Mitsubishi general-purpose sequencer Provider for MELSEC-A series

Version 1.1.3

User's guide

March 27, 2020

【 remarks 】

This document is translated from Japanese into English by the machine translation.

Some of the images used in this manual are quoted from "A-compatible Ethernet interface module user's manual (detailed)" and "computer link / multidrop link module user's manual".

【 revision history 】

Version	Date	Content
1.0.0	2017-10-10	First edition.
1.1.0	2018-08-03	Added UDP and Retry options
1.1.1	2018-10-30	Memory leak was corrected.
1.1.2	2019-12-25	Support array partition size specification. Added communication protocol command correspondence table.
1.1.3	2020-03-27	Bug fixed at AddController.

【 hardware 】

Model	Version	Notes
A2ACPU		It corresponds to Ethernet and the serial connection.

Contents

1. Introduction

This book is an user's guide of the CAO provider that write/reads data to a general-purpose sequencer made by Mitsubishi Electric Corporation (MELSEC-A series). CAO provider (CaoProvMELSECAnA.dll) that treats in this book is called MELSEC-A provider.

The MELSEC-A provider performs MELSEC communication protocol (hereinafter referred to as MC protocol) communication for A-compatible Ethernet interface units and computer link units of the Mitsubishi Electric sequencer MELSEC-A series.

Details of the outline and the variable of the MELSEC-A provider have been described to the chapter 2.

The correspondence situation and the data string of the communication command mounted in the MELSEC-A provider depend on a general-purpose sequencer (MELSEC-A series) that becomes a destination.(1)

For details on communication and the setting method, refer to "A-Compatible Ethernet Interface Unit User's Manual (Details)" and "Computer Link / Multi-Drop Link Unit User's Manual".

[1] The MELSEC-FX1 / 2/3 series uses the same MC protocol, so it works theoretically, but we have not confirmed the operation.

2. Outline of provider

2.1. Outline

The MELSEC-A provider is a CAO provider that writes / reads data to Mitsubishi Electric sequencers using the MC protocol QnA compatible 1E frame and QnA compatible 1C frame format 1, 2, 3, and 4.
Its file format is DLL (Dynamic Link Library), which is dynamically loaded from the CAO engine when used. To use the MELSEC-A provider, you need to install the ORiN2 SDK or manually register the registry by referring to the table below.

Table 2-1 MELSEC-A provider

File name	CaoProvMELSECAnA.dll
ProgID	CaoProv.MELSEC.AnA
Registry registration	regsvr32 CaoProvMELSECAnA.dll
Blotting out of registry registration	regsvr32 /u CaoProvMELSECAnA.dll

2.2. Directions point

2.2.1. About the connection with CPU unit

For the connection between the PC and CPU unit, refer to "A-compatible Ethernet Interface Module User's Manual (Detailed Edition)" and "Computer Link / Multidrop Link Module User's Manual".

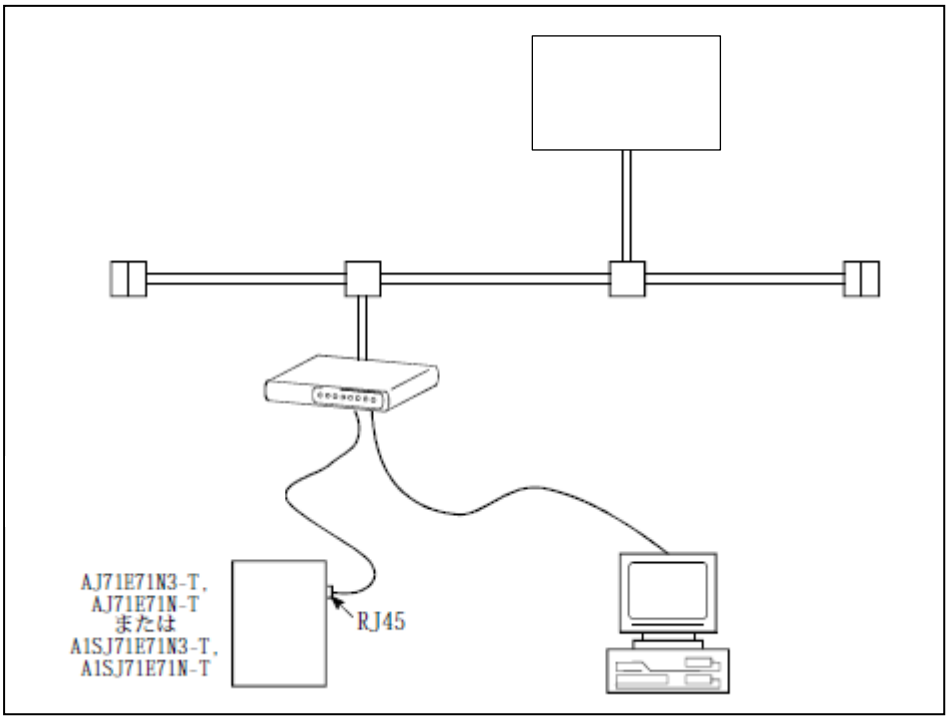


Figure 1 ethnernet connection ex.

