FeSCADA & DL06 PLC - AutomationDirect

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 DL06 PLC - AutomationDirect.

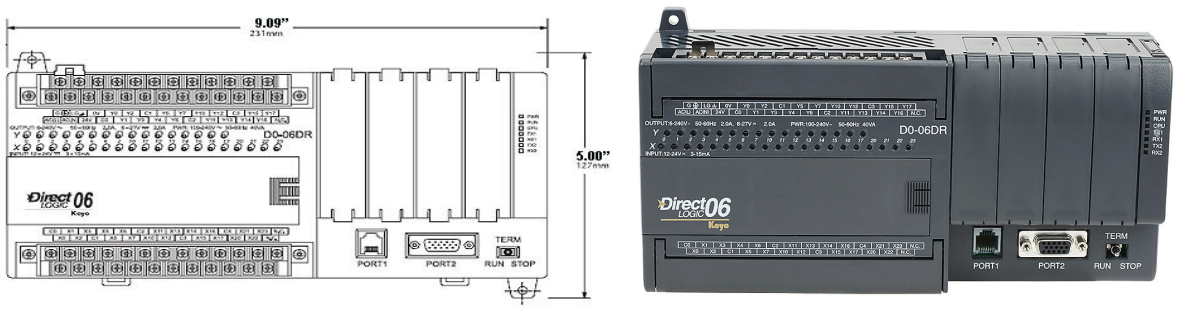
1. Description
2. Hardware
3. RS485 serial communication
4. DL06 Modbus registers
5. FeMODBUS communication setup
6. FeSCADA project
7. Binary Coded Decimal (BCD) format
8. Conclusions

1) Description

In the following pages an application is developed for monitoring inputs, outputs, control relays, timers and counters of a DL06 PLC - AutomationDirect. The outputs can be switched on/off. The preset numbers for timers and counters can be changed from FeSCADA.

2) Hardware

The hardware is composed of one AutomationDirect DirectLOGIC06 PLC. The DL06 line of micro PLCs offers a solution for small to medium size I/O applications (36 built-in I/O points and up to 100 I/O points total). The DL06 micro PLC line has nine base units to choose from, each has 20 built-in inputs and 16 built-in outputs. The DL06 is a great fit for high voltage, 220-240 VAC applications or remote 12 VDC applications, and everything in between.



More information at:
https://www.automationdirect.com/adc/overview/catalog/programmable_controllers/directlogic_series_plcs_(micro_to_small,_brick_-a-_modular)/directlogic_06_(expandable_micro_brick_plc)/plc_units

3) RS485 serial communication

The DL06 PLC has two serial communication ports. Port1 is used primarily for programming and is RS232. Port2 is used for other serial connections: other PLCs, HMI, SCADA, etc., and can be used as an RS232 or RS485.

In our project we are using the adapter ZL-CMA15A to connect Port2 to an RS485 serial network. The PLC is programmed to work on Port2 as a Modbus RTU server with the address = 3 (three). The Port2 settings are: 38400 baud, no parity, 8 data and 1 stop bits.

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
D0-06DR	$382.00	1	$382.00	DL06 PLC AutomationDirect
ZL-CMA15A	$46.50	1	$46.50	RS485 Adapter to 15pin D-sub
USOPTL4-LS	$143.00	1	$143.00	USB to RS485 Adapter
	TOTAL =		**$571.50**	

4) DL06 Modbus registers

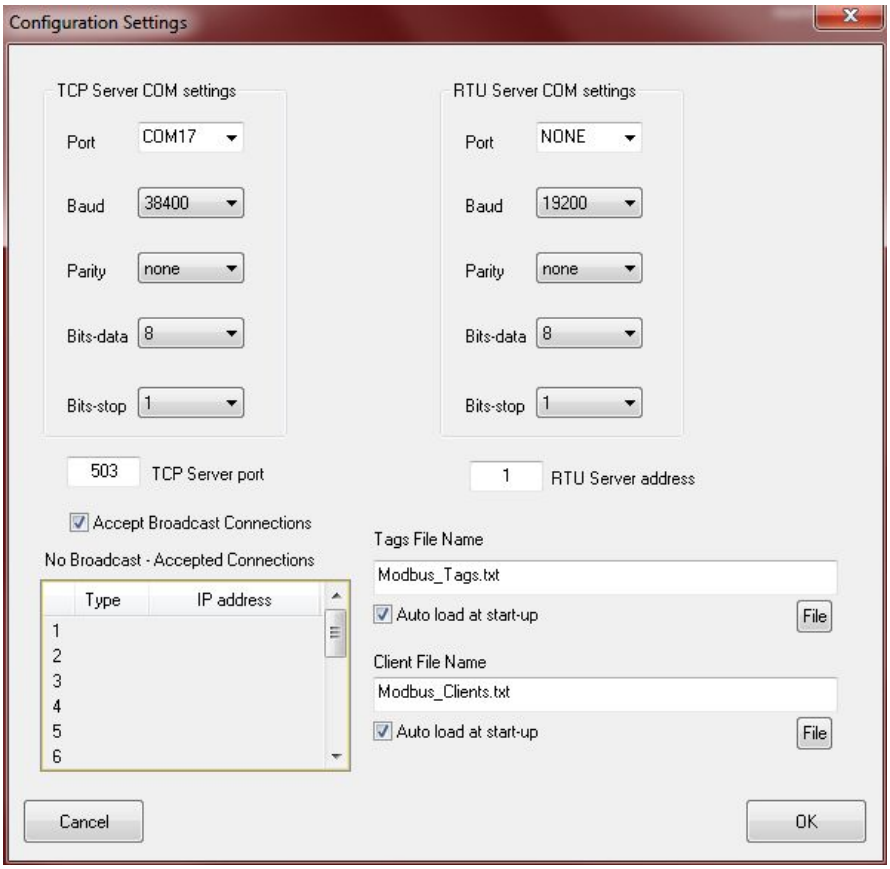
The following table is showing the Modbus register numbers for various data areas in the DL06 PLC: inputs, outputs, control relays, timers, counters, free memory data.

