O configuration parameter
S[37]			Abnormal program	Flag that checks and displays the abnormal of user program
S[38]			Abnormal access variable	Flag that checks and displays the abnormal of access variable
S[39]			Abnormal high-speed link parameter	Flag that checks and displays the abnormal of high-speed link parameter

S[40]	_CPU_ER		CPU configuration error	Error flag generated when CPU module is not operated normally due to the self diagnosis of CPU module, wrong installation of Base CPU, multi CPU configuration (refer to _SYS_ERR for details).
S[41]	_IO_ER		Module type inconsistency error	Representative flag. Detects and displays that I/O configuration parameter of each slot is different from the installed module or certain module is installed in the slot that can not be installed. (Refer to _IO_TYER_N and _IOTYER[n]).
S[42]	_IO_TYER		Module installation error	Representative flag. Detects and displays that module configuration of each slot is changed during operation. (Refer to _IO_DEER_N and _IO_DEER[n]).
S[43]	_FUSE_ER		Fuse cut-off error	Representative flag. Detects and displays that the fuse installed in module of each slot is broken. (Refer to _FUSE_ER_N and _FUSE_ER[n]).
S[44]	_IO_RWER		Read/Write error of input/output (trouble)	Representative flag. Detects and displays the error that can not read or write the I/O module of each slot. (Refer to _IP_RWER_N and _IO_RWER[n]).

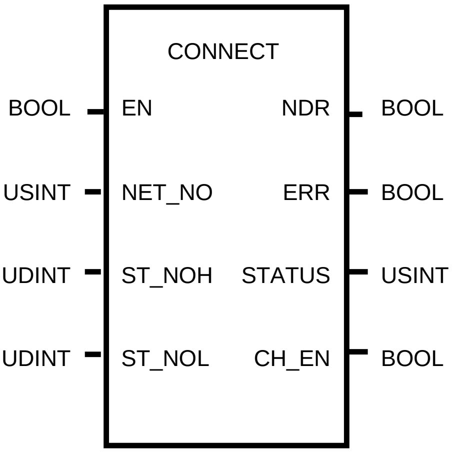
Bit number	Representative content	Bit classification	Detailed mark	Description
S[45]	_SP_IFER		Special/Communication module interface error(trouble)	Representative flag. Detects and displays the failure of special or communication module initialization of each slot or disable of normal interface due to malfunction of module. (Refer to _IP_IFER_N and _IP_IFER[n]).
S[46]	_ANNUN_ER		Trouble detection error of external device	Representative flag. Displays the trouble detection generation when the trouble of external device is recorded in _ANC_ERR[n] by user program.
S[47]	Not used			
S[48]	_WD_ER		SCAN WATCH-DOG error	Error generated when the scan time exceeds SCAN WATCH-DOG TIME set by parameter.
S[49]	_CODE_ER		Program code error	Error generated when the user program can not decode the command.
S[50]	_STACK_ER		STACK OVERFLOW error	Error generated when the program stack exceeds normal range during running the program.
S[51]	_P_BCK_ER		Program error	Error of the program memory damage or execution disable due to abnormal program.
S[52]	_RTC_ERR		System alarm RTC data abnormality	Flag displaying the data abnormality of RTC
S[53]	_D_BCK_ER		Data backup abnormality	If Hot/Warm restart program can not be executed since the data memory is damaged due to BACK_UP abnormality but Cold restart is executed, this flag indicates this problem and used in the initialization program. It will be automatically reset when initialization program is completed.
S[54]	_H_BCK_ER		Hot restart disable error	If restart program(Warm/Cold) is executed by the parameter since the recover of electricity exceeds hot restart time or the operation data for hot restart is not back-up normally, the flag indicates this problem and used in the initialization program. It will be automatically reset when initialization program is completed.
S[55]	_AB_SD_ER		Abnormal shutdown	When the program is stopped by the cutoff of power supply and restarted by warm restart, the flag indicates the operation error and used in the initialization program. It will be automatically reset when initialization program is completed. Indicates that the program is stopped by 'ESTOP' function.

