EPLC96XX Operator Panel and PLC Controller Specification & Configuration Files Description for CoDeSys V2.3

CoDeSys® V2.3

Phone : +90 224 261 1900
Faks : +90 224 261 1912
Url : www.emkoelektronik.com.tr
E-mail : Emko@emkoelektronik.com.tr
exposales@emkoelektronik.com.tr

CONTENT

1 GENERAL DESCRIPTIONS FOR PLC CONFIGURATION 3
1.1 Overview 3
1.2 Working in the plc configuration 3
1.3 Selecting of elements 3
1.4 Insert elements, 'Insert' 'Insert element', 'Insert' 'Append subelement’ 3
1.5 Replacing/switching Elements, 'Extras' 'Replace element'’ 4
1.6 General settings in the PLC configuration 4
1.7 Return to standard configuration, 'Extras' 'Standard configuration’ 4

2 INITIAL SETTINGS FOR EMKO EPLC96XX VIA CONFIGURATION FILE 5

3 PLC CONFIGURATION CONTENT 5
3.1 PLC settings parameters 6
3.1.2 LCD setting parameters 6
3.1.3 Input card type selection 6
3.1.4 High Speed Input (HSC) setting parameters 7
3.1.5 Digital input setting 7
3.1.6 Analog input setting 7
3.1.7 Output card type selection 8
3.1.8 PWM / PTO selection 8
3.1.9 PWM output setting parameters 8
3.1.10 PTO output setting parameters 9
3.1.11 Analog output setting 9
3.1.12 Digital output setting 9
3.1.13 Communication card type selection 10
3.1.14 RS232 communication settings 10
3.1.15 RS485 communication settings 10
3.1.16 Ethernet communication Settings 11
3.1.17 USB communication Settings 11

4 EPLC96XX TARGET SPECIFICATIONS 12
4.1 EPLC96XX MEMORY LAYOUT 12
4.1.1 Program Memory 13
4.1.2 Input Memory 13
4.1.3 Output Memory 14
4.1.4 Marker Memory 15
MODBUS 20
MOTORIZED VALVE CONTROLLER MODULE 22
HIGH SPEED COUNTER (HSC) 23
PWM/PTO OUTPUT 27
EEPROM READ/WRITE FUNCTION 29
ETHERNET COMMUNICATION SETTINGS 31

1 General Descriptions For PLC Configuration

1.1 Overview

The PLC Configuration is found as an object in the register card **Resources** in the Object Organizer. With the PLC Configuration editor, you must describe the hardware the opened project is established for. For the program implementation, the number and position of the inputs and outputs is especially important. With this description, **CoDeSys** verifies whether the IEC addresses used in the Program also actually exist in the hardware.

The PLC Configuration is displayed in the editor in tree structure and can be edited using menu commands and dialogs. The configuration contains input and/or output elements and also management elements which themselves also have sub elements (for example, a CAN-bus or a digital input card with 8 inputs).

For inputs and outputs, symbolic names can be assigned. The IEC-address where this input or output can be accessed is then located behind the symbolic name.

1.2 Working in the PLC Configuration

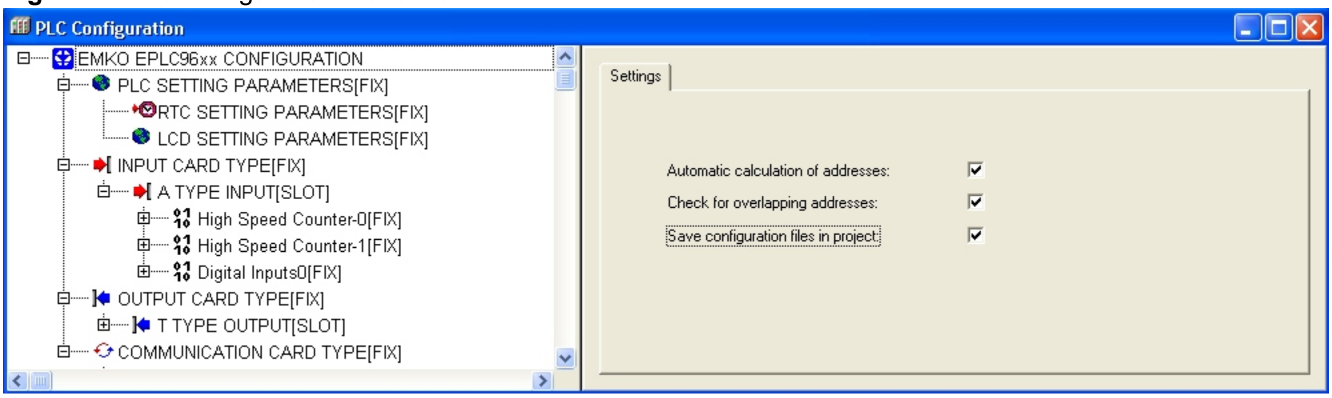
The configuration editor is divided up in two parts. In the left window the configuration tree is displayed.
In the right window the currently available configuration dialogs are shown on one or several tabs.

The right part of the window is per default visible, but can get faded out via the menu item 'Extras' 'Properties'.

On top of the configuration tree there is the entry of the "root" module with a name, which has been defined in the configuration file.

Below you find hierarchically indented the other elements of the configuration: Modules of different types (I/O, Communication), channels or bit channels.

Figure 1: PLC configuration menu view



1.3 Selecting of elements

For selecting elements, click the mouse on the corresponding element, or, using the arrow keys, move the dotted rectangle onto the desired element. Elements that begin with a plus sign are organization elements and contain sub elements. To open an element, select the element and double-click the plus sign or press <Enter>. You can close opened elements (minus sign in front of the element) the same way.

1.4 Insert elements, 'Insert' 'Insert element', 'Insert' 'Append subelement'

Depending on the definitions in the configuration file(s) and on the available device files, which have been read when the project was opened, a basic composition of elements is automatically positioned in the configuration tree. If one of those elements is selected, further elements may be added if this is allowed by the definitions in the configuration file and if the needed device files are available:

'Insert' 'Insert element': An element can be selected and inserted before the element which is currently marked in the configuration tree.

'Insert' 'Append subelement': An element can be selected and inserted as subelement of the element which is currently marked in the configuration tree. It will be inserted at the last position.

The most important commands are found in the context menu (right mouse button or <Ctrl>+<F10>).

1.5 Replacing/switching Elements, 'Extras' 'Replace element''

Depending on the definition in the configuration file, the currently selected element may be get replaced by an other one. The same way it may be possible to switch channels, which are set up in a way that they can be used as input or as output elements. Use the command 'Extras' 'Replace element'

1.6 General Settings in the PLC Configuration

Select the entry 'PLC configuration' ('root' module) at top of the configuration tree. Thereupon the dialog 'Settings' is shown in the right part of the window. The following options can be activated:

Calculate addresses: Each newly inserted module automatically is allocated with an address, which results from the address of the module inserted before plus the size of this address. If a module is removed from the configuration, the addresses of the following modules are adjusted automatically. When the command 'Extras' 'Compute addresses' is executed, all addresses starting at the selected node (module) will be recalculated.

Check for overlapping addresses: At compilation the project will be checked for overlapping addresses and a corresponding message will be displayed.