X_n , Y_n , C_n , T_n , CT_n , V_n , are symbols and 'n' is a number written in octal base. In the table below, '$)_{10}$' means that the number 'n' in octal base has to be converted to decimal base.

Data type	**Symbol** (number in octal base)	**Modbus register number** (in decimal base)	**Modbus register type**
Inputs	X0, X1, ... X7 X10, X11, ... X17 ...	2048, 2049, ... 2055 2056, 2057, ... 2063 ...	Digital inputs $2048+(X_n)_{10}$
Outputs	Y0, Y1, ... Y7 Y10, Y11, ... Y17 ...	2048, 2049, ... 2055 2056, 2057, ... 2063 ...	Coils $2048+(Y_n)_{10}$
Control relays	C0, C1, ... C7 C10, C11, ... C17 ...	3072, 3073, ... 3079 3080, 3081, ... 3087 ...	Coils $3072+(C_n)_{10}$
Timer contacts	T0, T1, ... T7 T10, T11, ... T17 ...	6144, 6145, ... 6151 6152, 6153, ... 6159 ...	Coils $6144+(T_n)_{10}$
Counters contacts	CT0, CT1, ... CT7 CT10, CT11, ... CT17 ...	6400, 6401, ... 6407 6408, 6409, ... 6415 ...	Coils $6400+(CT_n)_{10}$
V-memory Timers	V0, V1, ... V7 ...	0, 1, ... 7 ...	Input registers $0+(T_n)_{10}$
V-memory Counters	V1000, V1001, ... V1007 ...	512, 513, ... 519 ...	Input registers $512+(CT_n)_{10}$
V-memory	V2000, V2001, ... V2007 ...	1024, 1025, ... 1031 ...	Holding registers $0+(V_n)_{10}$

5) FeMODBUS communication setup

FeMODBUS software is used to connect to DL06 PLC – AutomationDirect.

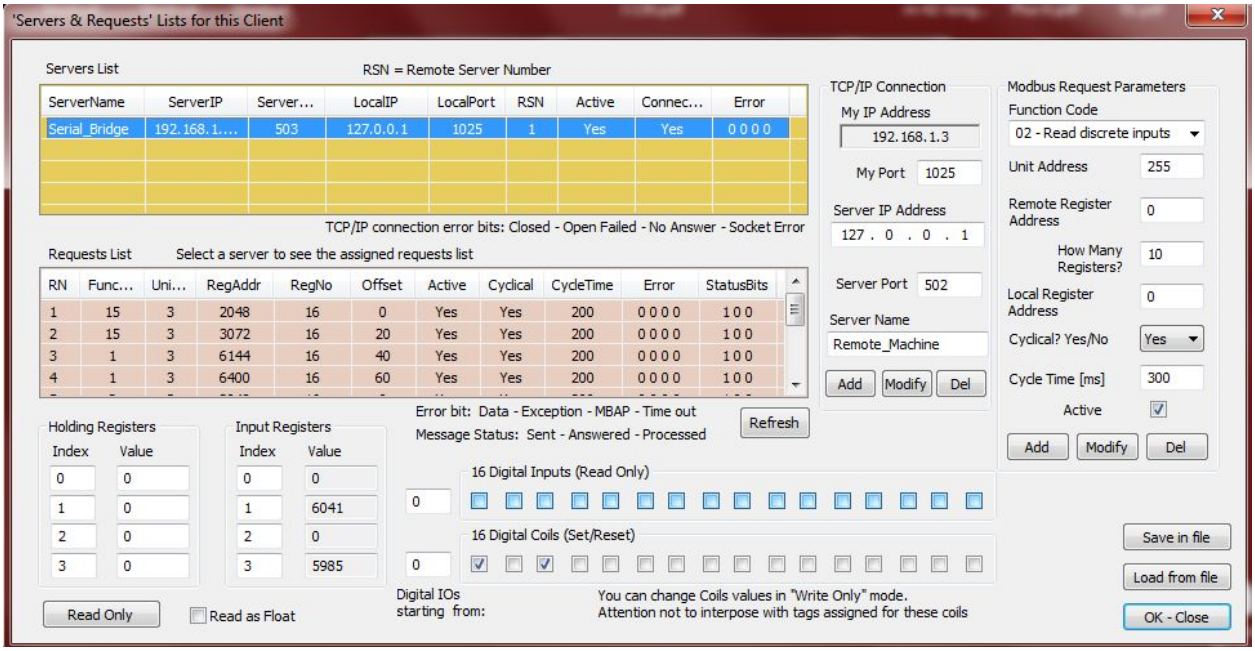


In the picture on the right the serial connection settings are presented.