Bit number	Representative content	Bit classification	Detailed mark	Description
S[56]	_TASK_ERR		Task conflict(Normal cycle, external task)	Flag indicates the task conflict when the tasks are duplicated during user program. (Refer to _TC_BMAP[n], _TC_CNT[n] for details.)
S[57]	_BAT_ERR		Battery trouble	Flag detects and indicates that the battery voltage is lower than rated value for user program and data memory back-up.
S[58]	_ANNUM_WR		Detect light trouble of external device	Representative flag indicates the trouble when the user program detect the alarm of external device and records it to _ANC_WB[n].
S[59]	Not used			
S[60]	Not used			
S[61]	_HSPMT1_ER		Over high-speed link parameter 1	Representative flag indicates high-speed link disable after checking the parameter of high-speed link. It will be reset during high-speed link disable.
S[62]	_HSPMT2_ER		Over high-speed link parameter 2	
S[63]	_HSPMT3_ER		Over high-speed link parameter 3	
S[64]	_HSPMT4_ER		Over high-speed link parameter 4	

CONNECT

Establish logical communication channel to another station
(For connection with other Mnet)

Product	GM1	GM2	GM3	GM4	GM5
Applicable	●	●	●		

Function	Description
	**Input** REQ: Function block execution request at rising edge(0→1) NET_NO: Slot number installing the communication module for F/B transfer(0 ~ 7) ST_NOH: Fix to 0 during using Fnet (SAP, upper station number of communication module in another station during using Mnet) ST_NOL: Station number of communication module installed in forced station(Lower station module of another station during using Mnet) **Output** NDR: On during receiving the data without error ERR: On when the error occurs after executing function block STATUS: Detailed code value for the error CH_EN: Result of channel establishment

Function and description

The communication to another station in Mnet is executed after establishing communication channel and SAP is required for the connection. SAP is classified by SSAP and DSAP and the manufacturer of Mini-MAP communication module supplies SSAP to the user.
The type of SAP is divided by Association SAP, Associationless SAP, Unspec.SAP.

- Association SAP
  Connect after establishing the channel by channel service during communication(Initiate) service.
- Associationless SAP
  Execute the communication assuming the communication channel is established internally without communication channel(Initiate) service.
- Unspec. SAP
  Satisfy both of Association SAP and Associationless SAP function.

If SAP supplied by other company is Association SAP or the communication shall be executed by the communication channel(Initiate) service, the communication channel(INITIATE) shall be established by CONNECT function block. However, for the communication with our product, this function block is not required.
The function block is operated not the edge but the level. Therefore, if the channel is established, the channel is kept when EN input level is "1" and CH_EN bit is set as "1". When the channel is released by the request, CH_EN bit is cleared to "0" and the user can use CH_EN bit for other function block.

EN

EN shall be operated when level is "1" and holds "1" in service. (BOOL)

NET_NO

Slot location where communication module to transfer the data by this FB among communication module in main base is installed. (0~7)

■ ST_NOH

Upper prefix number of communication module installed in another station for channel establishment and SAP.
ST_NOH=SSAP+DSAP+Upper station number of another communication module

■ ST_NOL

Lower station number of communication module installed in another station for channel establishment.
ST_NOL=Lower station number of another station

Ex) Connection to other Mnet

When the communication channel is established from A station(our company) to B station(other company) (Providing other Mini-Map module SAP=4E)
MAC address of A station: 16#00E091000000(our company),
MAC address of B station: 16#080070221C9A(other company)
ST_NOH: 16#10 (SSAP) 4E (DSAP:) 0800 (Upper station number of another communicat-ion module)
Therefore, ST_NOH = 16#104E0800
ST_NOL = 16# 70221C9A (Lower station number of another communication module)

- SAP(SSAP) provided by GLOFA Mini-MAP module for the connection with other Mimi-MAP is divided by 16#10 and 16#14.

■ NDR

When the function block is operated and ended normally, NDR is on till this function block is operated at next scan.

■ ERR

When the error occurs after operating function block or the channel release request is received from remote station, ERR is on till this function block is operated at next scan.

■ STATUS

Indicate detailed code value on the error and holds the value till this function block is operated at next scan.