Save configuration files in project: The information which is contained in the configuration file(s) *.cfg and the device description files, which underlie the current PLC Configuration, will be saved in the project. Thus (if it is not defined by the configuration file, that always the standard configuration should be reloaded !), the configuration as set up by the user will be kept in the project, even if configuration files are not found when the project is re-opened. Keep in mind that in such a case the complete project specific configuration will get lost, if the here described option is not activated !
By saving the configuration information with the project these also will be kept at a target change. But regard in this case, that the new target might bring own configuration files which then will be regarded additionally.

1.7 Return to standard configuration, 'Extras' 'Standard configuration'

The command 'Extras' 'Standardconfiguration' can be used to restore the original PLC configuration, which is defined by the configuration file *.cfg and saved in the project.

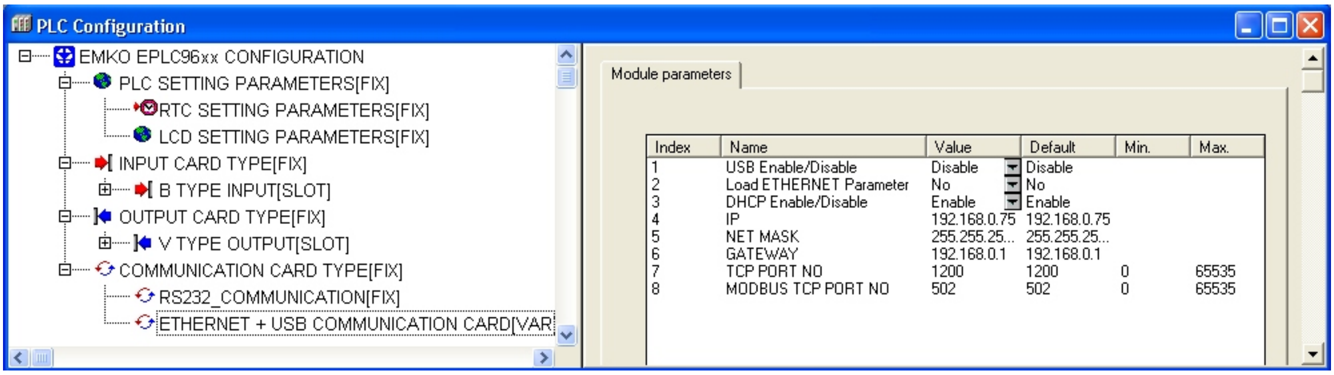
2 Initial Settings For EMKO EPLC96XX Via Configuration File

When the user load the PLC program into the PLC unit, the configuration file settings are also loading into the PLC unit with same way, so the user can adjust the PLC while loading the PLC program if you need.

3 PLC Configuration Content

In the left window of the configuration tree, there are 4 main title under the EMKO EPLC96XX CONFIGURATION root module.

Figure 2: PLC configuration content



* **PLC SETTINGS PARAMETERS:** There are 2 subtitles in this area

- * **RTC SETTINGS PARAMETERS:** User can adjust the Real Time(RTC) via in this section's parameters.

- * **LCD SETTINGS PARAMETERS:** User can adjust the LCD Backligth is ON/OFF and LCD Contrast value via in this section's parameters.

* **INPUT CARD TYPE :** There is 1 subtitle in this area, the user can select their Input card types in this section and adjust all setting for Digital Input, Analog Input, High Speed Counter Input.

* **OUTPUT CARD TYPE :** There is 1 subtitle in this area, the user can select their Output card types in this section and adjust all setting for Digital output, Analog output, PWM / PTO output.

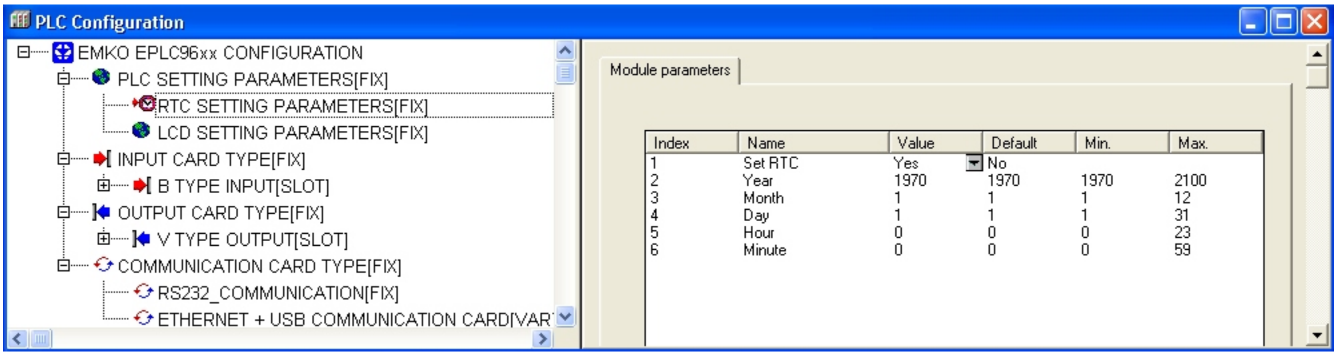
* **COMMUNICATION CARD TYPE :** There are 2 subtitles in this area, one of them is fix and the other is can be add if the optional communication card is available, the user can select their optional communication card type in this section and adjust all setting for RS232, RS485, Ethernet and USB communications.

3.1 PLC settings parameters: There are 2 subtitles in this area

3.1.1 RTC Setting Parameters:

User can adjust the Real Time (RTC) via in this section's parameters. If the **Set RTC** parameter is selected as a **YES**, initial RTC values (Year, Month, Day, Hour, Minute) are also loaded to RTC module while the PLC Program is downloading. If the **Set RTC** parameter is selected as a **NO**, RTC values is not loaded to RTC module while the PLC Program is downloading. RTC is saving their old values.

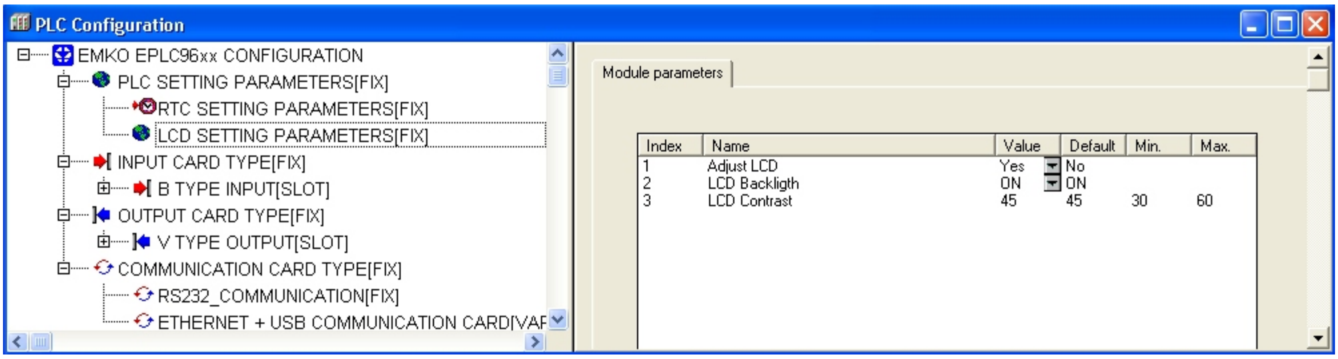
Figure 3: RTC settings



3.1.2 LCD Setting Parameters:

User can adjust LCD settings via in this section's parameters. If the **Adjust LCD** parameter is selected as a **YES**, LCD backligth status and LCD Contrast value are loaded to LCD module while the PLC Program is downloading. If the **Adjust LCD** parameter is selected as a **NO**, New LCD settings is not loaded to LCD module while the PLC Program is downloading.

