2. PLC_CommunicationTo2400S Specifications And Details

This paper describes the actual measurement system created and the recommended method of constructing the system. Project files are located in the following directories.

・“C:¥....¥PLC_CommunicationTo2400S¥ PLC_CommunicationTo2400S.kpr”

2.1. Structure of the created system

[Equipment List]

・PC

・CPU unit: Keyence KV-8000

・Communication unit: Keyence KV-XL202

・Power unit: Keyence CA-U4

・Source meter: Keithley 2400s

・RS-232c cable (Need to cutting)

・USB cable

・Ethernet cable

・3-terminals power cable

Tab 1 Connection Wire Color of KV-XL202

Terminal	Wire Color
SD	Orange
RD	Red
RS	Gray
CS	Purple
ER	Not con.
DR	Not con.
SG	Green

The following Figures of equipment connections (Fig 1) and system structure (Fig 2) are shown below. See Tab 1 for wire color when connecting RS-232c cable to KV-XL202. It is recommended that a continuity check be performed when changing RS-232c cables.

This measurement system consists of a PC, a PLCs CPU unit and Communication unit, and a source meter device. The PC is the user interface, the PLC is the processing unit, and the source meter is controlled by the PLC.

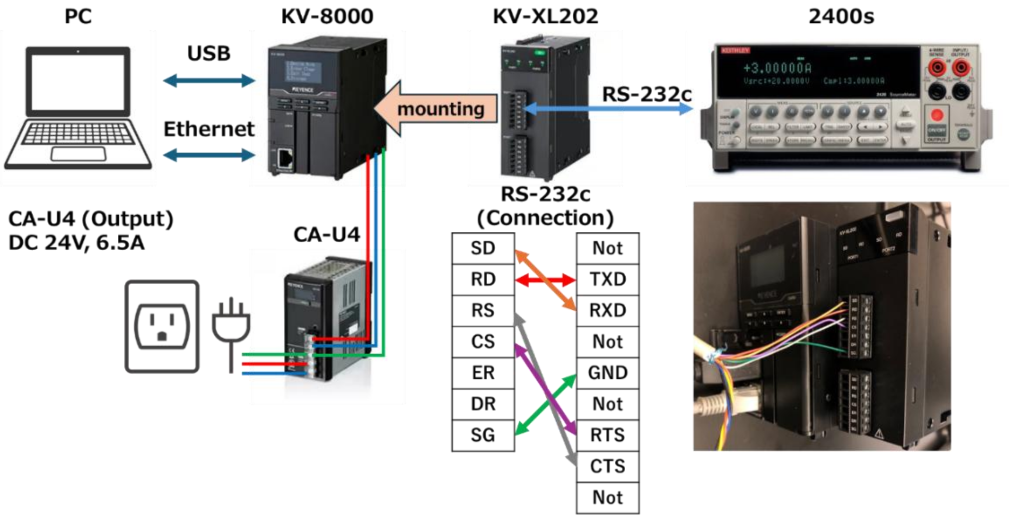


Fig 1 Equipment connection

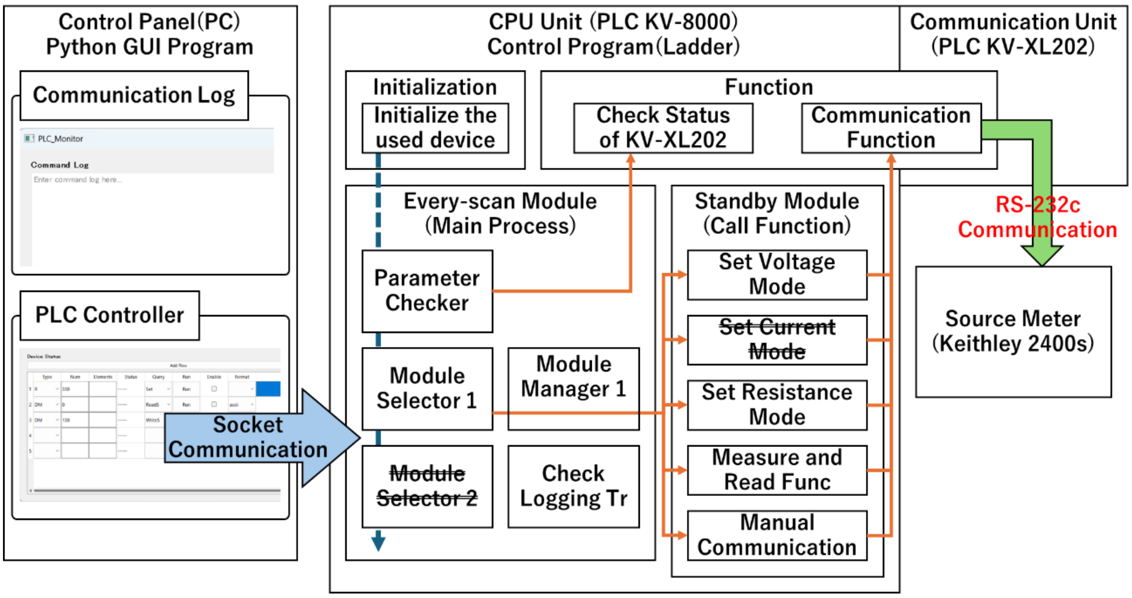


Fig 2 System Structure

[Process Flow]

① PLC: Initialization. Initialize process flags, Data Memory, and device values of KV-XL202.

