FeSCADA & Kinco PLC

Introduction

An application was done to show the possibilities of FeSCADA. By the end of this paper the reader will learn how FeSCADA can work with Kinco PLCs.

- 1. Description
- 2. Hardware
- 3. RS485 serial communication
- 4. Kinco PLC Modbus registers
- 5. FeMODBUS communication setup
- 6. FeSCADA project
- 7. Conclusions

1) Description

In the following pages an application is developed for Modbus communication and data exchange with a Kinco K506 PLC. The final goal is to be able to control two stepper motors from FeSCADA.

2)Hardware

The hardware is composed of one Kinco K506 PLC. The Kinco K5 series provides many diverse functions and is a cost-effective micro integrated PLC. The Kinco K5 series PLC line has different base units to choose from, each has 14 built-in inputs and 10 built-in outputs (relays or transistors). Users can connect up to 6 expansion modules offering up to 136 I/O points.



More information at: _

https://www.kincoautomation.com/products/plc/K5_Series/K506_CPUs/

3) RS485 serial communication

The Kinco K506 PLC has three serial communication ports. Port0 is used primarily for programming and is RS232. Port1 and Port2 are only RS485 and are used for peripheral serial connections: other PLCs, HMI, SCADA, etc.

The setup of the Port2 is done in the programming stage of the PLC with the KincoBuilder software. The PLC was programmed to work as a Modbus RTU server with the address = 10, at 38400 baud, no parity, 8 data and 1 stop bits.

E 2 0 0 € 2 0 0 × 0	間間間の U		e e 🕂 🖓 📳	1 Hol HH HH HT -D - 関 1 🛠	" ₽ ₽
Project: Steve02	Module	I Address	Q Address	Comment	
	▶ 1 K506-24DT	01	01	CPU506, DC24V Power Suply, DI 14*D	C24V, [
	2		III		•
	Address	s: 1 ▼ s: 19200 ▼	Address: 1	Address: 10 Baudrate: 38400	•

On the PC side we used an optically isolated USB to RS485 adapter, B&B Electronics USOPTL4-LS.



Hardware prices.

Name	Unit price	Qty	Price	Description
KNC-PLC-K506-24DT	\$137.00	1	\$137.00	K5 PLC Kinco Automation
USOPTL4-LS	\$143.00	1	\$143.00	USB to RS485 Adapter
	тс	DTAL =	\$280.00	

4) Kinco PLC Modbus registers

The following table is showing the Modbus register numbers for various data areas in the Kinco K5 PLC: digital inputs, digital outputs, memory bits, analog inputs, analog outputs, free memory data.

Data type	Symbol first number - bytes second number - bits	Modbus register type	Modbus register number
Inputs	%10.0 1%31.7	Digital inputs	0 255
Outputs	%Q0.0 %Q31.7	Coils	0 255
Memory bits	%M0.0 %M31.7	Digital inputs/Coils	320 8511
Analog Inputs	%AIW0 %AIW62	Input registers	0 31
Analog Outputs	%AQW0 %AQW62	Holding registers	0 31
Memory words	%VW0 %VW4097	Input/Holding registers	100 2147

5) FeMODBUS communication setup

FeMODBUS software is used to connect to Kinco K5 PLC.

In the picture on the right the serial connection settings are presented.

COM17 was assigned for our USB to RS485 adapter by Windows, when we plugged in the adapter. The other settings should match the settings for COM Port2 on the Kinco PLC.

The next settings are used to connect to the server and to send requests.

We connect to our own PC, which it is a Modbus

	PTU Server COM settings
Ter berver com soungs	HTO Server COM secongs
Port COM17 -	Port NONE -
Baud 38400 -	Baud 19200 -
Parity none 💌	Parity none
Bits-data 8 🔹	Bits-data 8
Bits-stop 1	Bits-stop 1
502 TCP Server port	t 1 RTU Server address
V Accept Broadcast Conne	ections Tags File Name
	nections Modbus Tags by
o Broadcast - Accepted Conne	moubus_ r dgs.twt
o Broadcast - Accepted Conne Type IP address	SS A VAuto load at statt-up
o Broadcast - Accepted Conne Type IP address	SS Auto load at start-up File
o Broadcast - Accepted Conne Type IP address	SS Auto load at start-up File
o Broadcast - Accepted Conne Type IP address	SS Auto load at start-up File Client File Name Modbus_Clients.txt

TCP server, and we send requests to the address 10. Because the address 10 is less than 255, the server will send them onto the serial COM port assigned for TCP server, and it will send back the answers.