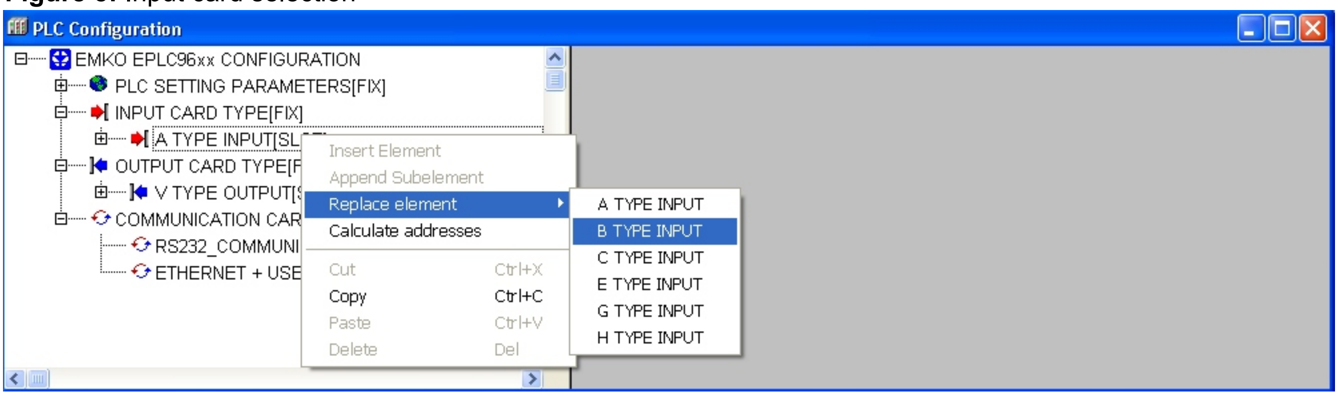
Figure 4: LCD settings



3.1.3 Input Card Type Selection:

Select the **INPUT CARD TYPE** elements according to the section "**1.3 Selecting of elements**". A TYPE INPUT is showing as a submodule after opened this element. When rigth clicked the mouse on A TYPE INPUT element, the properties section of this element is opened and pressing on the replace element section, available input card types are listed. User can select the input card type which is pluged on PLC.

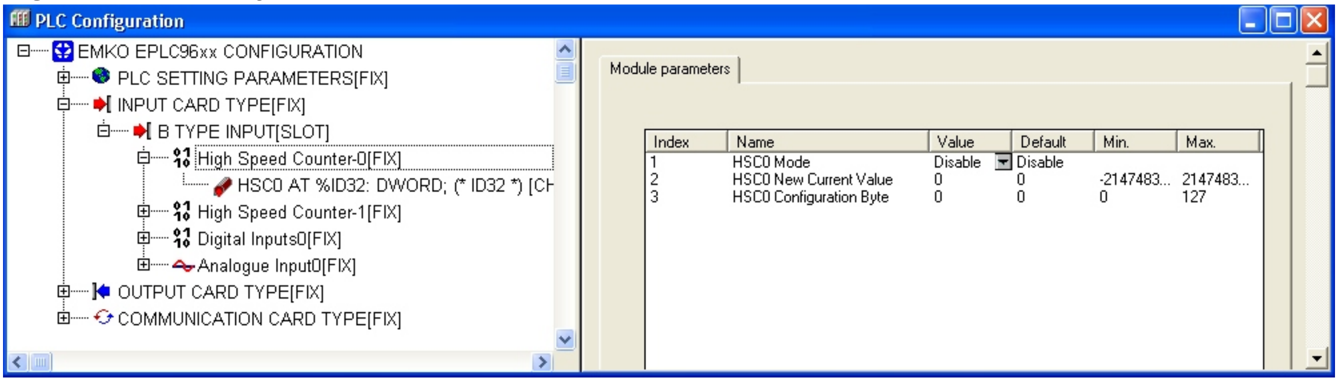
Figure 5: Input card selection



3.1.4 High Speed Input (HSC) Setting Parameters:

User can adjust the HSC settings under the input card type section . **HSC Mode** and **HSC Configuration Byte** parameters descriptions are given in section 4.1.4

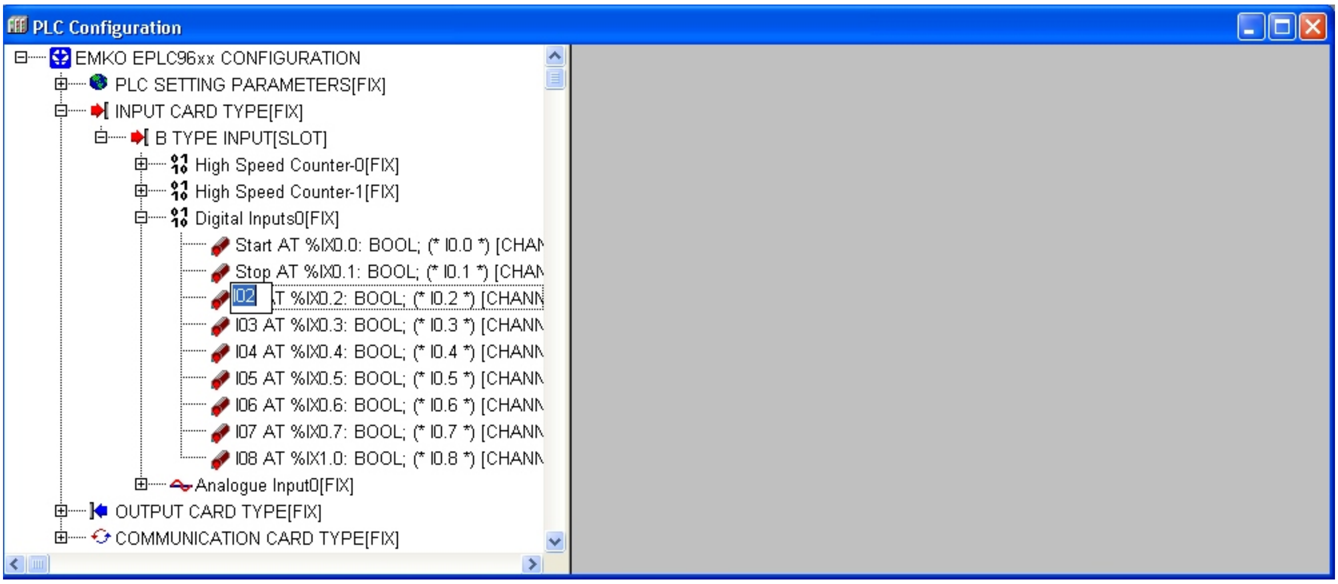
Figure 6: HSC settings



3.1.5 Digital Input Setting:

User can see the digital input adresses and when pressing the name section like a showing below figure, can change the input name for using PLC program.