② PLC: Every-scan module is started. And wait for orders from the user. It also monitors device status for error handling.

③ GUI: Send relay rise command to flag the module startup. (*Host-Link communication can be used to change device values for most PLCs. In other words, a similar system can be built by controlling the ladder program with a Python Program without transferring it to a PLC. However, that would lose the advantage of PLCs. Therefore, it is necessary to properly restrict the devices to which GUI programs have access. This system assumes access only to the start flag and DM for manual communication.)

④ PLC: When the module start flag is raised, check if the module is executable in Module Manager and call the Standby Module.

⑤ PLC: Each Standby Module calls a “Communication Function” to send commands to the 2400s for RS-232c communication.

⑥ PLC: The “Communication Function” handles the sending and receiving of RS-232c communications, and controls the KV-XL202 appropriately. (*Controls the physical devices of the KV-XL202. Therefore, it must be single-threaded and properly processed.)

⑦ 2400s: Processing is performed according to the received commands.

⑧ PLC: Check triggers and perform logging. This system records sending command and receiving command. (*To use the logging function, PLC settings must be made in addition to ladder program.)

PLCs have a very low logic level compared to C++, Python, etc. Therefore, careful specification is necessary. Also, the execution method is not the sequential execution method of C++, Python, etc., but the every scan method, so it is necessary to create programs that take device state transitions into consideration.

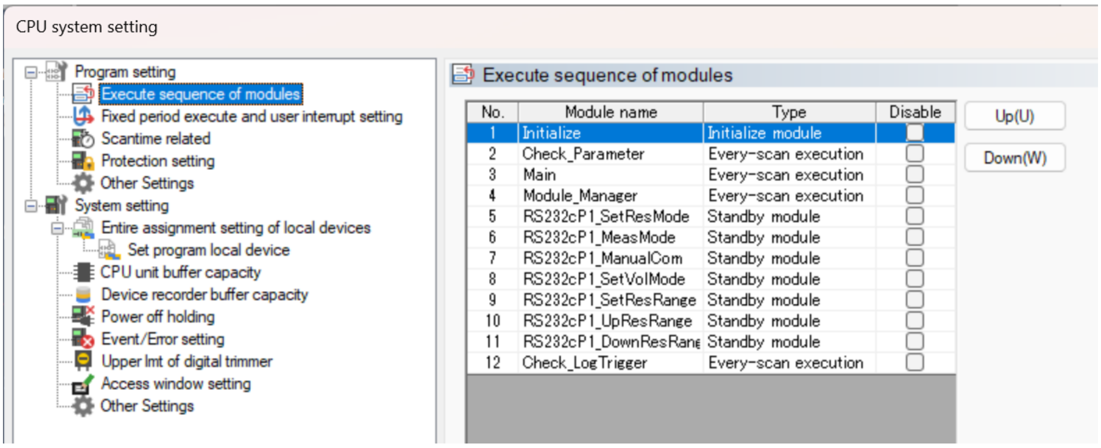
A state transition refers to a change in output values depending on various input conditions. In a ladder program, set modules continue to be scanned. And there is no concept of waiting; it is asynchronous. Therefore, process flags must be used to properly process the program.

2.2. Module execution sequence

The order in which modules are scanned can be set. As the system becomes more complex, this setting becomes more important. The setup method and the Execute sequence of modules of this system (Fig 3) are shown below.

[Setup Method]

① Click on [Program] -> [Setup module execute sequence].
② CPU system setting: Select on [Program setting] -> [Execute sequence of modules].
③ CPU system setting: The order of scanning modules can be set.



CPU system setting

Execute sequence of modules

No.	Module name	Type	Disable
1	Initialize	Initialize module	□
2	Check_Parameter	Every-scan execution	□
3	Main	Every-scan execution	□
4	Module_Manager	Every-scan execution	□
5	RS232cP1_SetResMode	Standby module	□
6	RS232cP1_MeasMode	Standby module	□
7	RS232cP1_ManualCom	Standby module	□
8	RS232cP1_SetVolMode	Standby module	□
9	RS232cP1_SetResRange	Standby module	□
10	RS232cP1_UpResRange	Standby module	□
11	RS232cP1_DownResRang	Standby module	□
12	Check_LogTrigger	Every-scan execution	□

Fig 3 Module execution sequence

2.3. Definition of PLC variables

Similar to C++ and Python, PLC can define global and local variables. This can be used to properly manage process flags and status. As a point of reference, global variables are defined statically, while local variables have the characteristic of being managed dynamically.