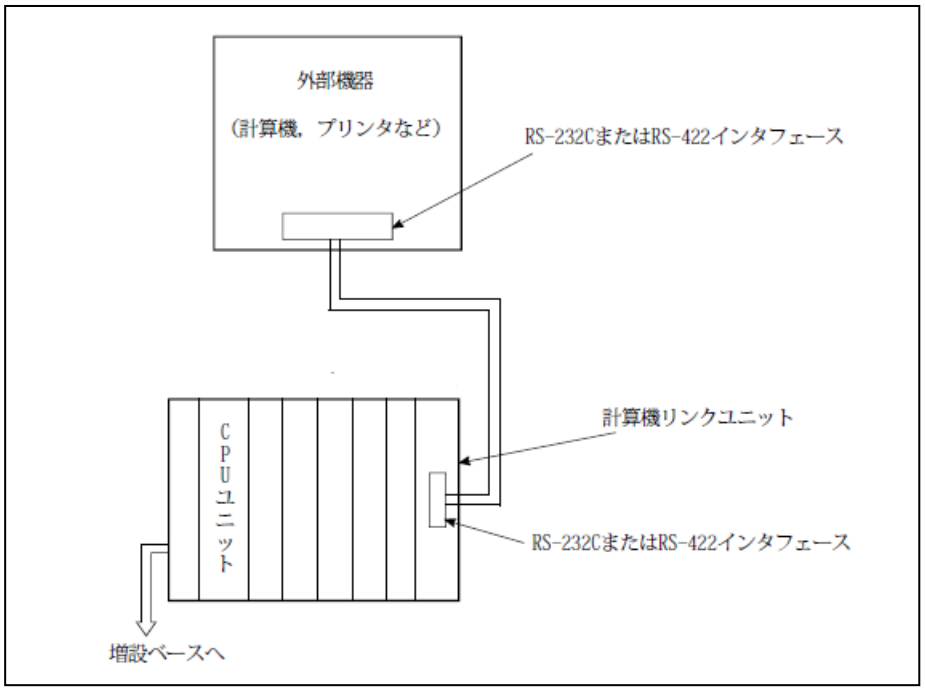


Figure 2 sirial connection ex.

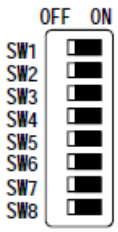
2.2.2. Data Code Settings

In Ethernet connection, communication is performed using binary code of QnA compatible 1E protocol of MC protocol.

You need to set the data code to binary.

4.3.2　交信条件の設定

交信条件設定スイッチ ＊1



スイッチ名称	設定項目	設定内容	工場出荷時
SW1	TCPタイムアウトエラー時の回線処理選択	TCP ULPタイムアウトエラー発生時の回線処理を選択する。 OFF：TCP ULPタイムアウトエラーの発生により回線をクローズする。 ON ：TCP ULPタイムアウトエラーが発生しても回線をクローズしない。	OFF
SW2	データコード設定	他ノードとの交信データのデータコード種別を選択する。(3.3項参照) OFF：バイナリコードにより交信を行う。 ON ：ASCIIコードにより交信を行う。	OFF
SW3	使用不可（OFF固定）		OFF
SW4			OFF
SW5			OFF
SW6			OFF
SW7	CPU交信タイミング設定	シーケンサCPUがRUN中に，他ノードからのデータ書込みの許可／禁止を選択する。(シーケンサCPU内データの読出し／書込み交信用) OFF：シーケンサCPU RUN中には，他ノードからの書込みを禁止する。 ON ：シーケンサCPU RUN中でも，他ノードからの書込みを行う。	OFF
SW8	イニシャルタイミング設定	イニシャル処理を起動するタイミングを選択する。 OFF：クイックスタート（遅延時間なしで起動） 一つのネットワークで全体構成されている場合に設定する。 ON ：ノーマルスタート（20秒の遅延時間後に起動） 複数のネットワークで全体構成されている場合に設定する。	OFF

＊1　ハードウェアバージョンが「B版」以降のA1SJ71E71N-B5の交信条件設定スイッチは，下記のようになります。

Figure 3　Ethernet unit setting method

In the serial connection, the frame format 1, 2, 3, and 4 of QnA compatible 1C protocol of MC protocol is exchanged by ASCII code.

Since the data code is ASCII only, setting of the data code is not required.

2.2.3. About reconnection processing when using TCP communication

Please note the following points when using TCP communication.

If you do not have enough time to reconnect the session, use UDP communication.

After disconnecting the session, allow enough time to re-open.

2.2.4. About communication with external devices

When communicating with an external device, communication cannot be performed unless a sequencer ladder is created.

For Ethernet connection, it is necessary to create a ladder and open ports as shown below.

In the following example, set the IP address to "192.168.2.1", open UDP / IP to connection 1 on port "1025", and open TCP / IP Unpassive to connection 2 on port "1026".