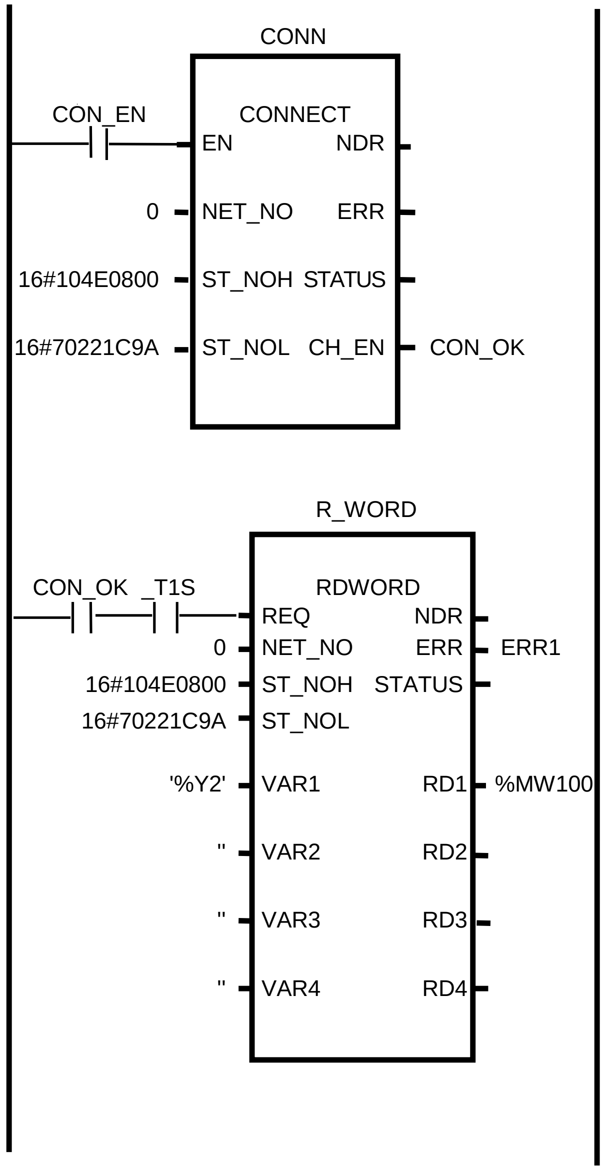
■ CH_EN

When the channel is established, CH_EN is "1" and when the channel is released, CHEN holds "0". When NDR is set to "1", CH_EN becomes "1" and holds "1" though NDR bit is cleared next. When ERR bit is set to "1", CH_EN bit is set to "0" and holds "0" till the channel is established.

■ **Program example : Providing that Mnet is used and communicated with other Mnet and Mini-MAP module is installed to No.0 slot.**

If MAC address of other company is 16#080070221C9A and read Y2 as 1 WORD(16Bit) after establishing the channel for other Mini-MAP and store it %MW100 area of station.

LD	IL

 CONN
CONNECT
CON_EN
EN
NDR
0
NET_NO
ERR
16#104E0800
ST_NOH
STATUS
16#70221C9A
ST_NOL
CH_EN
CON_OK
R_WORD
RDWORD
CON_OK
_T1S
REQ
NDR
0
NET_NO
ERR
ERR1
16#104E0800
ST_NOH
STATUS
16#70221C9A
ST_NOL
'%Y2'
VAR1
RD1
%MW100
''
VAR2
RD2
''
VAR3
RD3
''
VAR4
RD4
	CAL CONNECT CONN
	EN := CON_EN
	NET_NO := 0
	ST_NOH := 16#104E0800
	ST_NOL := 16#70221C9A
	LD CONN.CH_EN
	ST CON_OK
	
	LD CON_OK
	AND _T1S
	ST RD_REQ
	CAL RDWORD R_WORD
	REQ := RD_REQ
	NET_NO := 0
	ST_NOH := 16#104E0800
	ST_NOL := 16#70221C9A
	VAR1 := '%Y2'
	VAR2 := ''
	VAR3 := ''
	VAR4 := ''
	LD R_WORD.ERR
	ST ERR1
	LD R_WORD.RD1
	ST %MW100

* CON_EN is variable of initial value 1.