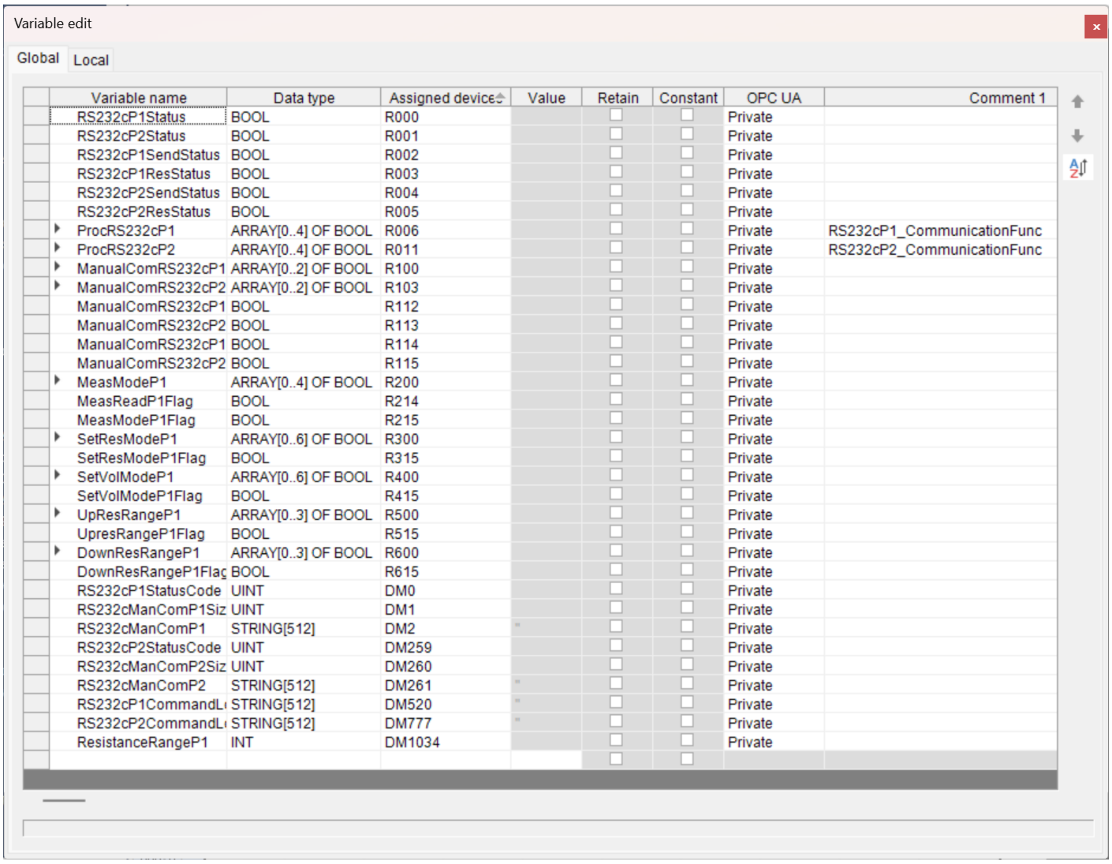
[Setup Method]

① Click on [View] -> [Variable edit window].

② Variable edit: Select the Global or Local tab. (Fig 4)

③ Variable edit: For Global variables, enter the Variable name, Data type, Assigned device.

④ Variable edit: For Local variables, select the program from the downlist and enter the Local Variable name, Data type, Value (initial value).