COM17 was assigned for our USB to RS485 adapter by Windows, when we plugged in the adapter. The other settings are matching the settings for COM Port2 on the DL06 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 TCP server, and we send requests to the address 3. Because the address 3 is less than 255, the server will send them onto the serial COM port assigned for TCP server, and will send the answers back.



Requests List Select a server to see the assigned requests list

RN	Func...	Uni...	RegAddr	RegNo	Offset	Active	Cyclical	CycleTime	Error	StatusBits
1	15	3	2048	16	0	Yes	Yes	200	0 0 0 0	1 0 0
2	15	3	3072	16	20	Yes	Yes	200	0 0 0 0	1 0 0
3	1	3	6144	16	40	Yes	Yes	200	0 0 0 0	1 0 0
4	1	3	6400	16	60	Yes	Yes	200	0 0 0 0	1 0 0

Requests List Select a server to see the assigned requests list

RN	Func...	Uni...	RegAddr	RegNo	Offset	Active	Cyclical	CycleTime	Error	StatusBits
5	2	3	2048	16	0	Yes	Yes	200	0 0 0 0	1 0 0
6	4	3	0	16	0	Yes	Yes	200	0 0 0 0	1 0 0
7	4	3	512	16	20	Yes	Yes	200	0 0 0 0	1 0 0
8	16	3	640	16	0	Yes	Yes	200	0 0 0 0	1 0 0

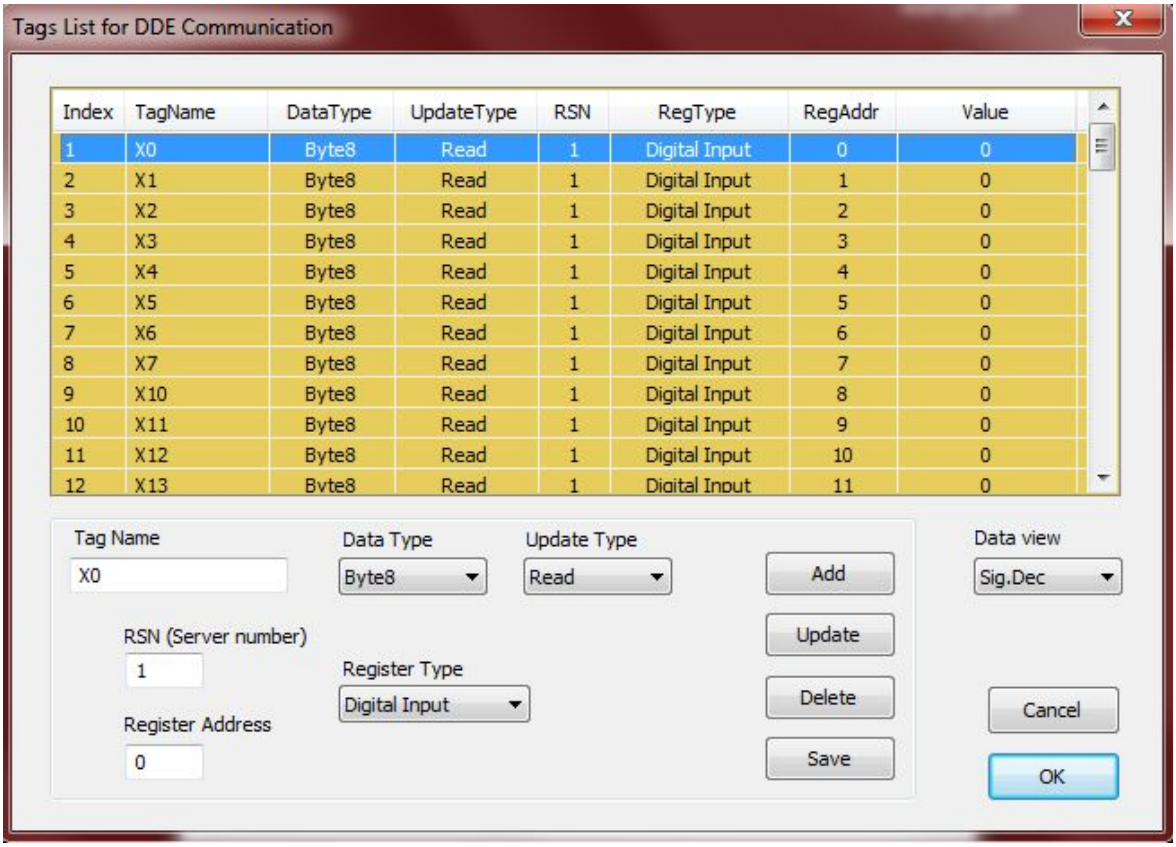
Requests List Select a server to see the assigned requests list