STATUS code value of function block and description

1) Error received from communication module

Value (Decimal)	Description
0	– OK(Success : No Error)
1	– Physical Layer Error of LINK(Send/Receive disable) – Cause of station Error and power-off of another station, station number wrong description and trouble.
	Receive (-) response from another station. Classified by the error.
33	– Can not find variable identifier(Object Undefined). – Not defined in the access variable area.
34	– Invalid Address- Struct error described in the specification of communication module and out of range.
50	– Invalid Response – When the response is not received as requested or CPU trouble of another station
113	– Object Access Unsupported – VMD Specific, Symbolic Address – Maximum value out
114	– Access variable is not downloaded(Object non Existent) – Download disable in the access variable area
187	– Receive the error code except assigned code(Communication code value of other company)- Receive the code except defined error code
3	– Identifier of function block to be received does not exist in the communication channel. – Value not used in our company.
4	– Data Type Mismatch
5	– Receive reset from other station – Value not used in our company.
6	– Function block of another station is not ready.(Receiver Not Enabled) – Value not used in our company.
7	– Function block status of station is wrong. (Remote Device in Wrong State) – Value not used in our company.
8	– OBJECT requested by user can not be accessed. (Access Denied to Remote Object) – Value not used in our company.
9	– Receive disable due to excessive function block of another station (Receiver Overrun) – Value not used in our company.

Value (Decimal)	Description
10	– Response wait time out(Time out) – If the response is not received from another station in certain period
11	– Struct error
12	– Abort(Local/Remote) – Disconnect the connection due to serious error.
13	– Reject(Local/Remote) – Unsuitable format for MMS or error due to noise
14	– Communication channel establishment error(Connect/Disconnect) – Error relating logical communication channel establishment at the service on PI/DOMAIN/GEN and communication with other communication module.(For Mnet only)
15	– High-speed communication and connection service error

2) Error in CPU

Value (Decimal)	Description
16	If the location of computer communication module is wrong
18	Input parameter set error
20	If response frame not requested is received
21	If the response is not received from computer communication module

3) Error relating remote function block(FSM)

Value (Decimal)	Description
128	FSM power error
129	BASE(Rack) number error
130	Slot number error
131	Module information error
132	Data range error(Invalid Range)
133	Inconsistency of data type
134	IP module is not ready
135	Read/Write error of IP module
136	Access failure(Bus access error)
137	Error except assign code

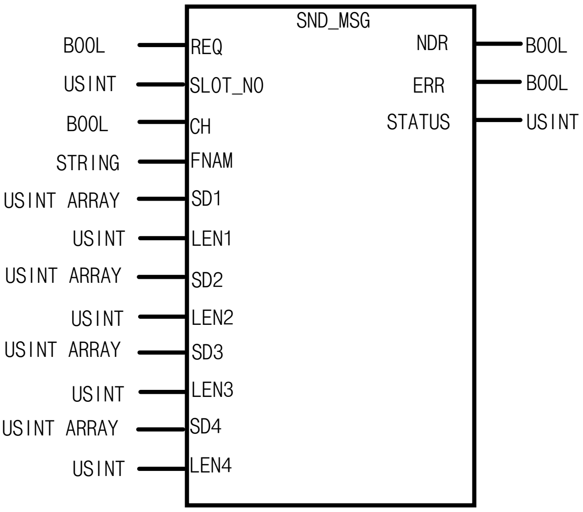
10.2 Computer link module function block libraries

Describes the function block that controls the frame, arranged by frame editor, in PLC Program example
Program that set the output contact %00.3.0 when No. 5 pulse inflows to input contact %I0.1.14.
Function block for CLM is SND_MSG and RCV_MSG.

SND_MSG(Send Message)

Send the data to another station

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

<table>
<tr><th>Function</th><th>Description</th></tr>
<tr><td>


</td><td>
<b>Input</b> REQ: Function block execution request at 1(rising edge)
SLOT_NO: Slot number input stalling CLM
CH: Channel number input to send the data
0: RS-232-C, 1: RS-422
FNAM: Frame name input to be sent
SDx: Array variable name input to be sent
LENx: Each array length input to be sent
In case of no data to be sent, input 0.
(x: 1,2,3,4)
<b>Output</b> NDR: On during receiving the data without error
ERR: On when the error occurs after executing function block
STATUS: Detailed code value for the error
</td></tr>
</table>

■ Function

Send the frame downloaded to computer link module(CLM) with input data of variable through the channel assigned to CLM.

Before executing this function block, download the frame of same name to the frame of 'FNAM' by using the frame editor. Further, input SDx of same number to array variable used in the frame. Refer to G3L-CUEA and G4L-CUEA technical document for frame download method.

■ Error

(1) Status value in CPU (Decimal)

STATUS value	Meaning	Treatment
16	CLM is located wrongly.	Input exact slot value.
20	1) Wrong library is used. 2) Response frame is wrong.	1) Check the data of library.(Check that [Communi.fb] file is dated after OCT, 1996) 2) Check again the receive frame of local station and send frame of remote station.
21	The response is not received from CLM.(Wait time out)	Check the operation mode of CLM. Check whether CLM is user defined communication status.