Variable edit

Variable name	Data type	Assigned device	Value	Retain	Constant	OPC UA	Comment 1
RS232cP1Status	BOOL	R000		☐	☐	Private	
RS232cP2Status	BOOL	R001		☐	☐	Private	
RS232cP1SendStatus	BOOL	R002		☐	☐	Private	
RS232cP1ResStatus	BOOL	R003		☐	☐	Private	
RS232cP2SendStatus	BOOL	R004		☐	☐	Private	
RS232cP2ResStatus	BOOL	R005		☐	☐	Private	
▸ ProcRS232cP1	ARRAY[0..4] OF BOOL	R006		☐	☐	Private	RS232cP1_CommunicationFunc
▸ ProcRS232cP2	ARRAY[0..4] OF BOOL	R011		☐	☐	Private	RS232cP2_CommunicationFunc
▸ ManualComRS232cP1	ARRAY[0..2] OF BOOL	R100		☐	☐	Private	
▸ ManualComRS232cP2	ARRAY[0..2] OF BOOL	R103		☐	☐	Private	
ManualComRS232cP1	BOOL	R112		☐	☐	Private	
ManualComRS232cP2	BOOL	R113		☐	☐	Private	
ManualComRS232cP1	BOOL	R114		☐	☐	Private	
ManualComRS232cP2	BOOL	R115		☐	☐	Private	
▸ MeasModeP1	ARRAY[0..4] OF BOOL	R200		☐	☐	Private	
MeasReadP1Flag	BOOL	R214		☐	☐	Private	
MeasModeP1Flag	BOOL	R215		☐	☐	Private	
▸ SetResModeP1	ARRAY[0..6] OF BOOL	R300		☐	☐	Private	
SetResModeP1Flag	BOOL	R315		☐	☐	Private	
▸ SetVolModeP1	ARRAY[0..6] OF BOOL	R400		☐	☐	Private	
SetVolModeP1Flag	BOOL	R415		☐	☐	Private	
▸ UpResRangeP1	ARRAY[0..3] OF BOOL	R500		☐	☐	Private	
UpresRangeP1Flag	BOOL	R515		☐	☐	Private	
▸ DownResRangeP1	ARRAY[0..3] OF BOOL	R600		☐	☐	Private	
DownResRangeP1Flag	BOOL	R615		☐	☐	Private	
RS232cP1StatusCode	UINT	DM0		☐	☐	Private	
RS232cManComP1Siz	UINT	DM1		☐	☐	Private	
RS232cManComP1	STRING[512]	DM2	""	☐	☐	Private	
RS232cP2StatusCode	UINT	DM259		☐	☐	Private	
RS232cManComP2Siz	UINT	DM260		☐	☐	Private	
RS232cManComP2	STRING[512]	DM261	""	☐	☐	Private	
RS232cP1CommandL	STRING[512]	DM520	""	☐	☐	Private	
RS232cP2CommandL	STRING[512]	DM777	""	☐	☐	Private	
ResistanceRangeP1	INT	DM1034		☐	☐	Private	
				☐	☐		

Fig 4 PLCs Global Variables

2.4. PC network settings

The GUI program requires PC network configuration for network communication.

[Setup Method]

① Launch the PC settings and open the Ethernet settings in the Network & internet section.

② Click on the Edit button of IP Assignment to open it and select Manual.

③ Set IPv4 to On and enter the following settings. (Fig 5)

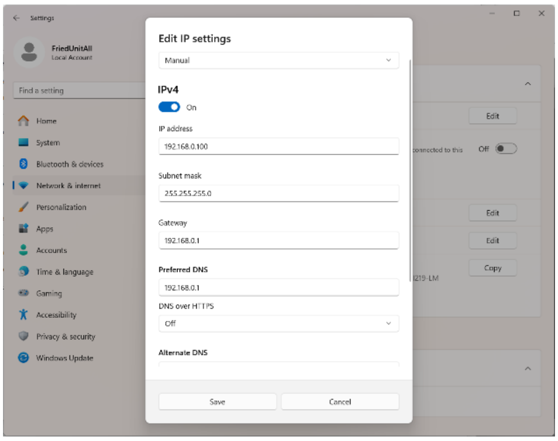
・IP address: 192.168.0.100 (Match the network settings of the KV-8000. 192.168.0.n)

・Subnet mask: 255.255.255.0 (Match the network settings of the KV-8000.)

・Gateway: 192.168.0.1 (Match the network settings of the KV-8000. 192.168.0.n)

・Preferred DNS: 192.168.0.1(Match the network settings of the KV-8000. 192.168.0.n)
・DNS over HTTPS: Off (default)
・Alternate DNS: *Do not fill in (default)
・DNS over HTTPS: Off (default)

④ Click on Save. And close Settings.



Edit IP settings

Manual

IPv4

On

IP address: 192.168.0.100

Subnet mask: 255.255.255.0

Gateway: 192.168.0.1

Preferred DNS: 192.168.0.1

DNS over HTTPS: Off

Alternate DNS

Fig 5 Network Configuration

2.5. Host-Link Communication by Python script

To build a user interface for the PLC system, a library for Host-Link communication and library for GUI creation were imported in Python, and a control monitor was created. The KV-8000 can perform Host-Link communication as an EtherNet/IP function. This function is used to override and control device values provided in the KV-8000.

The main Python program is located in the following directory.

“C:¥....¥PLC_CommunicationTo2400S¥PLC_ControlMonitor¥PLC_MonitorMain.py”

[How to use]