Figure 7: Digital input settings



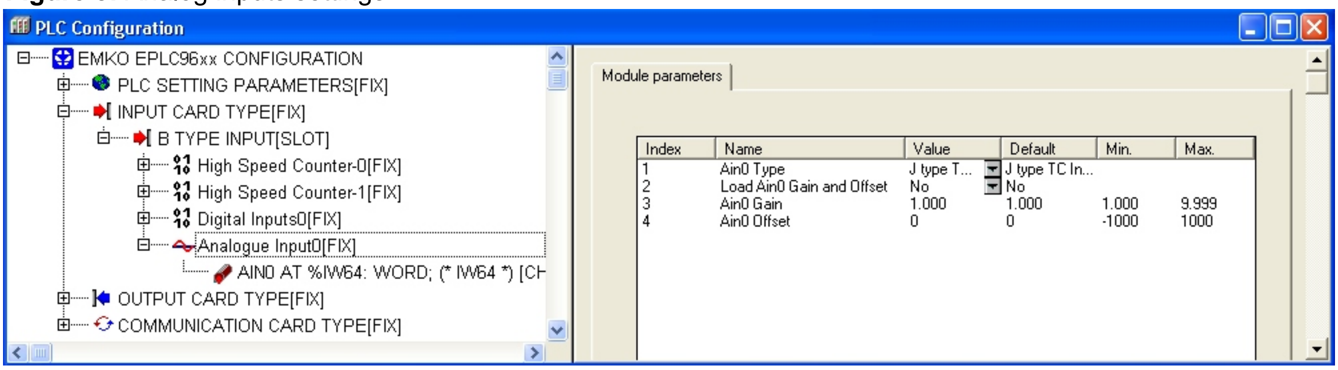
3.1.6 Analog Input Setting:

User can select the Analogue Input type in **Ain(x) Type** parameter in this section.

If the **Load Ain(x) Gain and Offset** parameter value is **YES**, then **Ain(x) Gain** and **Ain(x) Offset** parameter value is updated while the PLC program loading.

If the **Load Ain(x) Gain and Offset** parameter value is **NO**, then **Ain(x) Gain** and **Ain(x) Offset** parameter value is saving their old value.

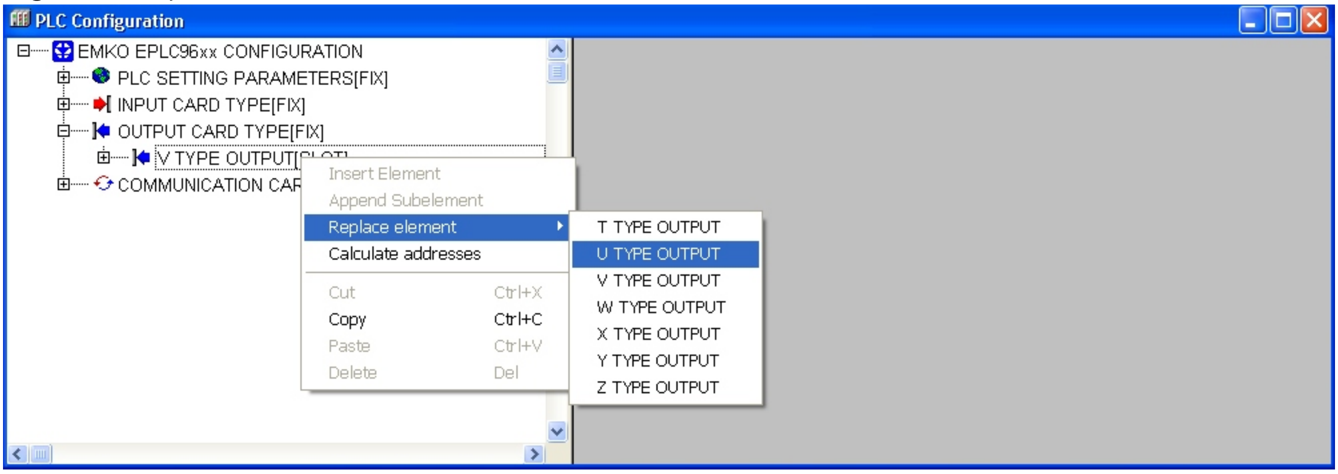
Figure 8: Analog inputs settings



3.1.7 Output Card Type Selection:

Select the **OUTPUT CARD TYPE** elements according to the section “**1.3 Selecting of elements**”. T TYPE OUTPUT is showing as a submodule after opened this element. When rigth clicked the mouse on T TYPE OUTPUT element, the properties section of this element is opened and pressing on the replace element section, available output card types are listed. User can select the output card type which is plugged on PLC.

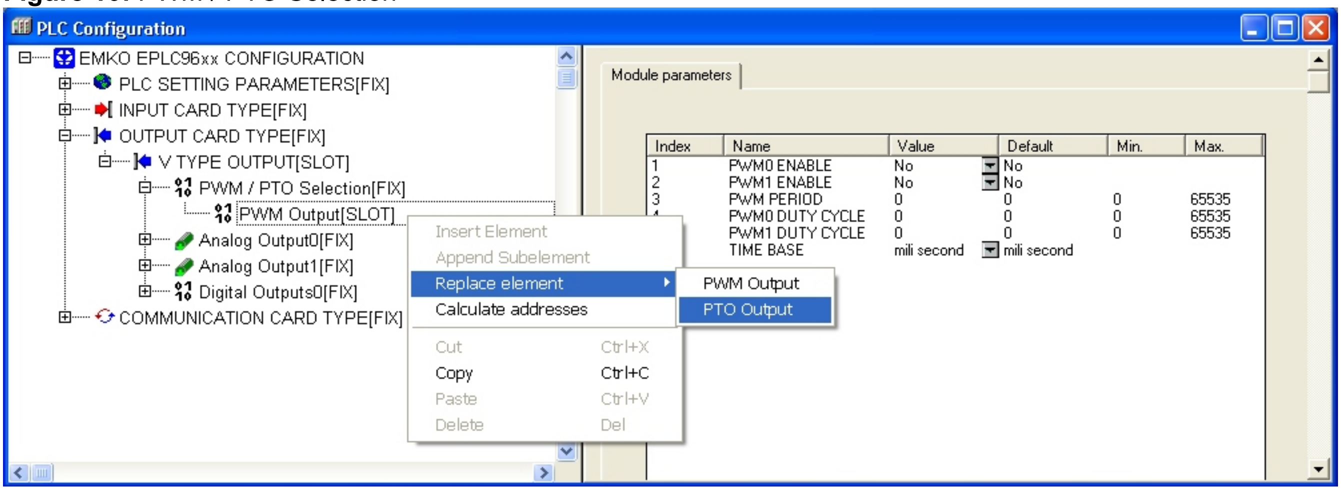
Figure 9: Output card selection



3.1.8 PWM / PTO Selection:

Select the **PWM / PTO Selection** elements according to the section “**1.3 Selecting of elements**”. PWM Output is showing as a submodule after opened this element. When rigth clicked the mouse on PWM Output element, the properties section of this element is opened and pressing on the replace element section, output types are listed. User can select the output type for PLC.