(2) Status value in CLM (Decimal)

STATUS value	Meaning	Treatment
64	Channel status of RS-232C/422 is not run.	Run the operation by frame editor. (Menu: On-line operation conversion)
65	The frame name used in frame editor does not match with that used in function block.	Make same the frame name used in frame editor to that used in function block.
66	Frame name can not be found due to CPU trouble.(For send)	1) Download again the frame. 2) Check the CPU.
67	The frame assigned to FNAM is not received from remote station.	1) Check the receive frame again. 2) Check the send frame of remote station.
68	The frame is not downloaded from the frame editor.	Download the frame.
69	ASCII <-> HEX conversion error	Check whether the receive data is ASCII or HEX.
70	Array size assigned by frame editor does not match with that (size assigned to LENx) used in function block.	Check the data size and fix it. (Data size is Byte value)
100	Array type assigned to SDx or RDx is different.	Set the array type to unsigned integer.
102	The frame name does not exist in FNAM.	1) Check the frame name again. 2) Download the frame again.
103	The frame definition is wrong.	1) Check respective frame content by frame editor. 2) Download the frame again.
104	The frame is not downloaded by the frame editor.	Download the frame
115	The operation mode is not the user defined communication mode.	Fix all switches exactly. User defined communication RS-232C: 0,2,4(0:Continuous mode) RS-422/485: 2,5,6

■ Program example

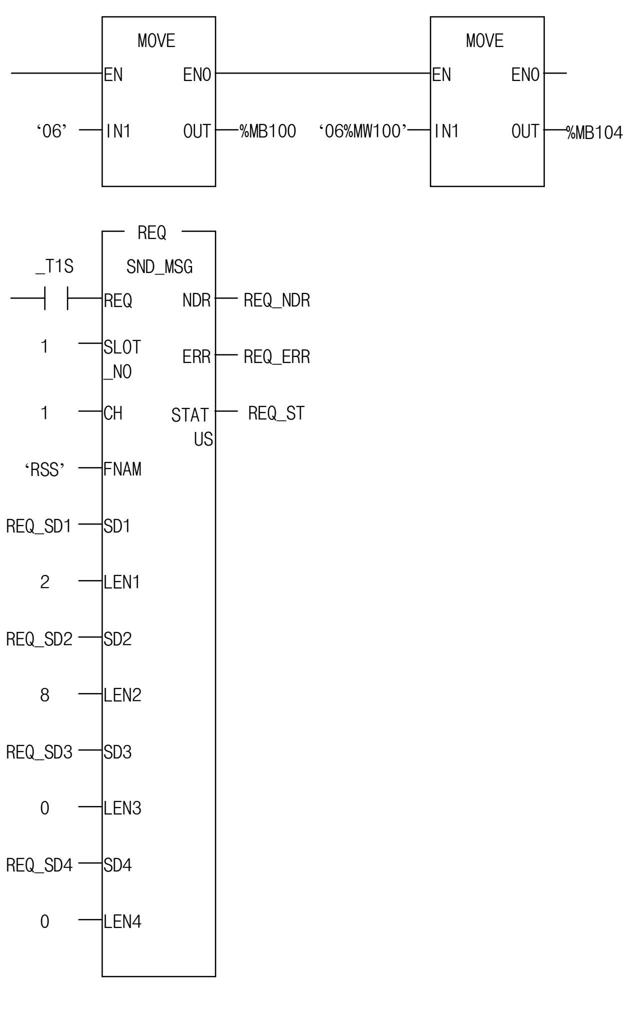
This example is prepared under below assumption.

1) Remote station: station number 06, RS-422, operation mode is GLOFA CLM
2) Frame named 'RSS' is download to CLM.

	Header	Segment1	Segment2	Segment3	Segment4 ~ Segment8	Tail
Type		ARRAY	CONSTANT	ARRAY	NONE	
Content	[ENQ]	SD1	PSS01	SD2		[EOT]
Type		2		8		
Remark	Set the header to ENQ(05H)	Set ARRAY variable to 'SD1' of 2Byte(used for another station set area in this example).	Fixed data set (Set 1 for Read command and block number in this example).	Set ARRAY variable name to 'SD2' of 8Byte(variable name size area 2Byte + Variable name area 6Byte)	Not used	Set tail to EOT (04T)

* Variable REQ_SD1 is set to %MB100 area and variable REQ_SD2 is set to %MB104 area.