RN	Func...	Uni...	RegAddr	RegNo	Offset	Active	Cyclical	CycleTime	Error	StatusBits
6	4	3	0	16	0	Yes	Yes	200	0 0 0 0	1 0 0
7	4	3	512	16	20	Yes	Yes	200	0 0 0 0	1 0 0
8	16	3	640	16	0	Yes	Yes	200	0 0 0 0	1 0 0
9	16	3	1024	16	20	Yes	Yes	200	0 0 0 0	1 0 0

The pictures above are showing that FeMODBUS was setup to connect at the address 192.168.1.3, which is our own PC address, and to cyclically (every 200ms) send some of the possible requests:

- Function 1 – read coils (digital outputs)
- Function 2 – read digital inputs
- Function 3 – read holding registers
- Function 4 – read input registers
- Function 15 – write coils (digital outputs)
- Function 16 – write holding registers

After setting up all the requests we defined tags which we linked 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 1 (RSN 1). See picture below.



Index	TagName	DataType	UpdateType	RSN	RegType	RegAddr	Value
13	X14	Byte8	Read	1	Digital Input	12	0
14	X15	Byte8	Read	1	Digital Input	13	0
15	X16	Byte8	Read	1	Digital Input	14	0
16	X17	Byte8	Read	1	Digital Input	15	0
17	Y0	Byte8	Write	1	Digital Coil	0	1
18	Y1	Byte8	Write	1	Digital Coil	1	0
19	Y2	Byte8	Write	1	Digital Coil	2	1
20	Y3	Byte8	Write	1	Digital Coil	3	0
21	Y4	Byte8	Write	1	Digital Coil	4	0
22	Y5	Byte8	Write	1	Digital Coil	5	0
23	Y6	Byte8	Write	1	Digital Coil	6	0
24	Y7	Byte8	Write	1	Digital Coil	7	0

Index	TagName	DataType	UpdateType	RSN	RegType	RegAddr	Value
25	Y10	Byte8	Write	1	Digital Coil	8	0
26	Y11	Byte8	Write	1	Digital Coil	9	0
27	Y12	Byte8	Write	1	Digital Coil	10	0
28	Y13	Byte8	Write	1	Digital Coil	11	0
29	Y14	Byte8	Write	1	Digital Coil	12	0
30	Y15	Byte8	Write	1	Digital Coil	13	0
31	Y16	Byte8	Write	1	Digital Coil	14	0
32	Y17	Byte8	Write	1	Digital Coil	15	0
33	C0	Byte8	Write	1	Digital Coil	20	0
34	C1	Byte8	Write	1	Digital Coil	21	1
35	C2	Byte8	Write	1	Digital Coil	22	0
36	C3	Byte8	Write	1	Digital Coil	23	1

Index	TagName	DataType	UpdateType	RSN	RegType	RegAddr	Value
37	C4	Byte8	Write	1	Digital Coil	24	1
38	C5	Byte8	Write	1	Digital Coil	25	0
39	C6	Byte8	Write	1	Digital Coil	26	0
40	C7	Byte8	Write	1	Digital Coil	27	0
41	T0	Byte8	Read	1	Digital Coil	40	0
42	T1	Byte8	Read	1	Digital Coil	41	1
43	T2	Byte8	Read	1	Digital Coil	42	0
44	T3	Byte8	Read	1	Digital Coil	43	1
45	T4	Byte8	Read	1	Digital Coil	44	1
46	T5	Byte8	Read	1	Digital Coil	45	0
47	T6	Byte8	Read	1	Digital Coil	46	0
48	T7	Byte8	Read	1	Digital Coil	47	0