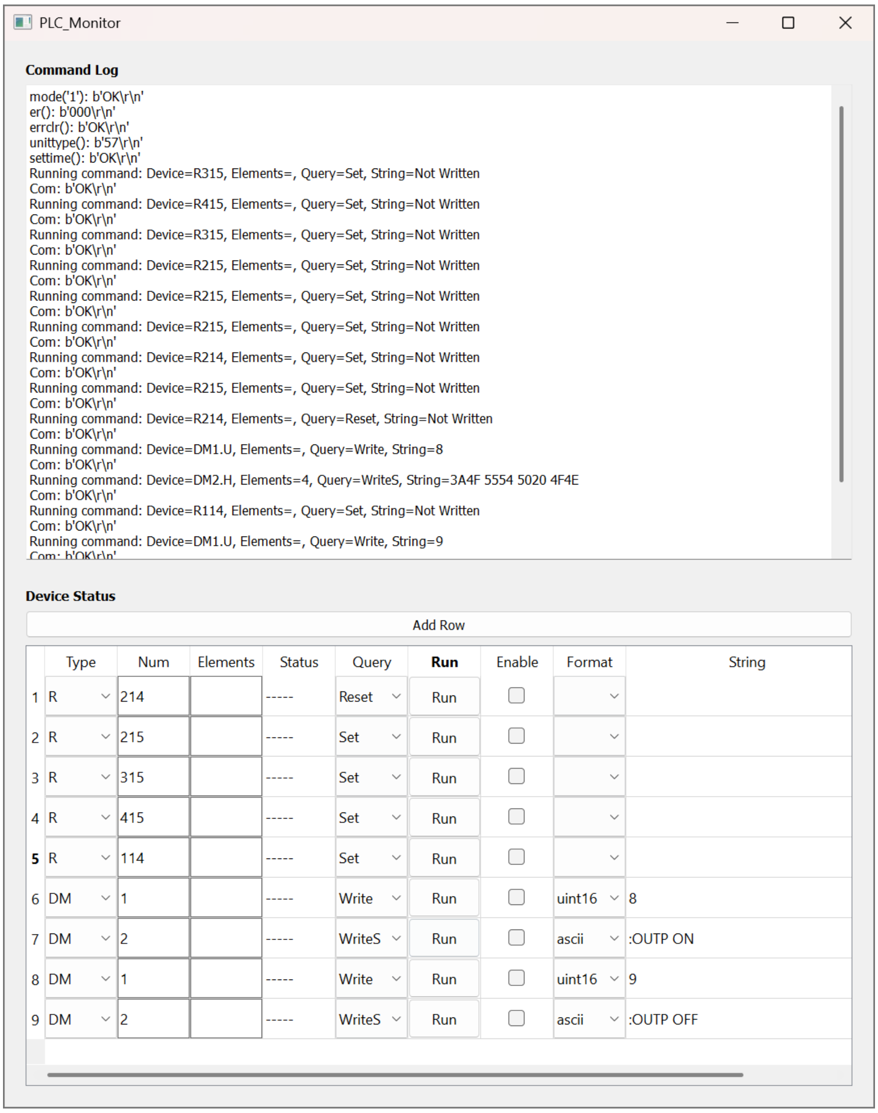
① Launch a terminal and execute the following command.

“¥PLC_CommunicationTo2400S¥PLC_ControlMonitor> python .¥PLC_MonitorMain.py”

② Fill in the command settings in the table at the bottom of the window, as shown in Fig 6. (*To control a Relay, enter R in the Type column, the number of the Relay to be controlled in the Num column, and Set (Rise) or Reset (Fall) in the Query column, and click Run.)

③ As shown in Tab 2, each device corresponds to a global variable in the PLC. Therefore, by clicking on Run and setting the start flag, the Standby Module can be executed.

④ To use the manual communication function, set the command settings as shown in rows 6~7 of the table in Fig 6, and set the command and command length in DM. Then, the “ManualComRS232cP1Flag” flag is set and executed.



```
mode('1'): b'OK\r\n'
er(): b'000\r\n'
errclr(): b'OK\r\n'
unittype(): b'57\r\n'
settime(): b'OK\r\n'
Running command: Device=R315, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=R415, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=R315, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=R215, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=R215, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=R215, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=R214, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=R215, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=R214, Elements=, Query=Reset, String=Not Written
Com: b'OK\r\n'
Running command: Device=DM1.U, Elements=, Query=Write, String=8
Com: b'OK\r\n'
Running command: Device=DM2.H, Elements=4, Query=WriteS, String=3A4F 5554 5020 4F4E
Com: b'OK\r\n'
Running command: Device=R114, Elements=, Query=Set, String=Not Written
Com: b'OK\r\n'
Running command: Device=DM1.U, Elements=, Query=Write, String=9
Com: b'OK\r\n'
```

	Type	Num	Elements	Status	Query	Run	Enable	Format	String
1	R	214		-----	Reset	Run	☐		
2	R	215		-----	Set	Run	☐		
3	R	315		-----	Set	Run	☐		
4	R	415		-----	Set	Run	☐		
5	R	114		-----	Set	Run	☐		
6	DM	1		-----	Write	Run	☐	uint16	8
7	DM	2		-----	WriteS	Run	☐	ascii	:OUTP ON
8	DM	1		-----	Write	Run	☐	uint16	9
9	DM	2		-----	WriteS	Run	☐	ascii	:OUTP OFF

Fig 6 Control Monitor Window

Tab 2 Module flags and control monitor settings for PLCs

PLC variable name	Type	Num	Query	Format	String
MeasReadP1Flag	R	214	Reset		
MeasModeP1Flag	R	215	Set		
SetResModeP1Flag	R	315	Set		
SetVolModeP1Flag	R	415	Set		
ManualComRS232cP1Flag	R	114	Set		
RS232cManComP1Size	DM	1	Write	uint16	8
RS232cManComP1	DM	2	Writes	ascii	:OUTP ON
RS232cManComP1Size	DM	1	Write	uint16	9
RS232cManComP1	DM	2	Writes	ascii	:OUTP OFF

[Note.]