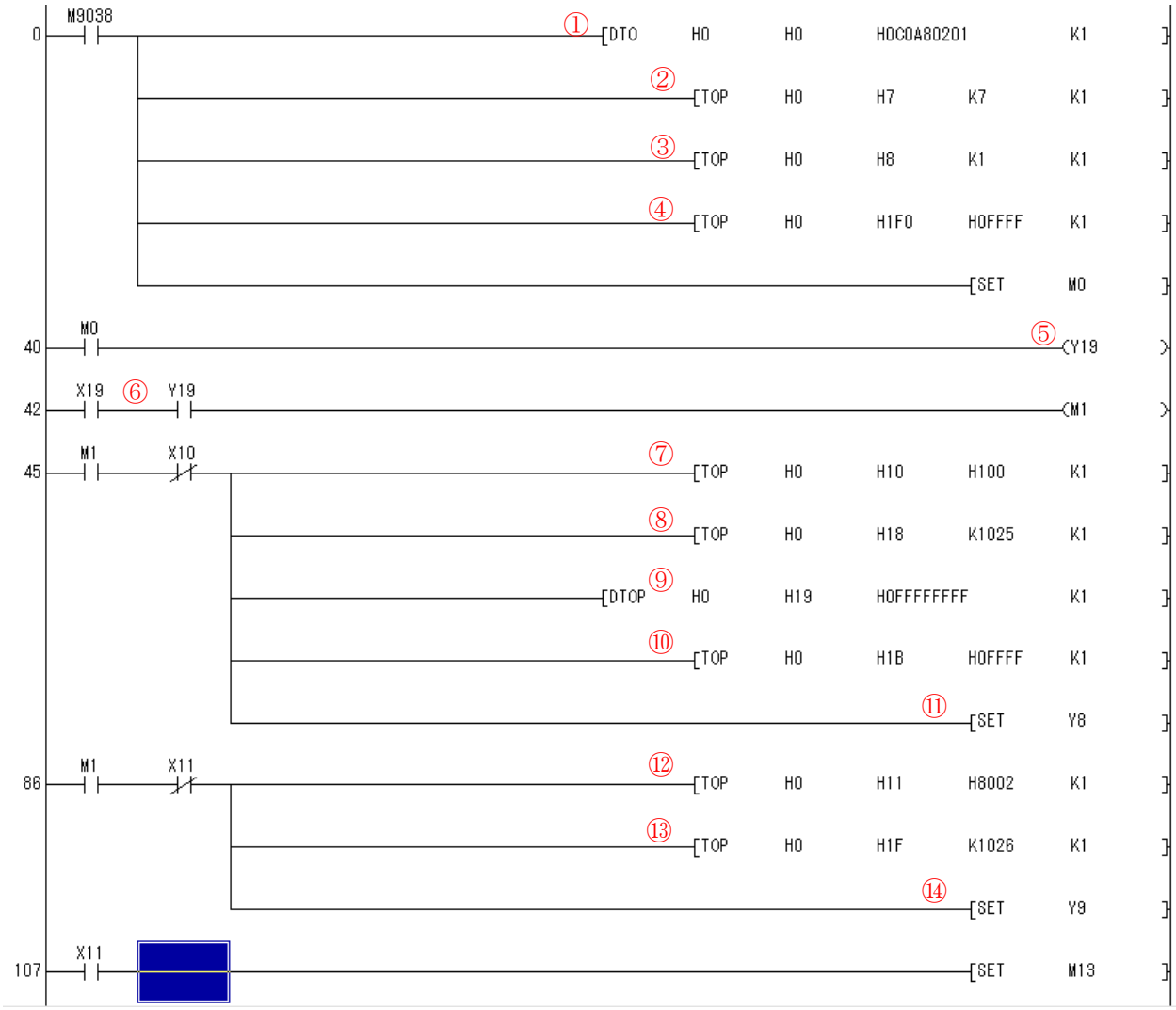


Figure 4　Ethernet connection setting method

① Set the IP address to "192.168.2.1"

② Set the survival confirmation start interval timer to 14 seconds (7 × 2 seconds)

③ Set the survival confirmation interval timer to 2 seconds (1 × 2 seconds)

④ Communication instruction during STOP is set to all connections "enabled" (0xFFFF)

⑤ Turn on the initial request signal (Y19)
⑥ After the initial complete signal (X10) turns ON, shift to open processing
⑦ Set UDP / IP (0x100) for connection 1 protocol
⑧ Set the own port number of connection 1 to "1025"
⑨ Set other IP address to "0xFFFFFFFF" to perform broadcast communication
⑩ Set other port number to "0xFFFF" to perform broadcast communication
⑪ Turn on open request signal (Y8) of connection 1
⑫ Set TCP / IP Unpassive (0x8002) for the connection 2 protocol
⑬ Set "1026" to the local port of connection 2
⑭ Set "1026" to the local port of connection 2

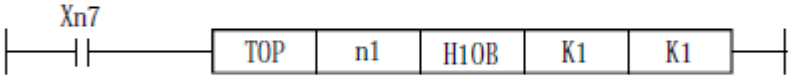
For serial connection, the following ladder needs to be created depending on whether or not the CD terminal check is performed.

(3) RS-232C CD端子チェック設定について

RS-232C CD端子チェック設定（バッファメモリのアドレス10BHで設定，3.10項参照）による計算機リンクユニットの，CD信号に対する動作は次のとおりです。

	CD端子チェックあり	CD端子チェックなし
全二重通信	計算機リンクユニットはCD信号（受信キャリア検出）のON状態で送受信処理を行う。データ交信時は，CD信号がOFFすると，計算機リンクユニットの伝送シーケンスを初期化する。	計算機リンクユニットは，CD信号のON/OFF状態に関係なく全二重通信方式で送受信処理を行う。 （計算機リンクユニットは，CD信号のON/OFFチェックを行わず，CD信号がONと同じ処理を行う。） CD信号をON/OFFさせられない外部機器とのデータ交信が可能。

RS-232C CD端子チェック設定で，“CD端子チェックなし”を設定するときは，次に示すシーケンスプログラムを組み込んでください。
“CD端子チェックあり”を設定するときは，計算機リンクユニット立上がり時のデフォルト値が「0」（CD端子チェックあり）のため，次に示すシーケンスプログラムの組込みは不要です。



＊RS-232C CD端子チェック設定については，第5章～第7章の各計算機リンク機能説明項の中(5.2.1項(2)②，6.2.4項(2)⑥，7.2.6項(2)④)でも説明しています。

Figure 5 Serial connection setting method

2.3. Method property

2.3.1. CaoWorkspace::AddController method

When the Controller object is generated, initial of Ethernet (TCP) communication or serial communications is processed in the MELSEC-A provider. When connecting it, the device is specified by an optional character string.

Format AddController (< bstrCtrlName:BSTR > , < bstrProvName:BSTR >,
<bstrPcName:BSTR >,<bstrOption:BSTR>)

bstrCtrlName : [in] controller name
bstrProvName : [in] provider name. fixed value ="CaoProv.MELSEC.AnA"
bstrPcName : [in] Execution machine name

bstrOption : [in] optional character string

The list specified for an optional character string is shown as follows.

Table 2-2 Optional character string of CaoWorkspace::AddController

Option [2]	Explanation
Conn=< connected parameter >	Indispensability. A communication form and the connected parameter are set. (See 2.3.1.1)
ProcessingPoints[=< Maximum processing points per communication >]	Set the maximum number of processing points to be performed in one communication. (See 2.3.1.2)
Format[=< format type>]	Form (1-4) used for serial communications is set. (default: 1) Only serial communications are effectively optional. (See2.3.1.3)
SumCheck[=< effective/invalidity >]	Effective (1)/invalidity (0) of the sum check of serial communications is set. (default: Effective) Only serial communications are effectively optional.
Timeout[=< timeout period >]	Set the timeout time in milliseconds for sending and receiving. (default: 3000)
Retry[=<retry count>]	The frequency in which the communication is sent again is set. (default：0)

2.3.1.1. Conn is optional.

Connected parameter character string of optional Conn is shown as follows. The parameter in the square bracket ("") shows a possible omission here. Moreover, the underlined part under the explanation of each parameter shows the default value when optional specification is omitted.