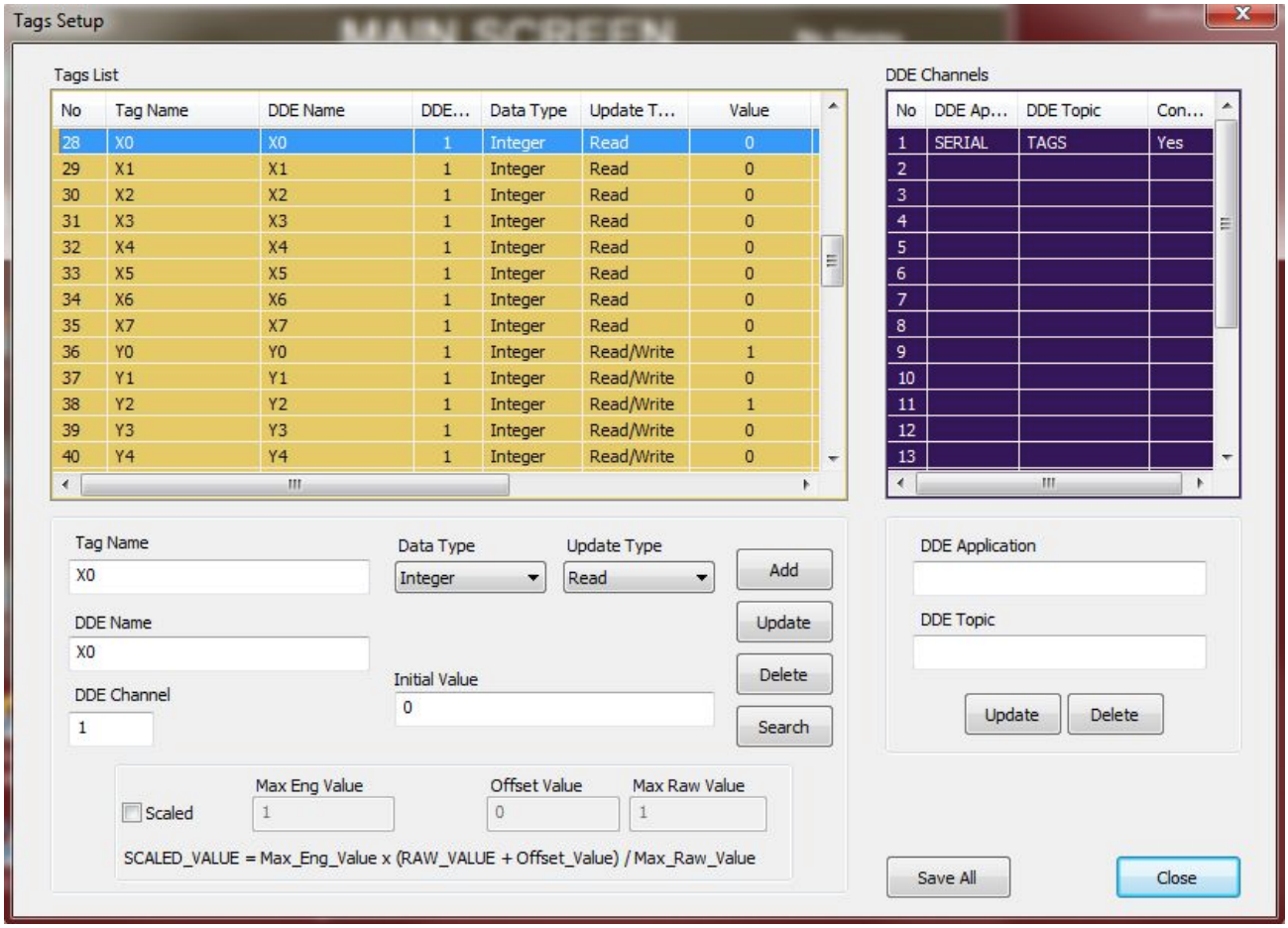
Index	TagName	DataType	UpdateType	RSN	RegType	RegAddr	Value
49	K0	Byte8	Read	1	Digital Coil	60	1
50	K1	Byte8	Read	1	Digital Coil	61	1
51	K2	Byte8	Read	1	Digital Coil	62	0
52	K3	Byte8	Read	1	Digital Coil	63	0
53	K4	Byte8	Read	1	Digital Coil	64	0
54	K5	Byte8	Read	1	Digital Coil	65	0
55	K6	Byte8	Read	1	Digital Coil	66	0
56	K7	Byte8	Read	1	Digital Coil	67	0
57	V2000	Word16	Write	1	Holding Register	20	200
58	V2001	Word16	Write	1	Holding Register	21	100
59	V2002	Word16	Write	1	Holding Register	22	50
60	V2003	Word16	Write	1	Holding Register	23	0

Index	TagName	DataType	UpdateType	RSN	RegType	RegAddr	Value
61	V2004	Word16	Write	1	Holding Register	24	0
62	V2005	Word16	Write	1	Holding Register	25	0
63	V2006	Word16	Write	1	Holding Register	26	0
64	V2007	Word16	Write	1	Holding Register	27	0
65	CT0	Byte8	Read	1	Digital Coil	60	1
66	CT1	Byte8	Read	1	Digital Coil	61	1
67	CT2	Byte8	Read	1	Digital Coil	62	0
68	CT3	Byte8	Read	1	Digital Coil	63	0
69	CT4	Byte8	Read	1	Digital Coil	64	0
70	CT5	Byte8	Read	1	Digital Coil	65	0
71	CT6	Byte8	Read	1	Digital Coil	66	0
72	CT7	Byte8	Read	1	Digital Coil	67	0