・When using “plural form” instruction, the array length must be entered in the Elements column. For example, Sets, Resets, Writes, Reads, etc.

・The Host-Link communication functions of the Python program are described in the following file. Socket library is required.

“¥PLC_CommunicationTo2400S¥PLC_ControlMonitor> python .¥PLC_CommandClass.py”

・For information on creating “PLC_CommandClass.py”, see the manual below.

“C:¥....¥PLC_CommunicationTo2400S¥Manual¥English¥EtherNetsIP_Function_KV-EP21V_KV-8000(A)etc_User's_Manual.pdf”

-> Section 8 HOST-LINK COMMUNICATION FUNCTION

・GUI library used PyQt5 version 5.15.11.

2.6. FTP server function

Set up an FTP server function to access the PLC’s internal storage and the SD card inserted in the KV-8000. Using this function, log data and other data can be comfortably obtained.

[How to use]

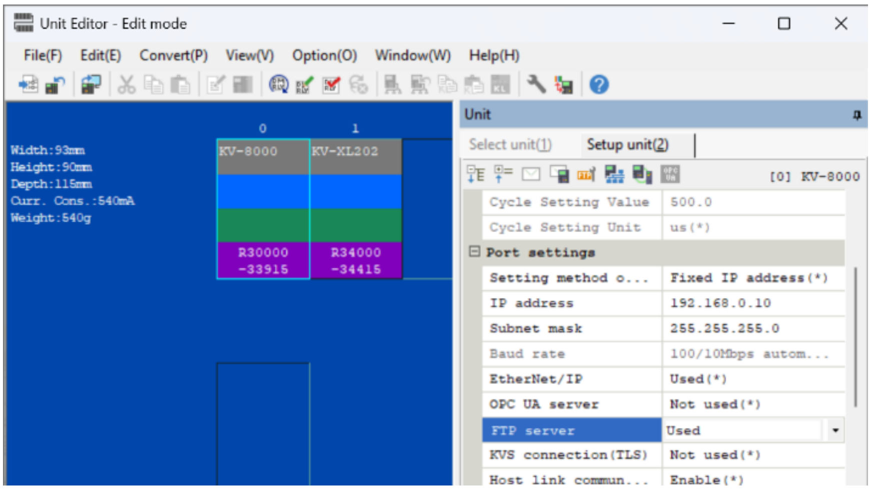
① As shown in Fig 7, select KV-8000 in the Unit Editor and set the “FTP server” to “Used”.

② After setting up the FTP server, transfer the settings and program to the PLC.

③ After confirming that the PC and KV-8000 are connected via EtherNet cable, launch Explorer of PC as shown in Fig 8 and enter the following command to access the system. “ftp://192.168.0.10”, (*Enter the IP address of the KV-8000 in Fig 7)

④ When the logon dialog appears, enter KV for the username and logon without a password.

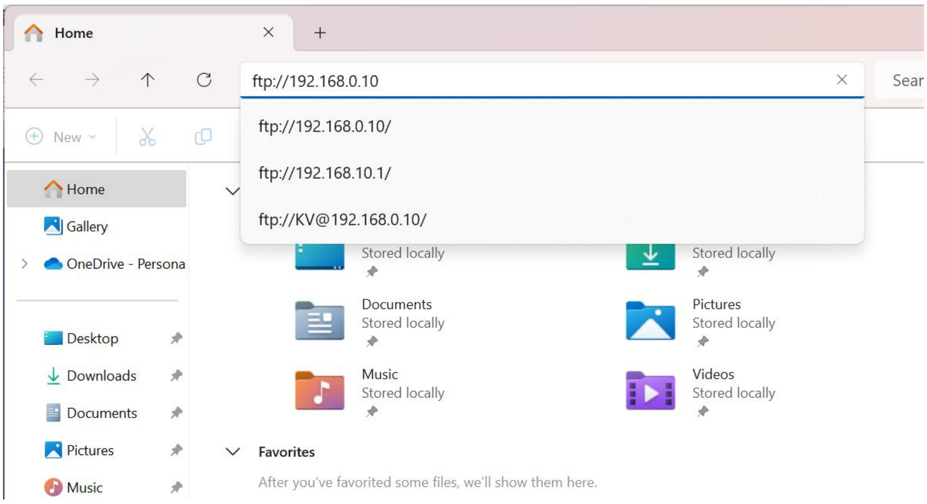
⑤ Access to PLC and SD card storage to view files.