【 Ethernet device 】

"Conn=ETH:<Dest IP Address>:<Dest Port No>"

"Conn=TCP:<Dest IP Address>:<Dest Port No>"

"Conn=UDP:<Dest IP Address>:<Dest Port No>"

< Dest IP Address > : Internet Protocol address of connection destination.

[2] Items enclosed with square brackets ("[]") are omittable. Underlined part shows the default value when the option is not specified.

< Dest Port No > : Port number of connection destination.

【 Serial device 】

"Conn=com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]"

<COM Port>	:	COM port number. '1'-COM1,'2'-COM2, …
<BaudRate>	:	Transmission rate.
		300,600,1200,2400,4800,9600,19200
<Parity>	:	Parity. 'N'-NONE,'E'-EVEN,'O'-ODD
<DataBits>	:	Number of data bits. '7'-7bit,'8'-8bit.
<StopBits>	:	Number of stop bits. '1'-1bit,'2'-2bit.

(example 1)	"com:1"	Communication port COM1 (, 19200bps, None, 8bits, 1bit)
(example 2)	"com:2:9600"	Communication port COM2 and 9600bps (, None, 8bits, 1bit)
(example 3)	"com:3:4800:N:8:2"	Communication port COM3, 4800bps, and None, 8bits, and 2bit

2.3.1.2. ProcessingPoints option

Specify the maximum number of processing points to be performed in one communication. If the number of processing points is exceeded, communication will be performed by dividing [3].

For example, if 700 word devices are communicated with the default settings when connecting to an Ethernet device, update 256 + 256 + 188 three times.

Each point can be set only in even point units. Also, you can only set up to 256 points..

【Ethernet device】

"[ProcessingPoints=[<BitDevInBitUnit>:<BitDevInWordUnit>:<WordDevInWordUnit>]]"

- < BitDevInBitUnit > : Communicates bit devices in bit (1 point) units.
  The default is 256 points..
- < BitDevInWordUnit > : Communicates bit devices in word (16 points) units.
  The default is 128 words (2048 points).
- < WordDevInWordUnit > : Communicates word devices in word (1 point) units.
  The default is 256 words.

【Serial device】

"[ProcessingPoints=[<BitDevInBitUnit>:<BitDevInWordUnit>:<WordDevInWordUnit>]]"

< BitDevInBitUnit > : Communicates bit devices in bit (1 point) units.
The default is 160 points.

< BitDevInWordUnit > : Communicates bit devices in units of words (16 points).
Default is 10 words (160 points).

< WordDevInWordUnit > : Communicates word devices in word (1 point) units.
The default is 64 words.

(ex) "Path=X0, Elem=5" Get the value of X0 to X4 as bit value

(ex) "Path=D10, Elem=0x10" Get the value of D10 to D25 as a word value

(ex) "Path=D10, Elem=&H10" Get the values of D10 to D25 as word values.

(ex) "Path=D10, Elem=10H" Get the value of D10 to D25 as a word value

(ex) "Path=M100, VT=I2, Elem=2" Get the value of M100 to M131 in word units

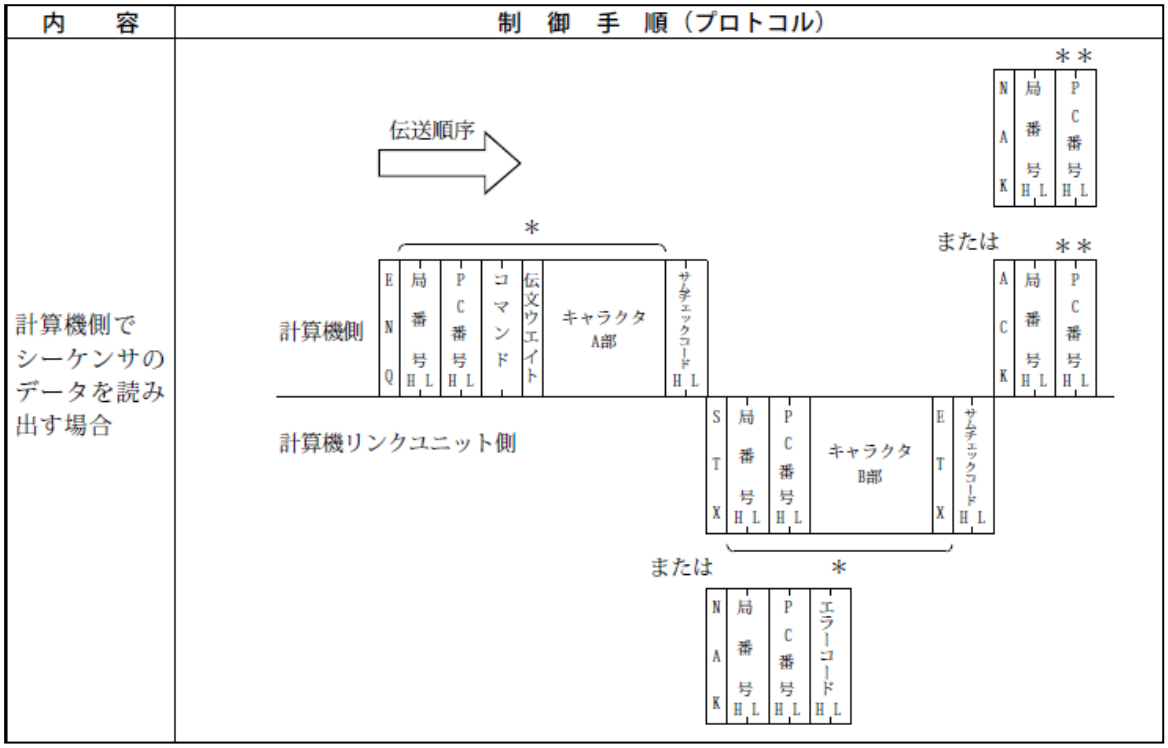
[3] The maximum number of processing points that can be communicated at one time differs for each unit. Refer to the unit's user's manual for details..