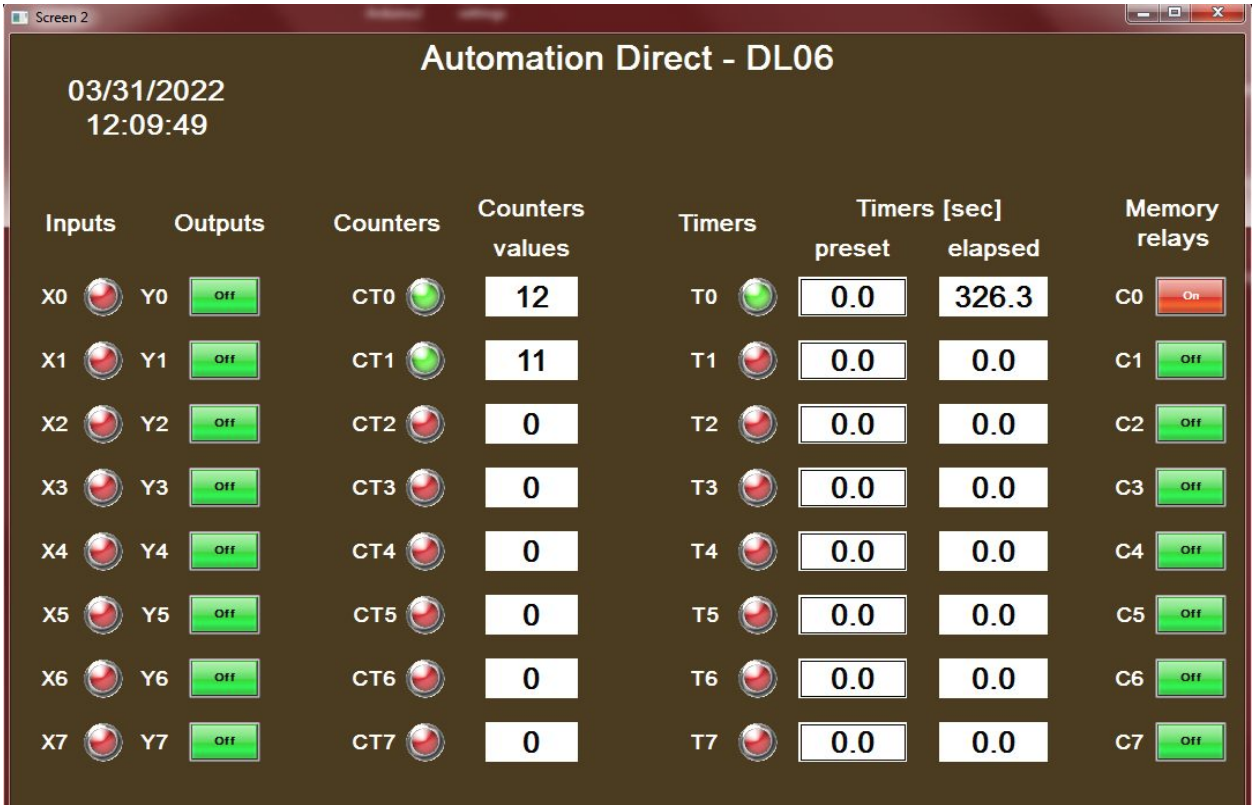
6) FeSCADA project

The first step in a FeSCADA project is to define the DDE communication channels and the tags. In the picture below we defined the DDE channel number 1(one) as: DDE_Application = “SERIAL” and DDE_Topic = “TAGS”.

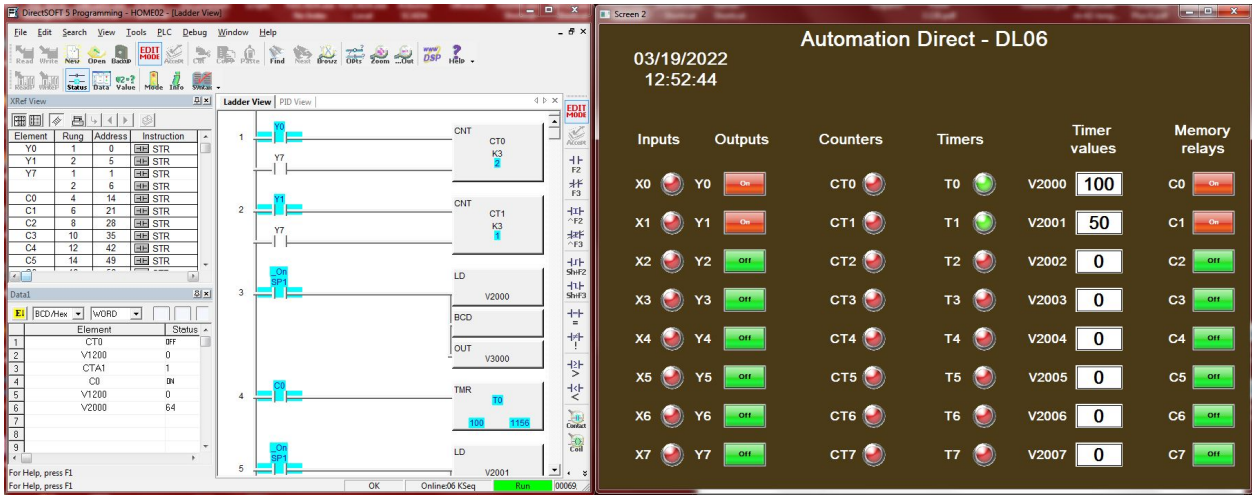
Every tag has an internal name used in FeSCADA and a DDE Name for communication with “SERIAL” DDE server. We kept the names the same. All the tags defined in FeMODBUS will have a correspondent tag in FeSCADA.