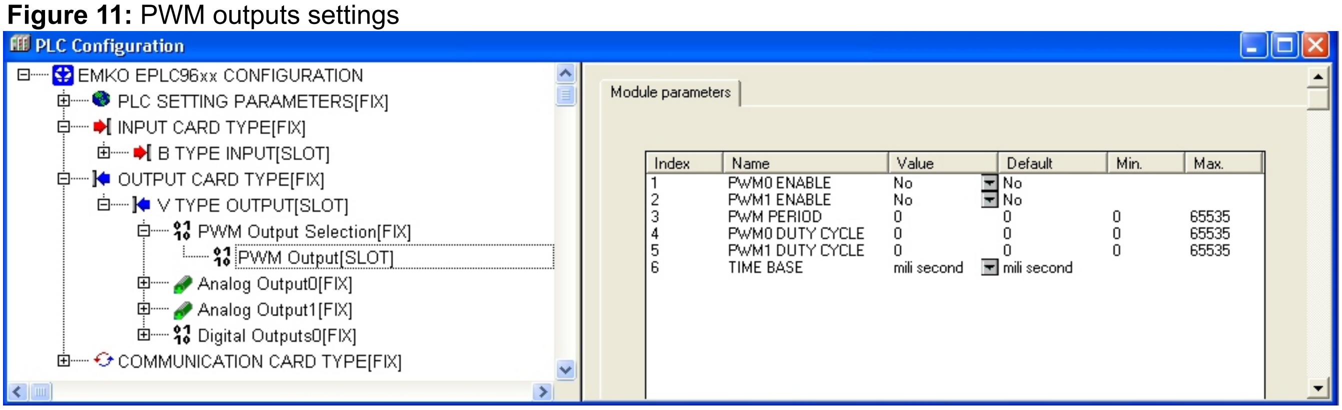
Figure 10: PWM / PTO Selection



3.1.9 PWM Ouput Setting Parameters:

User can adjust the PWM output settings under the output card type section. **PWM PERIOD, PWM0 DUTY CYCLE, PWM1 DUTY CYCLE** and **TIME BASE** parameters descriptions are given in section 4.1.4

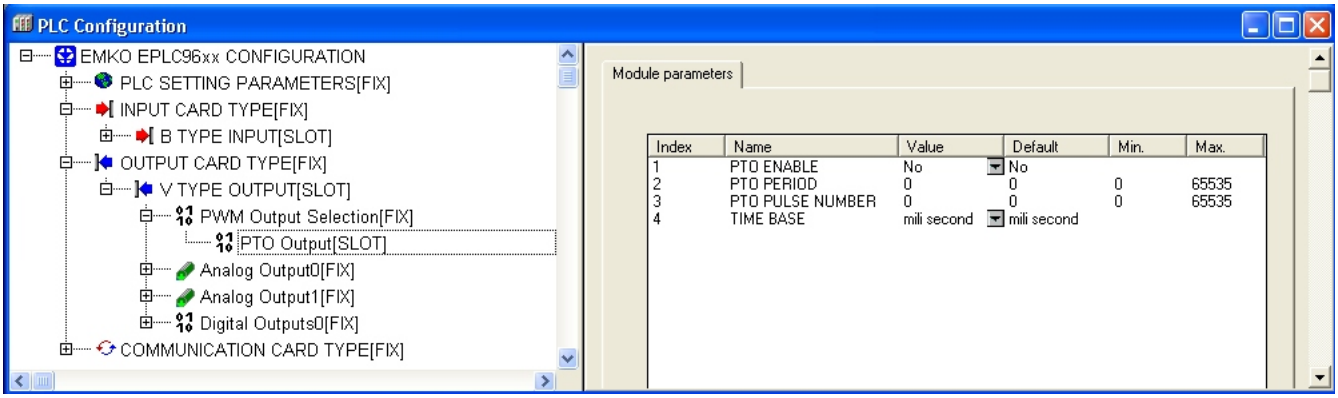
Figure 11: PWM outputs settings



3.1.10 PTO Ouput Setting Parameters:

User can adjust the PTO output settings for only single segment PTO output under the output card type section. Multiple segment PTO output settings and **PTO PERIOD, PTO PULSE NUMBER** and **TIME BASE** parameters descriptions are given in section 4.1.4

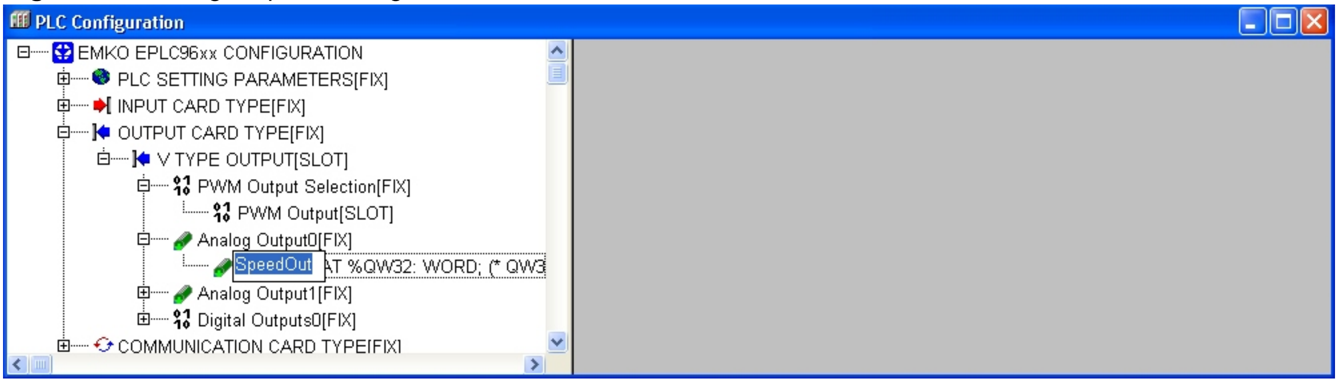
Figure 12: PTO outputs settings



3.1.11 Analog Output Setting:

User can see the Analogue output adresses and when pressing the name section like a showing below figure, can change the analogue output name for using PLC program.