2.3.1.3. Format is optional.

It is a setting for the communication by ASCII code. The difference of each form becomes the following if it thinks based on form 1. Please see the manual for details.

Form (form)	Explanation
2	Form that added block number to each telegram message
3	Form that encloses each telegram message with STX and ETX
4	Form that added CR and LF to each telegram message

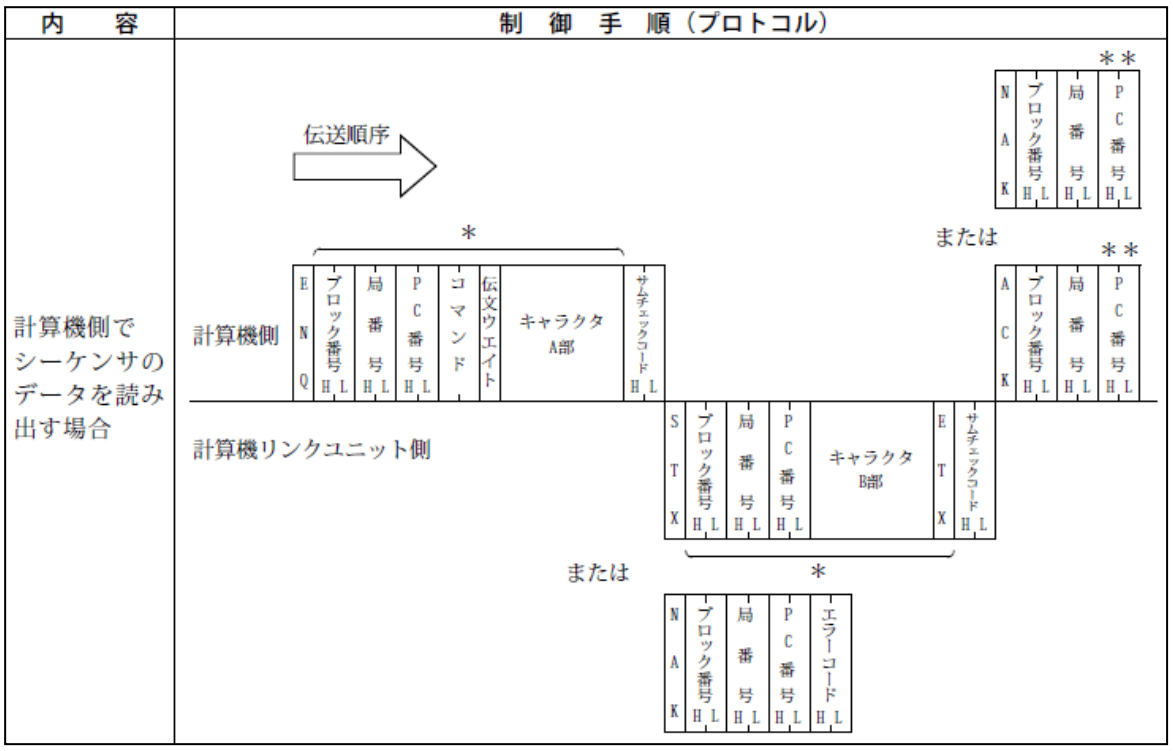
【 form 1 】



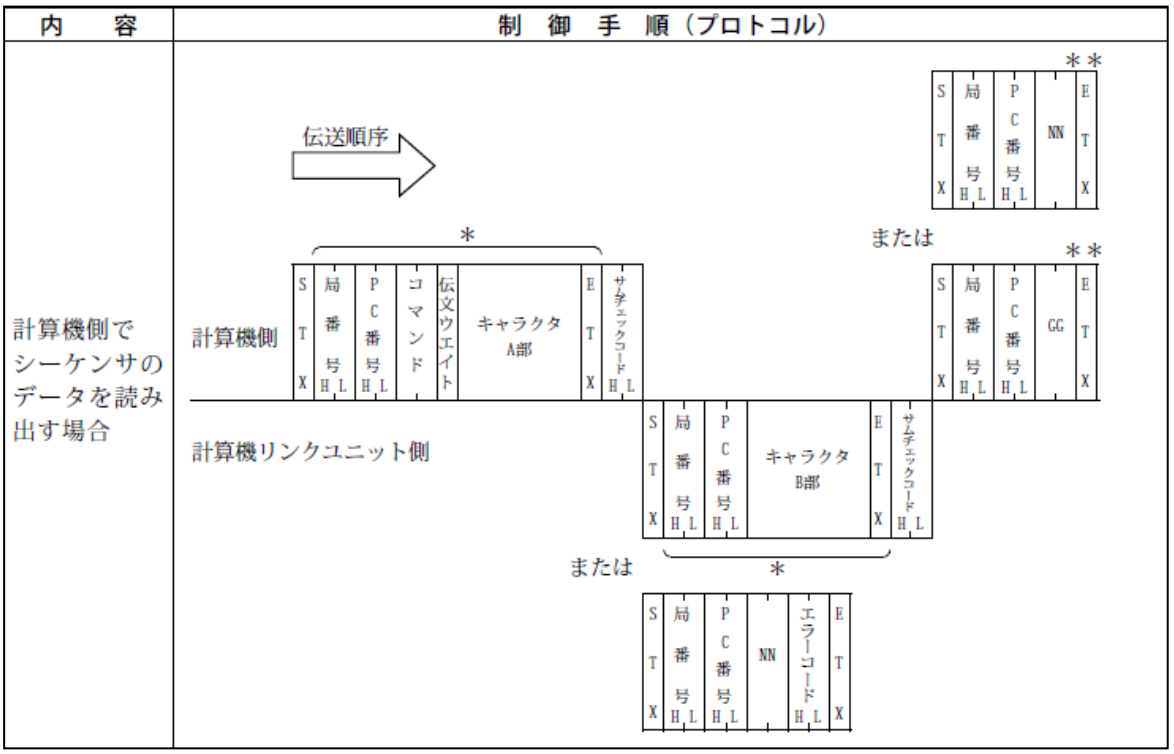
The sum check is done only to the character in * sign part in the above-mentioned chart when setting it to sum check "It is".