Now we can build a screen (window) to show the data, and to put some indicators and buttons. In the picture below one can see a snapshot of this screen. We display 8 inputs, outputs, counters, timers and memory relays. For the timers we show the status and the counting values. For the timers we show the status and the elapsed time. The preset time is a value that the user can change.



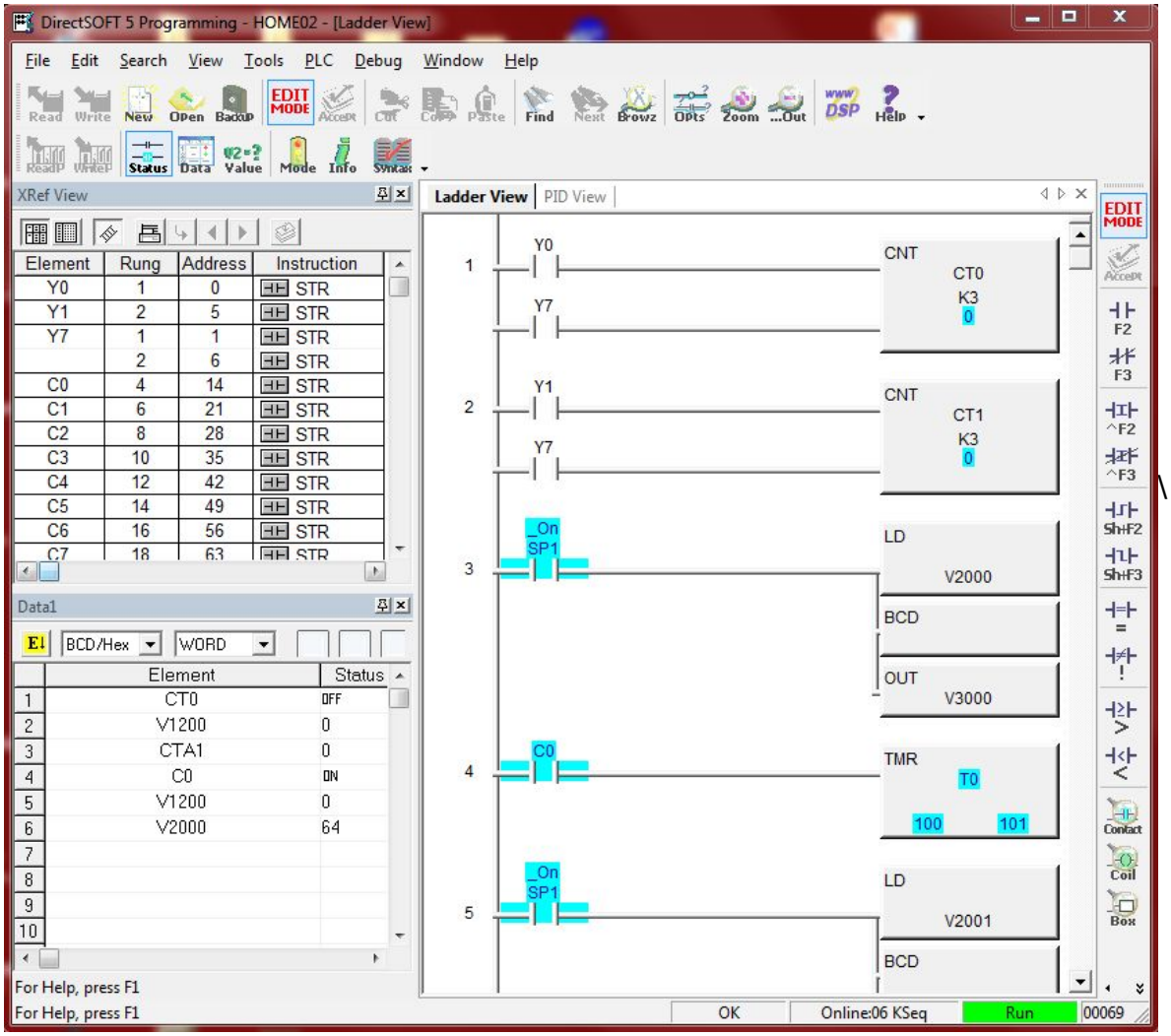
In the picture below the FeSCADA screen is shown in parallel with a snapshot from the DirectSOFT 5 programming software for DL06 PLC.



7) Binary Coded Decimal (BDC) format

In AutomationDirect PLCs the values used for counters and timers are in BCD format. In FeSCADA the integers are represented only in decimal base.