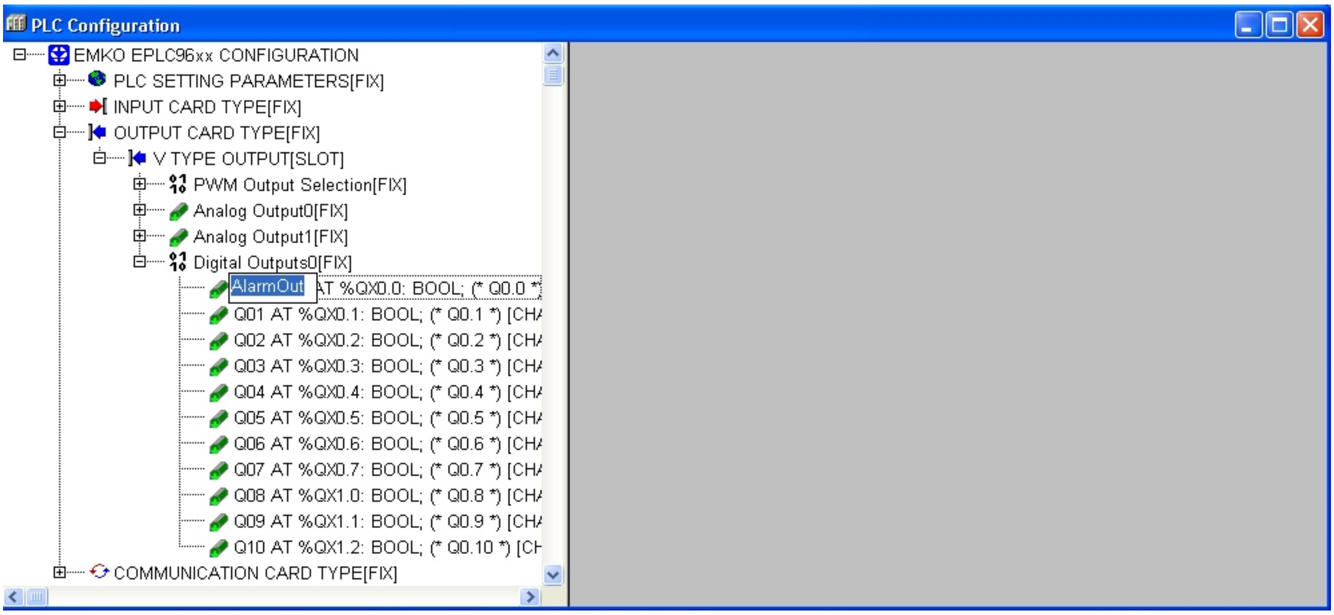
Figure 13: Analog outputs settings



3.1.12 Digital Output Setting:

User can see the digital output adresses and when pressing the name section like a showing below figure, can change the digital output name for using PLC program.

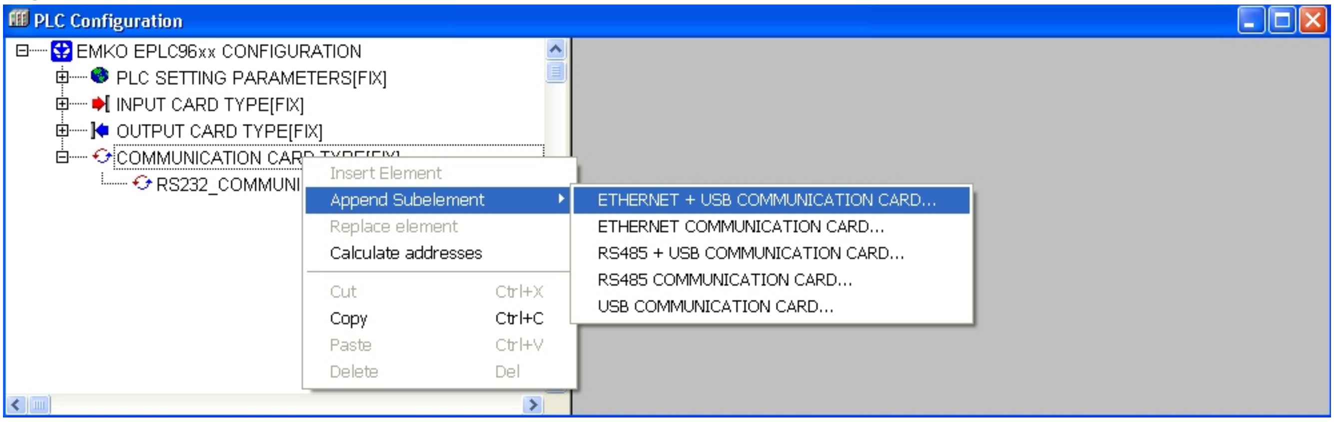
Figure 14: Digital outputs settings



3.1.13 Communication Card Type Selection:

Select the **COMMUNICATION CARD TYPE** elements according to the section "**1.3 Selecting of elements**". When rigth clicked the mouse on COMMUNICATION CARD TYPE element, the properties section of this element is opened and pressing on the **Append Subelement** section, available communication card types are listed. User can select the communication card type which is plugged on PLC.

Figure 15: Communicaiton card selection

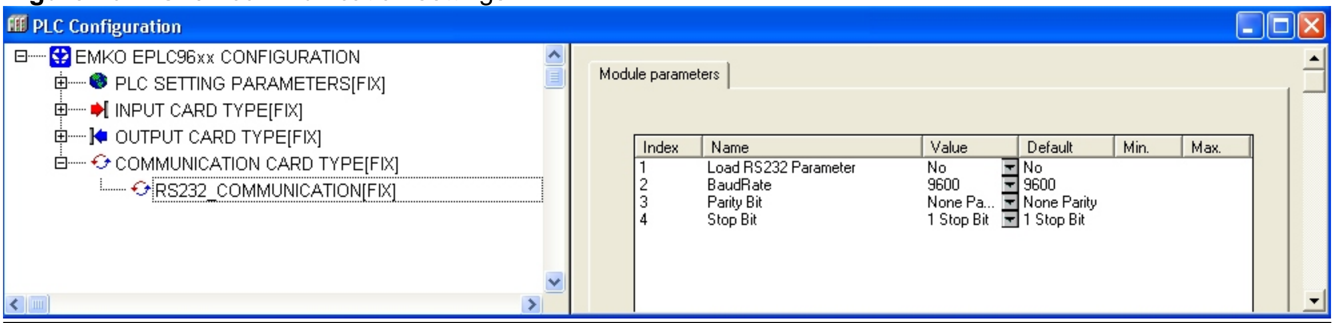


3.1.14 RS232 Communication Settings:

User can adjust **BaudRate**, **Parity Selection** and **Stop Bit Selection** parameter for RS232 Communication in this section. If the **Load RS232 Parameter** value is **YES**, then RS232 settings are updated while the PLC program loading.

If the **Load RS232 parameter** value is **NO**, then RS232 settings are saving their old value.

Figure 16: RS232 communication settings

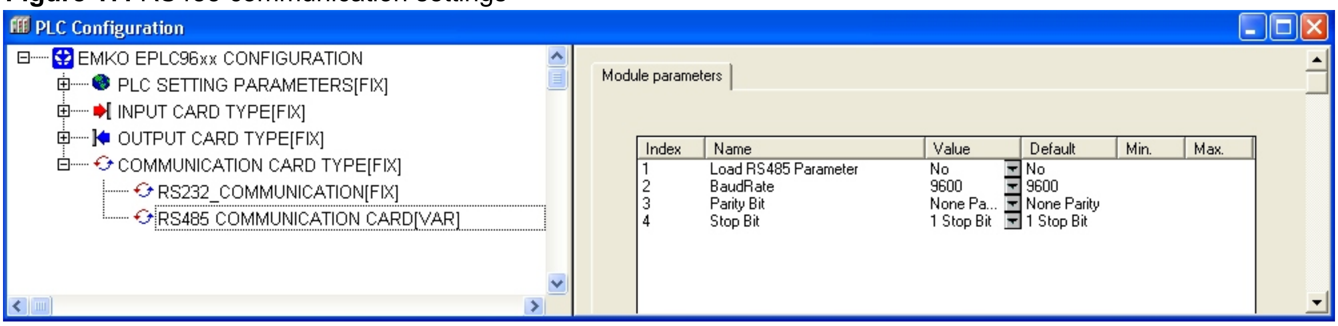


3.1.15 RS485 Communication Settings:

User can adjust **BaudRate**, **Parity Selection** and **Stop Bit Selection** parameter for RS485 Communication in this section. If the **Load RS485 Parameter** value is **YES**, then RS485 settings are updated while the PLC program loading.

If the **Load RS485 parameter** value is **NO**, then RS485 settings are saving their old value.