When the data of the sequencer is read on the computer side, the transmission of ACK/NAK telegram message in the ** sign part in the above-mentioned chart is omissible.

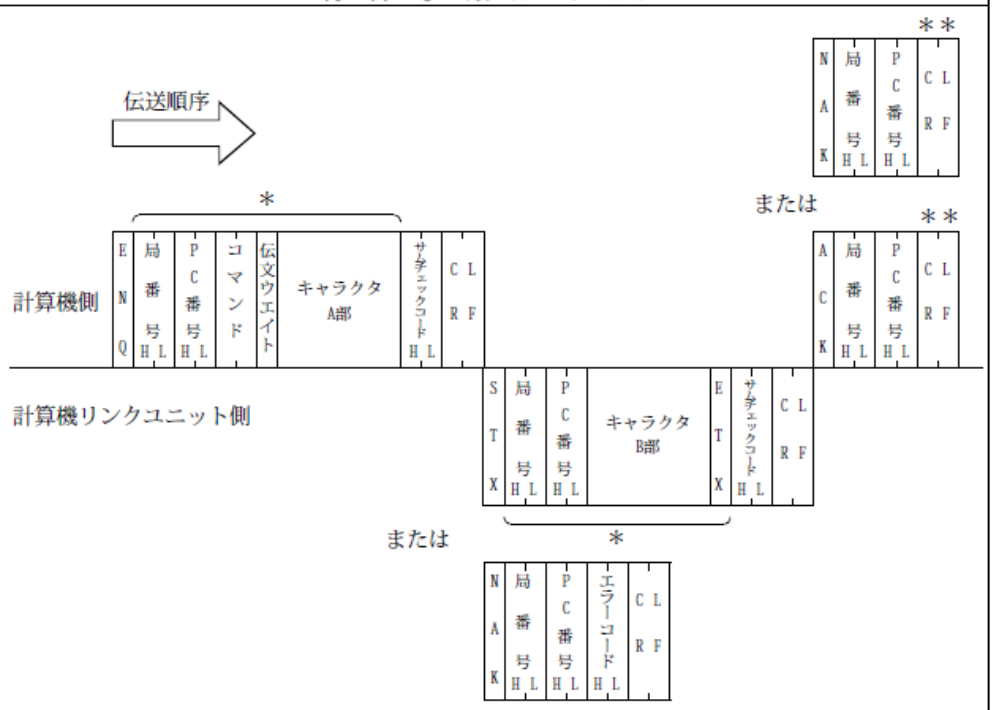
【 form 2 】


内　容
制　御　手　順（プロトコル）
計算機側で
シーケンサの
データを読み
出す場合
伝送順序
＊
＊＊
または
計算機側
計算機リンクユニット側
ENQ
NAK
ACK
STX
ETX
ブロック番号
局番号
PC番号
コマンド
伝文ウエイト
キャラクタ
A部
キャラクタ
B部
サムチェックコード
エラーコード
H L

【 form 3 】


内　容
制　御　手　順（プロトコル）
計算機側で
シーケンサの
データを読み
出す場合
伝送順序
＊
＊＊
または
計算機側
計算機リンクユニット側
STX
ETX
局番号
PC番号
NN
GG
コマンド
伝文ウエイト
キャラクタ
A部
キャラクタ
B部
サムチェックコード
エラーコード
H L

【 form 4 】

内　容	制　御　手　順（プロトコル）
計算機側でシーケンサのデータを読み出す場合	

show is a sample program of AddController.

【Ethernet】

```
    HRESULT hr = S_OK;
    ICaoEngine* pEng = NULL;
    ICaoWorkspaces *pWss = NULL;
    ICaoWorkspace *pWs = NULL;
    ICaoController *pCtrl = NULL;

    // Create CaoEngine
    hr = CoCreateInstance(CLSID_CaoEngine,
                          NULL,
                          CLSCTX_LOCAL_SERVER,
                          IID_ICaoEngine,
                          (void **)&pEng);
    if (FAILED(hr)) {
        goto EndProc;
    }
    // Get CaoWorkspace collection
    hr = pEng->get_Workspaces(&pWss);
    if (FAILED(hr)) {
        goto EndProc;
    }
    // Get CaoWorkspace
    hr = pWss->Item(CComVariant(0L), &pWs);
    if (FAILED(hr)) {
        goto EndProc;
    }

    // Create CaoController
    hr = pWs->AddController(CComBSTR(L"MELSEC_AnA"),
                          CComBSTR(L"CaoProv.MELSEC.AnA"),
                          CComBSTR(L""),
                          CComBSTR(L"Conn=ETH:192.168.2.1:1026"),
                          &pCtrl);
    if (FAILED(hr)) {
        goto EndProc;
    }

    // Put the necessary processing here
    // For example, put value, get value...

EndProc:
    if (pCtrl) pCtrl->Release();
    if (pWs) pWs->Release();
    if (pWss) pWss->Release();
    if (pEng) pEng->Release();
```