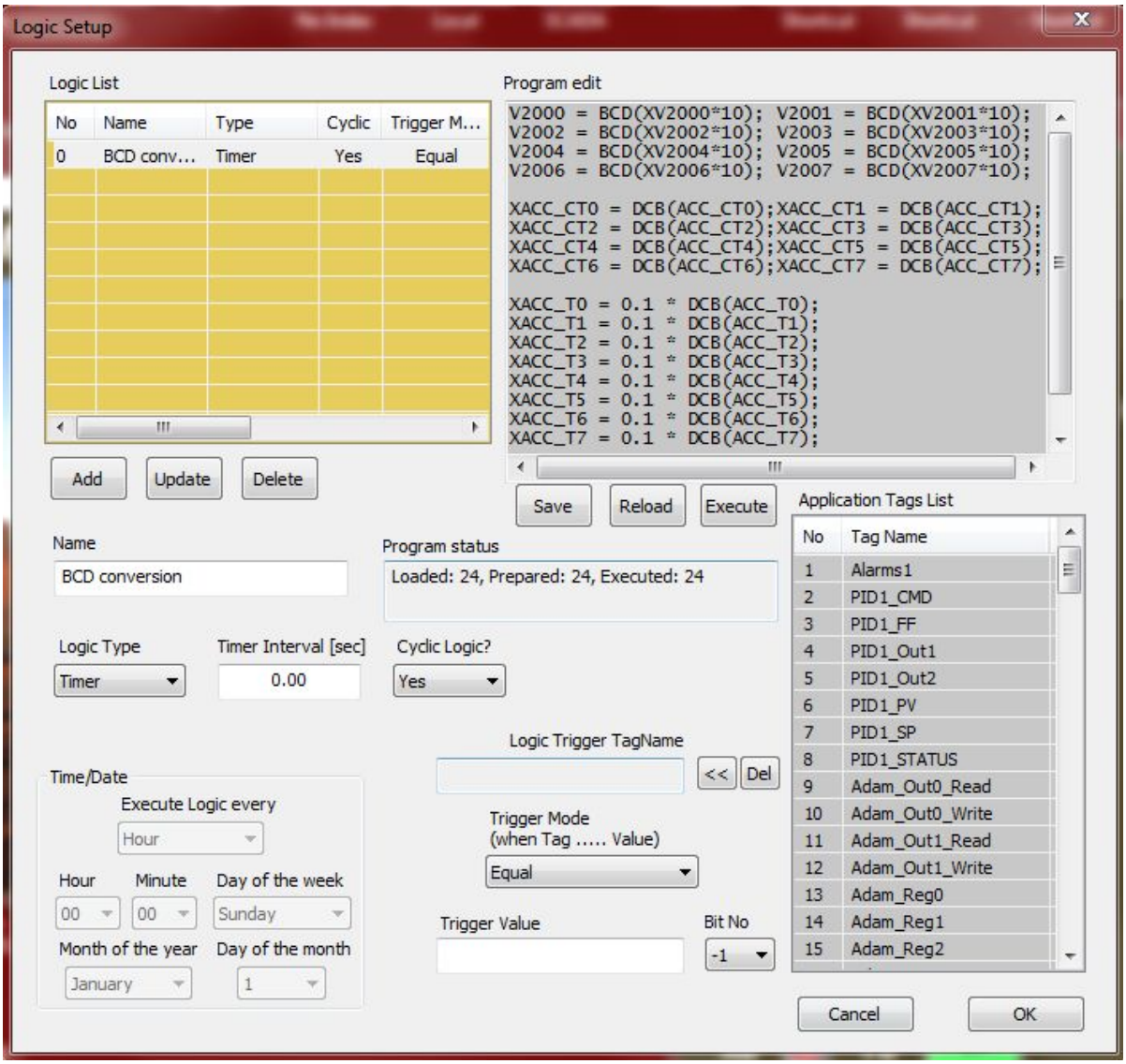
There are 2 possibilities to deal with this format difference. The first is to do more programming in the PLC, to convert any register changed from FeSCADA to a BCD format before loading that number as a parameter for a timer or a counter. And to convert to DCB any value from a timer or a counter that has to be read with FeSCADA. In the picture below it is a snapshot from a DirectSOFT 5 program.



The second possibility is to use a logic program in FeSCADA and to use the functions BCD(X) and DCB(Y) that were created in FeSCADA especially for AutomationDirect PLC users. In this later case, some more memory tags have to be defined.

In the program from the picture below, the tag *XV2000* is a memory tag that the user can change with a numerical display/entry control, as a preset value, and the tag *V2000* is the corresponding BCD format. The tag *V2000* is sent to the PLC.

The tag *XACC_T0* is a memory tag that is showing the accumulator of timer T0 after its conversion from BCD format to decimal integer, with the function DCB(*ACC_T0*). The tag *ACC_T0* is the communication tag, with a value received from the PLC.



8) Conclusions

The application has shown an example of using FeSCADA and FeMODBUS to visualize and/or to change: inputs, outputs, timers, counters, data registers from a DL06 PLC – AutomationDirect. The communication used is serial RS485. The protocol is Modbus RTU.