Figure 17: RS485 communication settings

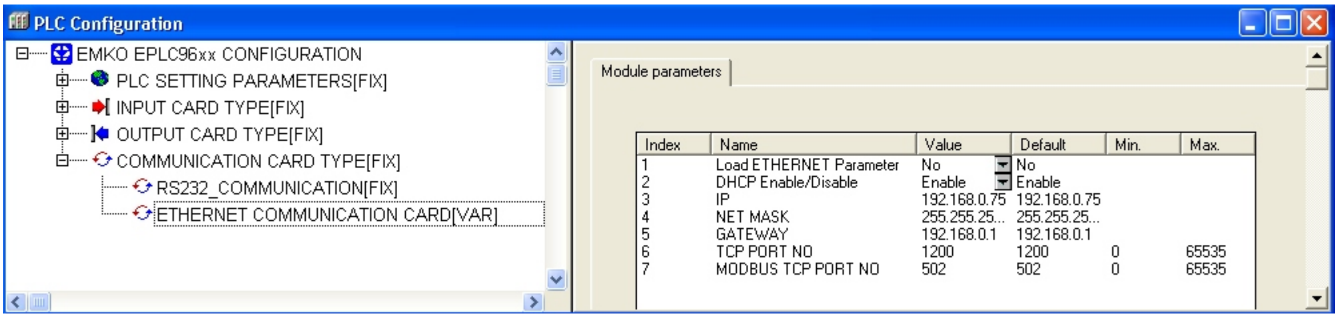


3.1.16 Ethernet Communication Settings:

User can adjust **DHCP Enable/Disable, IP, Net Mask, Gateway, TCP Port** and **Modbus Tcp Port** parameter for Ethernet Communication in this section. If the **Load Ethernet Parameter** value is **YES**, then Ethernet settings are updated while the PLC program loading.

If the **Load Ethernet parameter** value is **NO**, then Ethernet settings are saving their old value.

Figure 18: ETHERNET communication settings

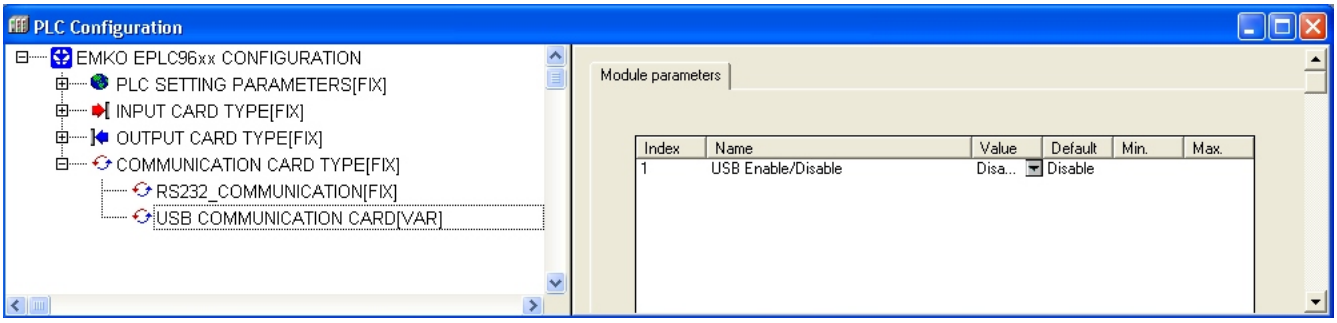


3.1.17 USB Communication Settings:

User can enable or disable USB function in this section parameter.

İf the **USB Enable/Disable paramater** value is Enable, then the USB file record function is enabled for PLC program otherwise the USB file record function is disabled.

Figure 19: USB communication settings



4 EPLC96XX TARGET SPECIFICATIONS

4.1 EPLC96XX Memory Layout

EPLC96XX PLC has a 5 different memory areas. These are like below,

Program memory
Input Memory
Output Memory
Data Memory
Retain Data Memory

The direct display of individual memory locations is done through the use of special character sequences. These sequences are a concatenation of the percent sign "%", a range prefix, a prefix for the size and one or more natural numbers separated by blank spaces.
The following range prefixes are supported:

I	Input
Q	Output
M	Memory location

The following size prefixes are supported:

X	Single bit
None	Single bit
B	Byte (8 Bits)
W	Word (16 Bits)
D	Double word (32 Bits)

Example:

%QX7.5	Output bit 7.5
%Q7.5	Output bit 7.5
%IW215	Input word 215
%QB7	Output byte 7
%MD48	Double word in memory position 48 in the memory location.

4.1.1 Program Memory

The total program memory size for EPLC96XX PLC is **276kByte**.

4.1.2 Input Memory

The total Input Memory size for EPLC96XX PLC is **256Byte.** Input memory area is allocated as byte addressing mode and accessing scheme for Input memory area is showing in figure 20.

Figure 20: Input memory area accessing scheme

<table>
<tr><td>%IX0.0
%IX0.7</td><td>%IX1.0
%IX1.7</td><td>%IX2.0
%IX2.7</td><td>%IX3.0
%IX3.7</td><td>%IX4.0
%IX4.7</td><td>%IX5.0
%IX5.7</td><td>%IX6.0
%IX6.7</td><td>%IX7.0
%IX7.7</td><td>%IX8.0
%IX8.7</td><td>%IX9.0
%IX9.7</td></tr>
<tr><td>%IB0</td><td>%IB1</td><td>%IB2</td><td>%IB3</td><td>%IB4</td><td>%IB5</td><td>%IB6</td><td>%IB7</td><td>%IB8</td><td>%IB9</td></tr>
<tr><td></td><td colspan="2">%IW1</td><td colspan="2">%IW3</td><td colspan="2">%IW5</td><td colspan="2">%IW7</td><td>%IW9</td></tr>
<tr><td colspan="2">%IW0</td><td colspan="2">%IW2</td><td colspan="2">%IW4</td><td colspan="2">%IW6</td><td colspan="2">%IW8</td></tr>
<tr><td colspan="4">%ID0</td><td colspan="4">%ID4</td><td colspan="2">%ID8</td></tr>
<tr><td></td><td colspan="4">%ID1</td><td colspan="4">%ID5</td><td>%ID9</td></tr>
<tr><td colspan="2"></td><td colspan="4">%ID2</td><td colspan="4">%ID6</td></tr>
<tr><td colspan="3"></td><td colspan="4">%ID3</td><td colspan="3">%ID7</td></tr>
</table>

Input memory area is divide in 3 different group.

* **Digital Input address :** All digital inputs is defined with their own BIT address.
  Max 256 digital input is can be defined.
* **High Speed Counter Input address :** All HSC input is defined their own DWORD address
  Max 8 HSC input is can be defined.
* **Analogue Input address :** All analogue input is defined their own WORD address
  Max 96 Analogue input is can be defined.

The input memory allocation is given in figure 21.

%IX0.0	1. Digital Input address
%IX0.1	2. Digital Input address
%IX0.2	3. Digital Input address
%IX31.5	254. Digital Input address
%IX31.6	255. Digital Input address
%IX31.7	256. Digital Input address
%ID32	1. High Speed Counter (HSC) Value addrees
%ID36	2. High Speed Counter (HSC) Value addrees
%ID56	7. High Speed Counter (HSC) Value addrees
%ID60	8. High Speed Counter (HSC) Value addrees
%IW64	1. Analogue Input addrees
%IW66	2. Analogue Input addrees
%IW68	3. Analogue Input addrees
%IW252	95. Analogue Input addrees
%IW254	96. Analogue Input addrees

Figure 21: input memory allocation.