Width:93mm
Height:90mm
Depth:115mm
Curr. Cons.:540mA
Weight:540g

Cycle Setting Value	500.0
Cycle Setting Unit	us(*)
⊟ Port settings	
Setting method o...	Fixed IP address(*)
IP address	192.168.0.10
Subnet mask	255.255.255.0
Baud rate	100/10Mbps autom...
EtherNet/IP	Used(*)
OPC UA server	Not used(*)
FTP server	Used
KVS connection(TLS)	Not used(*)
Host link commun...	Enable(*)

Fig 7 Enable FTP Server Settings



ftp://192.168.0.10

ftp://192.168.0.10/

ftp://192.168.10.1/

ftp://KV@192.168.0.10/

Fig 8 Logon to KV-8000

2.7. Logging function

Set up a logging function to obtain experimental data and operation records. Operation records can be used to debug the program.

[Setup Method]

① Click on [Tool] -> [Setup logging/trace].

② Logging/Trace setting list: Click on Set. This system was set up as follows. (Fig 9)

・Function select: Logging

・Saving location: Memory card

・File comment: SendCommandLog

・File name, File No.: Auto numbering

・File name, No. upper limit: 99

・File name, Start operation: New a file

・File name, When no empty No.: Overwrite by the oldest files

・File name, String: SendCommandLog

・File name, Date/time: Year, Month, Day

・Device/variable, Device/variable: RS232cP1CommandLog

・Device/variable, Word Pts: 255

・Device/variable, Data format: ASCII16BIT

・Trigger, Type: Bit device

・Trigger, Device: RS232cP1SendStatus

・Option, Insert file comments into the leading of CSV file: Enable

・Option, Add comments row of device: Enable

・Option, Attach time stamp to each row: Enable

・Option, Add data No. to each row: Enable

・Option, Add data obtain span to each row: Enable

・Option, Add space to secure the character count: Disable

・Option, Perform CSV file save for each trigger: Enable

・Option, Save the file to non-volatile memory: Enable

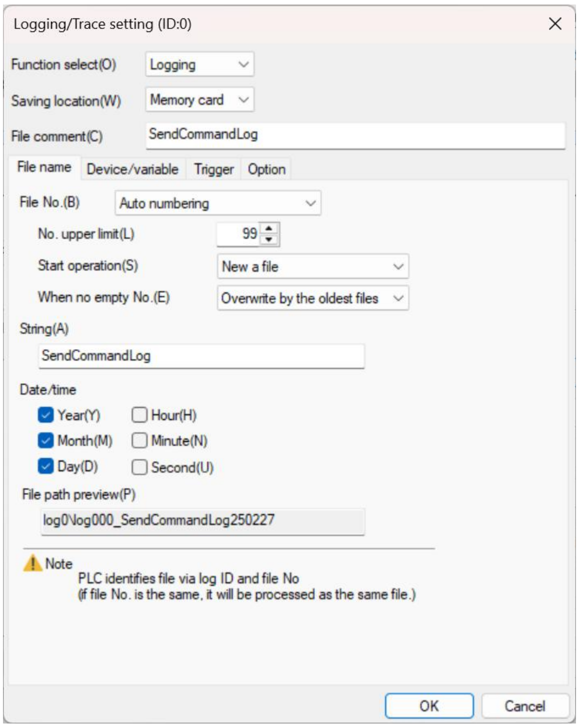
・Option, Auto-restart logging by inserting memory card: Enable

・Option, Set upper limit of file capacity: Enable

・Option, Upper limit: 1000, Row

③ The logging function for reception was also set up with reference to ②

④ In this setup, logging continues as long as the Bit device (Relay) is up, so a "Check Logging Trigger" module was add for the logging function.



Function select(O): Logging

Saving location(W): Memory card

File comment(C): SendCommandLog

File No.(B): Auto numbering

No. upper limit(L): 99

Start operation(S): New a file

When no empty No.(E): Overwrite by the oldest files

String(A): SendCommandLog

Date/time

☑ Year(Y) ☐ Hour(H)

☑ Month(M) ☐ Minute(N)

☑ Day(D) ☐ Second(U)

File path preview(P): log0\log000_SendCommandLog250227

⚠ Note
PLC identifies file via log ID and file No
(If file No. is the same, it will be processed as the same file.)

Fig 9 Loggint/Trace setting

2.8. Program module type

This section describes the "Initialize module", "Every-scan module", "Standby module", and "Function/Function Block" used to build this system. For specifications of each module, refer to the following manual.

“C:¥….¥PLC_CommunicationTo2400S¥Manual¥English¥Programmable_Controller_KV-8000_User's_Manual.pdf”
-> Section 4-9 Module Type

2.9. Description of each module

2.9.1. Initialize the used device

[Process Flow]

① Execute “Initialization of KV-XL202” Function.

2.9.2. Parameter Checker

[Process Flow]

① Execute “Check Status of KV-XL202” Function.