Serve	ers List				RSN = R	emote Serve	er Numbe	er							
Serve	erName	Serv	verIP	Server	LocalIP	LocalPort	t RSN	Active	Connec	. Error		TCP/IP Conne	ection	Modbus Request Par	ameters
Remo	ote_Ma	192.16	8.10	502	192.168.1.3	1025	1	Yes	No	0100		My IP Addre	0.1	3	
Kinco	PLC	127.	0.0.1	502	127.0.0.1	1025	2	Yes	Yes	0000		127.0	.0.1		-
Adva	ntech	192.16	8.10	502	127.0.0.1	1027	3	Yes	No	0100		My Port	1025	Unit Address	10
	and all the second											Server IP Ad	dress	Remote Register	200
					ICP/IP connec	tion error bit	is: Closed	d - Open Faile	ed - No Ansv	ver - Socket	Error	127 . 0 .	0.1	, iddi coo	-
Requ	iests List	Sel	ect a serv	er to see th	ie assigned req	uests list								How Many Registers?	10
RN	Func	Uni	RegAdd	ir RegN	o Offset	Active (Cyclical	CycleTime	Error	StatusBits	-	Server Port	502	Local Register	0
1	3	10	200	10	0	Yes	Yes	300	0100	100	-	Server Name		Address	
2	16	10	200	10	10	Yes	Yes	100	0100	100		Kinco PLC		Cyclical? Yes/No	Yes 🔻
3	2	10	320	16	0	Yes	Yes	300	0100	100		hance j ee			
4	2	10	400	16	20	Yes	Yes	300	0100	100	-	Add Modi	ify Del	Cycle Time [ms]	300
_						Error bit: Da	ata - Exc	eption - MBA	P - Time out	0-6	a a la			Active	
Holdi	ng Registe	rs	Inp	ut Register:	S	Message Sta	atus: Ser	nt - Answere	d - Processe	d	esn			Add Modify	Del
0			-			16	Digital In	puts (Read C	inly)						
0	0		0	_	- 7										
1	0		1	0		,									
2	0		2	0		16	Digital Co	oils (Set/Rese	t)						Save in f
3	0		3	0	0						-				

uests List	Sel	Select a server to see the assigned requests list								
Func	Uni	RegAddr	RegNo	Offset	Active	Cyclical	CycleTime	Error	StatusBits	-
2	10	320	16	0	Yes	Yes	300	0100	100	
2	10	400	16	20	Yes	Yes	300	0100	100	
15	10	329	10	0	Yes	Yes	100	0100	100	=
16	10	250	10	20	Yes	Yes	100	0100	100	-
	Func 2 2 15 16	Juests List Sel Func Uni 2 10 2 10 15 10 16 10	Select a server Func Uni RegAddr 2 10 320 2 10 400 15 10 329 16 10 250	Select a server to see the a Func Uni RegAddr RegNo 2 10 320 16 2 10 400 16 15 10 329 10 16 10 250 10	Select a server to see the assigned red Func Uni RegAddr RegNo Offset 2 10 320 16 0 2 10 400 16 20 15 10 329 10 0 16 10 250 10 20	Select a server to see the assigned requests listFuncUniRegAddrRegNoOffsetActive210320160Yes2104001620Yes1510329100Yes16102501020Yes	Select a server to see the assigned requests listFuncUniRegAddrRegNoOffsetActiveCyclical210320160YesYes2104001620YesYes1510329100YesYes16102501020YesYes	Select a server to see the assigned requests listFuncUniRegAddrRegNoOffsetActiveCyclicalCycleTime210320160YesYes3002104001620YesYes3001510329100YesYes10016102501020YesYes100	Select a server to see the assigned requests list Func Uni RegAddr RegNo Offset Active Cyclical Cycle-Time Error 2 10 320 16 0 Yes Yes 300 0 1 0 0 2 10 400 16 20 Yes Yes 300 0 1 0 0 15 10 329 10 0 Yes Yes 100 0 1 0 0 16 10 250 10 20 Yes Yes 100 0 1 0 0	Select a server to see the assigned requests list Func Uni RegAddr RegNo Offset Active Cyclical CycleTime Error StatusBits 2 10 320 16 0 Yes Yes 300 0100 100 2 10 400 16 20 Yes Yes 300 0100 100 15 10 329 10 0 Yes Yes 100 0100 100 16 10 250 10 20 Yes Yes 100 0100 100