LD



IL

```
LD                          '06'
MOVE
ST                          %MB100
LD                          '06%MW100'
MOVE
ST                          %MB104
CAL       SND_MSG           REQ
          REQ:=             _T1S
          SLOT_NO:=         1
          CH:=              1
          FNAM:=            'RSS'
          SD1:=             REQ_SD1
          LEN1:=            2
          SD2:=             REQ_SD2
          LEN2:=            8
          SD3:=             REQ_SD3
          LEN3:=            0
          SD4:=             REQ_SD4
          LEN4:=            0
LD                          REQ.NDR
ST                          REQ_NDR
LD                          REQ.ERR
ST                          REQ_ERR
LD                          REQ.STATUS
ST                          REQ_ST
```

(Description)

Assuming that the protocol receiving the data from another station through computer communication is as below.

Frame start + Station Number + Instruction + (Size + Data area) + Transmission end

Providing that prefix number of remote station is 06, command is RSS01, data area is sent to %MW100 and other except command is variable in the program, declare the variable as CONSTANT and ARRAY to input the data on the frame editor. When the data process compiles the send function block(SND_MSG) to frame name('RSS') and REQ_SD1 variable data to SD1 and REQ_SD2 variable data to SD2, send the data referencing 'RSS' frame name to ENQ + SD1(REQ_SD1) + RSS01 + SD2(REQ_SD2) + EOT type.
SND_MSG function block SD1~SD4 shall use Unsigned Short Integer(USINT:8Bit) and can not use the data like %MW100 for SD1~SD4 since USINT describes the data from 0 to 255. In this case, declare REQ_SD1 to 2 USINT array type(prefix '0' and '6' for send) and define the area to %MB100. As same method, declare REQ_SD2 to 8 USINT array type(06%MW100: for size 2, send data area 6) and define the area as %MB104. As SD3 and SD4 are not used, declare the area as USINT array type. Then, move string '6' to %MB100 and string '06%MW100' to %MB104, the data is sent to the required protocol sharing the data.

REQ_SD1 content

Memory address	String content
%MB100	0
%MB101	6

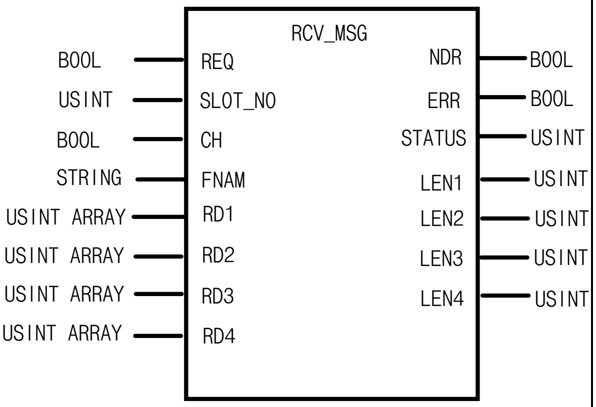
REQ_SD2 content

Memory address	String content
%MB104	0
%MB105	6
%MB106	%
%MB107	M
%MB108	W
%MB109	1
%MB110	0
%MB111	0

RCV_MSG(Receive Message)

Receive the data to remote station

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

Function	Description
 RCV_MSG BOOL USINT BOOL STRING USINT ARRAY USINT ARRAY USINT ARRAY USINT ARRAY REQ SLOT_NO CH FNAM RD1 RD2 RD3 RD4 NDR ERR STATUS LEN1 LEN2 LEN3 LEN4 BOOL BOOL USINT USINT USINT USINT USINT	**Input** REQ: Function block execution request at 1(rising edge) SLOT_NO : Slot number input stalling CLM CH: Channel number input to send the data 0: RS-232-C, 1: RS-422 FNAM: Frame name input to be sent RDx: Array variable name to store received data (x: 1,2,3,4) **Output** NDR: On during receiving the data without error ERR: On when the error occurs after executing function block STATUS: Detailed code value for the error LENx: Each array variable length received from remote station (x: 1,2,3,4)

■ Function

Function block that stores the data to RDx variable if receive frame downloaded to CLM is transferred from another station.