2.9.3. Module Selector

[Process Flow]

① Monitor process startup flags.
② Once the process flag is raised, check to see if the module can be started.
③ If the module can be activated, activate the corresponding module.

2.9.4. Module Manager

[Process Flow]

① Monitor process startup flags.
② If the module flag memory is not Busy, enter the corresponding Busy code.
③ When the process flag goes down, check to see if the busy code corresponds and clear the busy status.

2.9.5. Check Logging Trigger

[Process Flow]

① Monitor the corresponding relay to see if it comes up.
② Copy the command to the DM for logging at the timing when the corresponding relay comes up, and start up the trigger for logging.

2.9.6. Set Resistance Mode

[Process Flow]

① Initialize the process flags when called by the Module selector in the Main program.

② Invoke and execute “Communication Function” Function with the following command length and command order.

・4, “*RST”

・16, “:SENS:FUNC “”RES”””

・18, “:SENS:RES:RANG 2E8”

・23, “:SENS:RES:RANG:AUTO OFF”

・14, “:SYST:RSEN OFF”

③ Reset process flags.

④ Start flag is lowered.

2.9.7. Set Voltage Mode

[Process Flow]

① Initialize the process flags when called by the Module selector in the Main program.

② Invoke and execute “Communication Function” Function with the following command length and command order.

・4, “*RST”

・38, “:SOUR:FUNC CURR; :SOUR:CURR:MODE FIXED” (*Multiple commands can be sent like this.)

・58, “:SENS:FUNC “”VOLT””; :SENS:VOLT:PROT 30; :SENS:VOLT:RANG 200”

・37, “:SOUR:CURR:RANG MIN; :SOUR:CURR:LEV 0”

③ Reset process flags.

④ Start flag is lowered.

*This program sends commands together, but it is not necessary to send them all at once. Rather, it is easier to debug if the commands are sent one at a time.

2.9.8. Measure and Read Func

[Process Flow]

① Initialize the process flags when called by the Module selector in the Main program.

② Execute “Communication Function” Function with a command length and command (8, “:OUTP ON”).

③ Set the “ResRequest” flag and execute “Communication Function” Function with a command length and command (8, “:OUTP ON”).

④ When the receive flag is raised, the received command is copy to the DM.

⑤ If the “MeasReadP1Flag” flag is ON, processes ③ and ④ are repeated.

⑥ Execute “Communication Function” Function with a command length and command (9, “:OUTP OFF”).

⑦ Reset process flags.

⑧ Start flag is lowered.

2.9.9. Manual Communication

This module uses the Python scripts from Section 2.5. See there for command length and how to set up and run commands.

[Process Flow]

① Initialize the process flags when called by the Module selector in the Main program.

② Execute “Communication Function” Function using any command length and command set in the global variables.

③ Reset process flags.

④ Start flag is lowered.

2.9.10. Initialization of KV-XL202

[Process Flow]

① When this function is called, it initializes Relay, DM and process flags used in the communication process.

2.9.11. Check Status of KV-XL202

This ladder program is used to monitor the KV-XL202 communication processing device in KV-Studio. It has absolutely no impact on the system with respect to basic processing.

2.9.12. Communication Function

This program is the core function in communication. In order to control physical devices, it is recommended that modules be created in “Function” rather than “Function Block”. The process flow of this system program is described below.

[Process Flow]

① When this function is called, make sure that the busy flag is not raised.

② Set the busy flag and initialize the process flags.

③ Initialize Relay and DM occupied by the physical device of KV-XL202.

④ Scan the KV-XL202 read-only Relay to confirm that initialization is complete.

⑤ Set “R34200” to activate the communication device.

⑥ Check “R34300” to confirm that the communication device is up.

⑦ Once the device is up and running, set the DM command length and command as follows, start “R34201” and send the command.

・Command Length: DM10310

・Command: DM10310~DM10566 (512 Byte)

⑧ When the transmission completion relay “R34301” rises, the transmission relay “R34201” is turned down.

⑨ When a receive request is made, the receive flag “R34302” rises, the command length and command are copied from the DM as follows, and the receive completion relay “R34202” rises.

・Command Length: DM10567

・Command: DM10568~DM10823

⑩ Reset process flags. And Reset busy flags. Also If “R34302” is low, “R34202” is lowered.

2.10. References

[Project Repository]

・https://github.com/shimasou/PLC_CommunicationTo2400S

[PLC Manuals]

C:¥....¥PLC_CommunicationTo2400S¥Manual¥English¥Serial_Communication_Unit_KV-XL202sXL402_User's_Manual.pdf -> Section 10

C:¥....¥PLC_CommunicationTo2400S¥Manual¥English¥Programmable_Controller_KV-8000_User's_Manual.pdf -> Section 4-9 Module Type

C:¥....¥PLC_CommunicationTo2400S¥Manual¥English¥EtherNetsIP_Function_KV-EP21V_KV-8000(A)etc_User's_Manual.pdf-> Section 8 HOST-LINK COMMUNICATION FUNCTION