The pictures above are showing that FeMODBUS was setup to connect at the address 127.0.0.1, which is our own PC address, and to cyclically (every 300 and 100 ms) send some of the possible requests:

- Function 2 read digital inputs
- Function 3 read holding registers
- Function 15 write coils (digital outputs)
- Function 16 write holding registers

After setting up all the requests we defined tags which we link with different registers on the local computer. The tags are used for DDE communication between FeMODBUS and FeSCADA. All the tags are assigned to the Remote Server Number 2 (RSN 2). See picture below.

Index	TagName	DataType	UpdateType	RSN	RegType	RegAddr	Value	. 1
13	Kinco_Accel1	Word16	Write	2	Holding Register	13	0	
14	Kinco_Accel2	Word16	Write	2	Holding Register	23	0	
15	Kinco_Break	Byte8	Write	2	Digital Coil	6	0	
16	Kinco_Cmd1	Byte8	Read	2	Digital Input	9	0	
17	Kinco_Cmd2	Byte8	Read	2	Digital Input	10	0	
18	Kinco_Distance1	Int32	Write	2	Holding Register	14	0	
19	Kinco_Distance2	Int32	Write	2	Holding Register	24	0	
20	Kinco_Jog1	Byte8	Write	2	Digital Coil	1	0	
21	Kinco_Jog2	Byte8	Write	2	Digital Coil	4	0	
22	Kinco_Jog_Dir1	Word 16	Write	2	Holding Register	18	0	
23	Kinco_Jog_Dir2	Word 16	Write	2	Holding Register	28	0	
24	Kinco mc11	Bvte8	Read	2	Digital Input	21	0	
Tag N	lame	Data Typ	oe Upd	ate Type	a		Data view	
Kinco	_Accel1	Word 16	▼ Wri	te	•	Add	Sig.Dec	
	RSN (Server numbe	r)			U	pdate		
	2	Register	Туре					
	Register Address	Holding	Holding Register					el
	13					Save		

Frandos Engineering LLC

Index	TagName	DataType	UpdateType	RSN	RegType	RegAddr	Value	-
20	Kinco_Jog1	Byte8	Write	2	Digital Coil	1	0	
21	Kinco_Jog2	Byte8	Write	2	Digital Coil	4	0	
22	Kinco_Jog_Dir1	Word 16	Write	2	Holding Register	18	0	
23	Kinco_Jog_Dir2	Word 16	Write	2	Holding Register	28	0	
24	Kinco_mc11	Byte8	Read	2	Digital Input	21	0	
25	Kinco_mc22	Byte8	Read	2	Digital Input	22	0	E
26	Kinco_Speed1	Long32	Write	2	Holding Register	11	0	
27	Kinco_Speed2	Long32	Write	2	Holding Register	21	0	
28	Kinco_Start1	Byte8	Write	2	Digital Coil	0	0	
29	Kinco_Start2	Byte8	Write	2	Digital Coil	3	0	
30	Kinco_Stop1	Byte8	Write	2	Digital Coil	2	0	1
31	Kinco Stop2	Bvte8	Write	2	Digital Coil	5	0	-

Observation! *Kinco_Distance1* (2) tags are 32 bits integer with sign. *Kinco_Speed1* (2) tags are 32 bits integer with no sign. Two consecutive Modbus registers each.

6)FeSCADA project

The first step in a FeSCADA project is to define the DDE communication channels and the tags. In the picture below one can see that we defined one DDE channel as channel number 1: DDE_Application = "MB" and DDE_Topic = "TAGS".

Every tag has an internal name used in FeSCADA and a DDE Name for communication with the DDE server. We kept the names the same. All the tags defined in FeMODBUS will have a correspondent tag in FeSCADA.