【Serial】

```
    HRESULT hr = S_OK;
    ICaoEngine* pEng = NULL;
    ICaoWorkspaces *pWss = NULL;
    ICaoWorkspace *pWs = NULL;
    ICaoController *pCtrl = NULL;

    // CaoEngine の生成
    hr = CoCreateInstance(CLSID_CaoEngine,
                          NULL,
                          CLSCTX_LOCAL_SERVER,
                          IID_ICaoEngine,
                          (void **)&pEng);
    if (FAILED(hr)) {
        goto EndProc;
    }
    // Get CaoWorkspace collection
    hr = pEng->get_Workspaces(&pWss);
    if (FAILED(hr)) {
        goto EndProc;
    }
    // Get CaoWorkspace
    hr = pWss->Item(CComVariant(0L), &pWs);
    if (FAILED(hr)) {
        goto EndProc;
    }

    // Create CaoController
    hr = pWs->AddController(CComBSTR(L"MELSEC_AnA"),
                          CComBSTR(L"CaoProv.MELSEC.AnA"),
                          CComBSTR(L""),
                          CComBSTR(L"Conn=com:7"),
                          &pCtrl);
    if (FAILED(hr)) {
        goto EndProc;
    }

    // Put the necessary processing here
    // For example, put value, get value...

EndProc:
    if (pCtrl) pCtrl->Release();
    if (pWs) pWs->Release();
    if (pWss) pWss->Release();
    if (pEng) pEng->Release();
```

2.3.2. CaoController::AddVariable method

The variable object that write/reads the sequencer data is made.

Format AddVariable(<bstrName:BSTR >,<bstrOption:BSTR>)

bstrName : [in] variable identifier

bstrOption : [in] optional character string

Table2-3Optional character string of CaoController::AddVariable

Option [(2)]	Meaning
Path=< first device >	Indispensability. The first device of the device memory to be accessed is specified by "Device code + device number". (See 2.3.2.1)
Param[=<Variable parameter>]	The parameter of the variable is set. (See 2.3.2.2)
VT[=<Variable type>]	The type of the data used when inputting and outputting to the device memory is specified. (See 2.3.2.3)
Elem[=<Number of elements>]	The number of variables is specified. Please input it by the following formats when specifying it by the hexadecimal number. 0x[0-9,A-F]+, &H[0-9,A-F]+, [0-9,A-F]+H (See 2.3.2.4)
Array[=< True or False >]	It is specified whether to acquire the value in the form of the array at the time of reading one element. (default：FALSE) (See 2.3.2.5)

2.3.2.1. Path is optional.

It can access the sequencer by specifying the device code and the device number for optional Path.

< device >:

Bit device: X, Y, M, B, F, M, and TS, TC, CS, and CC

Word device: TN, CN, D, W, and R

< device number >:= address of variable indicated with device. The address is specified by the decimal number and the hexadecimal number.

- Please make the device number of the bit device a device number that must be able to be divided by 16 when you specify the word unit.

(example) “Path=X1F" It accesses input X15 value.

It accesses "Path=CC255" counter CC255 (coil) value.

Table 2-4 Device list

Device		Device code	Type	Method of addressing
Input		X	Bit	Hexadecimal number
Output		Y		
Internal relay (The latch relay and the step relay are included.)		M		Decimal number
Link relay		B		Hexadecimal number
Annunciator		F		Decimal number
Special relay		M		
Timer	Contact	TS		
	Coil	TC		
	Current value	TN	Word	
Counter	Contact	CS	Bit	
	Coil	CC		
	Current value	CN	Word	
Data register		D		
Link register		W		Hexadecimal number
File register		R		Decimal number
Special register		D		

2.3.2.2. Param is optional.

Connected parameter character string of optional Param is shown as follows. The parameter in the square bracket) shows a possible omission here.

For each element, it is necessary to set the set value by the specified number of digits in hexadecimal. Please fill in the missing digits with 0[4].

【 Ethernet device 】

"[Param=[<PCNo>[:<CPU Timer>]]]"

< PCNo > : PC number(2 digit). Default is FF_H (local station).

< CPU Timer > : CPU watch timer (unit 250ms)(4 digit). Default is 0000_H (infinity wait).

$000A_H$=2.5 second of set value 0001_H=0.25 second.

(example) "Path=X0, Param=FF:0000" The X0 value is acquired.

【 Serial device 】

"[Param=[<StationNo>:<PCNo>:<Wait Time>[:<BlockNo >]]]"

< StationNo > : Exchange number title(2 digit). (00_H～$1F_H$) Default is 00_H.

< PCNo > : PC number(2 digit). Default is FF_H.

< Wait > : telegram message weight(2 digit). (00_H～$0F_H$) Default is $0A_H$ (100ms).

< set value >× of telegram message weight time >=< 10ms

< BlockNo > : Block number(2 digit). Default is 00_H.

(example) The X10 value of "Path=X10" and "Param=01:FF:0A:00" exchange number title 1 is acquired.

[4] Refer to the unit user's manual for the setting values of each element.

2.3.2.3. VT is optional.

The point a read and written data type and element is specified (One point = 1 bit).
When optional VT is omitted, the value of default is set with the device specified for optional Path. When "BIT" specifies the word device when the bit device is specified for optional Path, "I2" reaches the value of default.

"[VT=[< VT optional character string >]]"

(example) "Path=X0000, VT=I2" The value of X000F is read and written from X0000 as word value (2Byte).

Table 2-5 Optional VT that can be specified list

VT is optional.	Data type	Point/number of elements	Meaning
BIT	VT_I2	One point	It reads and writes it by the unit of the bit (Every one point). Note) It is possible to specify it only for the bit device (X, Y, and M etc.).
BOOL	VT_BOOL	One point	It reads and writes it by the unit of the bit (Every one point). Note) It is possible to specify it only for the bit device (X, Y, and M etc.).
I1	VT_I1	Eight points	It reads and writes it in eight points. Note) It treats as an element of the even number piece when the number of elements of the odd number ..optional Elem it.. piece is specified and written, eight added points are written, and it writes it by 0 burials.
UI1	VT_UI1	Eight points	It reads and writes it in eight points. Note) It treats as an element of the even number piece when the number of elements of the odd number ..optional Elem it.. piece is specified and written, eight added points are written, and it writes it by 0 burials.
I2	VT_I2	16 points	It reads and writes it by the unit of the word (Every 16 points).