Before executing this function block, download the frame with same name of 'FNAM'. Refer to G3L-CUEA and G4L-CUEA technical document for frame download method.

■ Error

Refer to [Error] in SND_MSG function block.

■ Program example

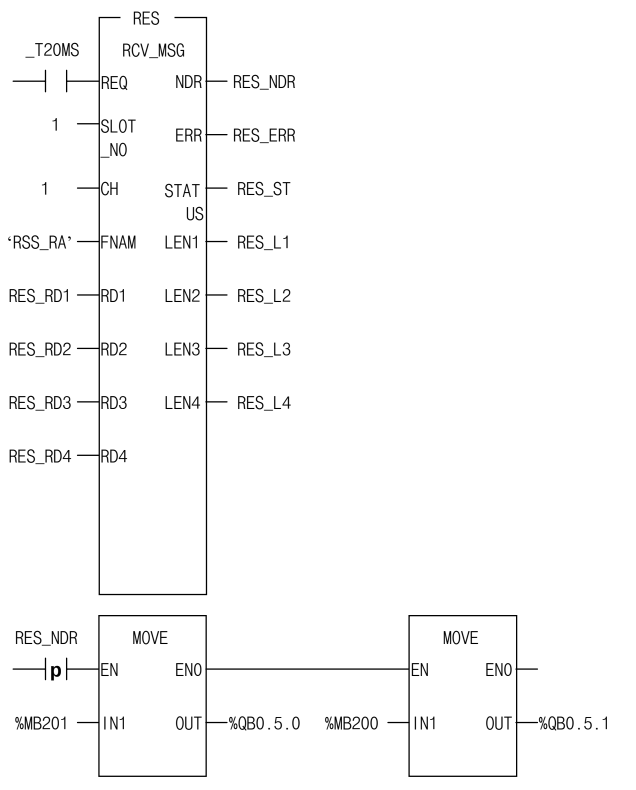
This example is prepared under below assumption.

1) Providing that the data of 06 station is received and RS-422 and operation mode is GLOFA PLC CLM.

2) Providing 'RSS_RA' frame is downloaded to CLM.

	Header	Segment1	Segment2	Segment3~Segment8	Tail
Type		CONSTANT	ARRAY	NONE	
Content	[ACK]	06RSS0102	RD1		[ETX]
Type			Hexa		
Size			2		
Remark	Set the header to ACK(06H)	Set the fixed receive data 01: Block number 02: Receive data size(2Byte)	Read 2Byte data and send it to RD1 of function block.	Not used.	Set tail to ETX (03H)

LD



IL

```
CAL     SND_MSG     RES
        REQ:=       _T20MS
        SLOT_NO:=   1
        CH:=        1
        FNAM:=      'RSS_RA'
        RD1:=       RES_RD1
        RD2:=       RES_RD2
        RD3:=       RES_RD3
        RD4:=       RES_RD4
LD                  RES.NDR
ST                  RES_NDR
LD                  RES.ERR
ST                  RES_ERR
LD                  RES.STATUS
ST                  RES_ST
LD                  RES.LEN1
ST                  RES_L1
LD                  RES.LEN2
ST                  RES_L2
LD                  RES.LEN3
ST                  RES_L3
LD                  RES.LEN4
ST                  RES_L4
LD                  RES_NDR
CAL     R_TRIG      M_EN
        CLK :=      RES_NDR
LD                  M_EN.Q
JMPN                JMP_1
LD                  %MB201
MOVE
ST                  %QB0.5.0
LD                  %MB200
MOVE
ST                  %QB0.5.1
JMP_1:
```

[Note] Providing that 'RES_RD1' memory is allocated to %MB200 in above example.

(Description)

Assuming that the protocol receiving the data from another station through computer communication is as below.

Frame start + Fixed data + Data + Transmission end

If send ACK to frame start and 06RSS0102 to fixed data and ETX to transmission end, the frame editor defines the header to [ACK], tail to [ETX], fixed data content to CONSTANT(06RSS0102), variable data to RD1. To process variable 2Byte in the function block, define RES_RD1 variable as 2 array of USINT data type and %MB200 of storage area and set this variable to RCV_MSG RD1. Output the receive data, which is input with the protocol defined at the frame editor 'RSS_RA' to %QB0.5.0 and %QB0.5.1.