						11.1					-	
NO	Tag Name	DDE Name	DDE	Data Type	Update I	Value		NO	DDE Ap	DDE Topic	Con	Ē
24	Kinco_Accel1	Kinco_Accel1	1	Integer	Read/Write	200		1	MB	TAGS	Yes	
25	Kinco_Accel2	Kinco_Accel2	1	Integer	Read/Write	200		2	RSLINX	TEST		
26	Kinco_Break	Kinco_Break	1	Integer	Write	0		3	Excel	Sheet1	No	
27	Kinco_Break	Kinco_Break	1	Integer	Write	0	E	4	Excel	Silviu_Work	No	Ε
28	Kinco_Distance1	Kinco_Distance1	1	Integer	Read/Write	0.000000		5	MB2	TAGS	No	
29	Kinco_Distance2	Kinco_Distance2	1	Integer	Read/Write	0.000000		6	FESDDE	TAGS	No	
30	Kinco_Jog1	Kinco_Jog1	1	Integer	Write	0		7				
31	Kinco_Jog2	Kinco_Jog2	1	Integer	Write	0		8				H
32	Kinco_Jog_Dir1	Kinco_Jog_Dir1	1	Integer	Write	0		9				
33	Kinco_Jog_Dir2	Kinco_Jog_Dir2	1	Integer	Write	0		10				
34	Kinco_Speed1	Kinco_Speed1	1	Integer	Read/Write	0.000000		11				
35	Kinco_Speed2	Kinco_Speed2	1	Integer	Read/Write	0.000000		12				
36	Kinco_Start1	Kinco_Start1	1	Integer	Read/Write	0	-	13	8			-
Tag Kin	g Name co_Accel1		Data Type Integer	บ ิ Я (▼	pdate Type	▼ Add	1		DDE Applicat	ion		
DD	E Name					Upda	te		DDE Topic			
Kin	co Accel1							6			-	
		I	nitial Value			Delet	te					
DD	E Channel		200									
1						Searc	ch		Upd	Delete		
		Max Eng Value		Offset Value	e Max Ray	w Value						
	Scaled	200		-508	1024							
				-								

Frandos Engineering LLC

	1000 000	1302030	Anne and	1.52.52.52.52	12000251 2021 1	10000	1.24
No	Tag Name	DDE Name	DDE	Data Type	Update T	Value	-
27	Kinco_Break	Kinco_Break	1	Integer	Write	0	
28	Kinco_Distance1	Kinco_Distance1	1	Integer	Read/Write	0.000000	
29	Kinco_Distance2	Kinco_Distance2	1	Integer	Read/Write	0.000000	
30	Kinco_Jog1	Kinco_Jog1	1	Integer	Write	0	
31	Kinco_Jog2	Kinco_Jog2	1	Integer	Write	0	=
32	Kinco_Jog_Dir1	Kinco_Jog_Dir1	1	Integer	Write	0	
33	Kinco_Jog_Dir2	Kinco_Jog_Dir2	1	Integer	Write	0	
34	Kinco_Speed1	Kinco_Speed1	1	Integer	Read/Write	0.000000	
35	Kinco_Speed2	Kinco_Speed2	1	Integer	Read/Write	0.000000	
36	Kinco_Start1	Kinco_Start1	1	Integer	Read/Write	0	
37	Kinco_Start2	Kinco_Start2	1	Integer	Read/Write	0	
38	Kinco_Stop1	Kinco_Stop1	1	Integer	Write	0	
39	Kinco_Stop2	Kinco_Stop2	1	Integer	Write	0	-

Now we can build a screen (window) to let the user to input the data and trigger the commands. In the picture below one can see a snapshot of this screen. We are controlling 2 stepper motors. We used sliders to set the moving distances and speeds between min and max limits. We use numerical displays for acceleration value. Also we have buttons to start, stop and jog the stepper motors. We can select forward or reverse rotation.



The picture below is showing a snapshot from the KincoBuilder programming software for K5 PLC. The image is showing instructions to control a stepper motor. PREL is an instruction for relative positioning. PJOG is for jogging the stepper motor with constant speed. PSTOP is for stopping all moves of a stepper motor.





7) Conclusions

The application has shown an example of using FeSCADA and FeMODBUS to communicate with Kinco K5 PLC. In this application, with FeSCADA software, we controlled two stepper motors. We can start, stop relative positioning moves for which we can set the distance and the speed. We can jog the stepper motors forward and reverse with adjustable speeds.

The communication used is serial RS485. The protocol is Modbus RTU.