UI2	VT_UI2	16 points	It reads and writes it in 16 points.
I4	VT_I4	32 points	It reads and writes it in 32 points.
UI4	VT_UI4	32 points	It reads and writes it in 32 points.
R4	VT_R4	32 points	It reads and writes it in 32 points.
R8	VT_R8	64 points	It reads and writes it in 64 points.
BSTR	VT_BSTR	Eight points	The character string of ASCII (One character: 8 bit) is read and written. Note) When a character string that is shorter than the number of elements specified ..optional Elem it.. is written, 0 buries the point of the remainder.

2.3.2.4. Elem is optional.

The number of elements is specified by the decimal number or the hexadecimal number. Please specify the numerical value as it is when specifying it by the decimal number. Please specify &H 0-9 and A-F + or 0-9 and A-F in the form of + H when specifying it by the hexadecimal number 0x 0-9 and A-F +. The value of default when optional Elem is omitted becomes one.

"[Elem=[< number of elements>]]"

(example) "Path=X0, Elem=5" The value of X4 is acquired from X0 as a bit value.
(example) "Path=D10, Elem=0x10" The value of D25 is acquired from D10 as a word value.
(example) "Path=D10, Elem=&H10" The value of D25 is acquired from D10 as a word value.
(example) "Path=D10, Elem=10H" The value of D25 is acquired from D10 as a word value.
(example) "Path=M100, VT=I2, Elem=2" The value of M131 is wording acquired from M100.

2.3.2.5. Array is optional.

The read value is specified when specifying it excluding BSTR and whether it acquires it is specified by one specification ..optional Elem it.. in the form of the array optional VT moreover it. When True is specified, it becomes the form of the data type specified when False is specified in the form of the array. The value of default when optional Array is omitted becomes False.

"[Array=[< True or False >]]"

(example) "Path=X0, VT=BOOL, Elem=1, Array=True" The value of X0 is acquired as an array of the BOOL type.
(example) "Path=X0, VT=BOOL, Elem=1, Array=False" The value of X0 is acquired as BOOL type.

2.3.2.6. Sample program

A sample program of AddVariable is shown below.

Example) Set a value in the array of bit device M (number of elements: 10).

```
HRESULT hr = S_OK;
ICaoEngine* pEng = NULL;
ICaoWorkspaces *pWss = NULL;
ICaoWorkspace *pWs = NULL;
ICaoController *pCtrl = NULL;
ICaoVariable *pVar = NULL;
CComVariant vntGet;

// Create CaoEngine
hr = CoCreateInstance(CLSID_CaoEngine,
                      NULL,
                      CLSCTX_LOCAL_SERVER,
                      IID_ICaoEngine,
                      (void **)&pEng);
if (FAILED(hr)) {
    goto EndProc;
}

// Get CaoWorkspace collection
hr = pEng->get_Workspaces(&pWss);
if (FAILED(hr)) {
    goto EndProc;
}

// Get CaoWorkspace
hr = pWss->Item(CComVariant(0L), &pWs);
if (FAILED(hr)) {
    goto EndProc;
}

// Create CaoController
hr = pWs->AddController(CComBSTR(L"MELSEC_AnA"),
                        CComBSTR(L"CaoProv.MELSEC.AnA"),
                        CComBSTR(L""),
                        CComBSTR(L"Conn=ETH:192.168.2.1:1026"),
                        &pCtrl);
if (FAILED(hr)) {
    goto EndProc;
}

// Create variable
hr = pCtrl->AddVariable(CComBSTR(L"BIT_DEVICE_M"), CComBSTR(L"Path=M16, Elem=10"), &pVar);
if (FAILED(hr)) {
    goto EndProc;
}
```

```
    // Set Value or get value
    CComVariant vntPut
    vntPut.vt     = (VT_I2 | VT_ARRAY);
    vntPut.parray = SafeArrayCreateVector(VT_I2, 0, 10);
    pVar->put_Value(vntPut);
    pVar->get_Value(&vntGet);

EndProc:
    if (pVar) pVar->Release();
    if (pCtrl) pCtrl->Release();
    if (pWs) pWs->Release();
    if (pWss) pWss->Release();
    if (pEng) pEng->Release();
```

2.4. Error code

In the MELSEC-A provider, the following and peculiar the error code is defined. Moreover, "of the ORiN2 commonness error. Please refer to the chapter of the error code of "ORiN2 Programming guide".

Table 2-6 Peculiar error code

Error name	Error number	Explanation
Internal error	0x80100000	Internal error in Ethernet or serial communications.
Sequencer error	0x8010yyxx	If an error occurs in the sequencer, return the error code of the sequencer in yyxx. If xx is 0x5B, the error code is stored at the location of yy. If xx is other than 0x5B, 0x00 is stored in yy. Refer to the sequencer reference for the contents of the error code.
Checksum error	0x80100100	Sum check of message received from sequencer is invalid
Receive data format error	0x80110000	Returns when the message received from the sequencer has an unexpected abnormal format.
Missing received data	0x80110001	Returns when the size of the response packet of the message received from the sequencer does not satisfy the sufficient size.

3. Communication protocol command correspondence table

Table 3-1 shows a list of commands supported by Variable variables implemented by this provider.

Table 3-1 Command correspondence table

function		command	Description	example
Batch reading	Bit	BR	Reads bit devices one by one.	Path=X0, VT=BIT Path=X0, VT=BOOL
	WORD	WR	Read bit device in 16-point units.	Path=X0, VT=I1
			Read the word device one point at a time.	Path=D0, VT=I2
Bulk loading	Bit	BW	Write the bit device one point at a time.	Path=X0, VT=BIT Path=X0, VT=BOOL
	WORD	WW	Write the bit device in units of 16 points.	Path=X0, VT=I1
			Write the word device one point at a time.	Path=D0, VT=I2