4.1.3 Output Memory

The total output memory size for EPLC96XX PLC is **256Byte.** Output memory area is allocated as byte addressing mode and accessing scheme for output memory area is showing in figure 22.

Figure 22: Output memory area accessing scheme

<table>
<tr><td>%QX0.0
%QX0.7</td><td>%QX1.0
%QX1.7</td><td>%QX2.0
%QX2.7</td><td>%QX3.0
%QX3.7</td><td>%QX4.0
%QX4.7</td><td>%QX5.0
%QX5.7</td><td>%QX6.0
%QX6.7</td><td>%QX7.0
%QX7.7</td><td>%QX8.0
%QX8.7</td><td>%QX9.0
%QX9.7</td></tr>
<tr><td>%QB0</td><td>%QB1</td><td>%QB2</td><td>%QB3</td><td>%QB4</td><td>%QB5</td><td>%QB6</td><td>%QB7</td><td>%QB8</td><td>%QB9</td></tr>
<tr><td></td><td colspan="2">%QW1</td><td colspan="2">%QW3</td><td colspan="2">%QW5</td><td colspan="2">%QW7</td><td>%QW9</td></tr>
<tr><td colspan="2">%QW0</td><td colspan="2">%QW2</td><td colspan="2">%QW4</td><td colspan="2">%QW6</td><td colspan="2">%QW8</td></tr>
<tr><td colspan="4">%QD0</td><td colspan="4">%QD4</td><td colspan="2">%QD8</td></tr>
<tr><td></td><td colspan="4">%QD1</td><td colspan="4">%QD5</td><td>%QD9</td></tr>
<tr><td colspan="2"></td><td colspan="4">%QD2</td><td colspan="4">%QD6</td></tr>
<tr><td colspan="3"></td><td colspan="4">%QD3</td><td colspan="3">%QD7</td></tr>
</table>

Output memory area is divide in 2 different group.

* **Digital Output address :** All digital outputs is defined with their own BIT address. Max 256 digital output is can be defined.
* **Analogue Output address :** All analogue output is defined their own WORD address Max 112 Analogue output is can be defined.

The output memory allocation is given in figure 23.

%QX0.0	1. Digital output address
%QX0.1	2. Digital output address
%QX0.2	3. Digital output address
%QX31.5	254. Digital output address
%QX31.6	255. Digital output address
%QX31.7	256. Digital output address
%QW32	1. Analogue output addrees
%QW34	2. Analogue output addrees
%QW36	3. Analogue output addrees
%QW252	111. Analogue output addrees
%QW254	112. Analogue output addrees

Figure 23: output memory allocation.

4.1.4 Marker Memory

Marker memory data provides status and control functions and performs the communication between the EPLC96XX with the program. Marker memory is accessible as a BIT, BYTE, WORD or DWORD addresses. The total marker memory size for EPLC96XX PLC is **256Byte.** Accessing scheme for marker memory area is showing in figure 24.

Figure 24: Marker memory area accessing scheme

<table>
<tr><td>%MX0.0
%MX0.7</td><td>%MX1.0
%MX1.7</td><td>%MX2.0
%MX2.7</td><td>%MX3.0
%MX3.7</td><td>%MX4.0
%MX4.7</td><td>%MX5.0
%MX5.7</td><td>%MX6.0
%MX6.7</td><td>%MX7.0
%MX7.7</td><td>%MX8.0
%MX8.7</td><td>%MX9.0
%MX9.7</td></tr>
<tr><td>%MB0</td><td>%MB1</td><td>%MB2</td><td>%MB3</td><td>%MB4</td><td>%MB5</td><td>%MB6</td><td>%MB7</td><td>%MB8</td><td>%MB9</td></tr>
<tr><td></td><td colspan="2">%MW1</td><td colspan="2">%MW3</td><td colspan="2">%MW5</td><td colspan="2">%MW7</td><td>%MW9</td></tr>
<tr><td colspan="2">%MW0</td><td colspan="2">%MW2</td><td colspan="2">%MW4</td><td colspan="2">%MW6</td><td colspan="2">%MW8</td></tr>
<tr><td colspan="4">%MD0</td><td colspan="4">%MD4</td><td colspan="2">%MD8</td></tr>
<tr><td></td><td colspan="4">%MD1</td><td colspan="4">%MD5</td><td>%MD9</td></tr>
<tr><td colspan="2"></td><td colspan="4">%MD2</td><td colspan="4">%MD6</td></tr>
<tr><td colspan="3"></td><td colspan="4">%MD3</td><td colspan="3">%MD7</td></tr>
</table>

%MB0 : Status Bits.

As describe in table 1, there are 6 bits in %MB0 which are updated every PLC cycle. The last two bits are reserved for future version.

Table 1: %MB0 status bits description

%MB0 Bits	Description (Read Only)
%MX0.0	This bits is always 1 during programm running
%MX0.1	This bits is 1 for one cycle after the PLC is returning the **Run** mode from **Stop** mode.
%MX0.2	This bits is 1 for one cycle after the PLC is first powered up.
%MX0.3	This bits is a blinker with 1 second period. (0.5 second **ON** and 0.5 second **OFF**)
%MX0.4	This bits is a blinker with 1 minute period. (30 second **ON** and 30 second **OFF**)
%MX0.5	This bits is a toogled bit. It’s value is 1 for one cycle, 0 for next cycle and so on.
%MX0.6	Reserved
%MX0.7	Reserved

%MB1 : Status Bits.

As describe in table 2, there are 8 bits in %MB1 which are reserved for future version.

Table 2: %MB1 status bits description

%MB1 Bits	Description
%MX1.0	Reserved
%MX1.1	Reserved
%MX1.2	Reserved
%MX1.3	Reserved
%MX1.4	Reserved
%MX1.5	Reserved
%MX1.6	Reserved
%MX1.7	Reserved

%MB2 : Status Bits.

As describe in table 3, there are 7 bits in %MB2 . The last bit is reserved for future version.

Table 3: %MB2 status bits description

%MB2 Bits	Description
%MX2.0	LCD backligth ON/OFF controlling bit. If 1 LCD backligth is **ON** otherwise LCD Backligth is **OFF**.
%MX2.1	RS232 Modbus Master/Slave selection bit. If this bit is 1 PLC actions as a master for Modbus, otherwise PLC actions as a slave for modbus communication.
%MX2.2	RS485 Modbus Master/Slave selection bit. If this bit is 1 PLC actions as a master for Modbus, otherwise PLC actions as a slave for modbus communication.
%MX2.3	Buzzer active bit. If 1 buzzer is active, otherwise buzzer is passive.
%MX2.4	Data writing to USB memory stick is completed bit. If 0 usb wirting is completed and new data is enable for writing otherwise data writing to the USB memory stick is still continue.
%MX2.5	USB memory stick connection status bit. If 1 USB memory stick is connected to the PLC otherwise USB memory stick is not connected to PLC.
%MX2.6	DHCP function return bit. If Ethernet DHCP spesification is enabled, this bit can be used for control the DHCP function. If 1 DHCP function is return with success, otherwise DHCP function is return wtih unsucces.
%MX2.7	Motorized valve control enable. (see section Motorized valve controller module)

%MB3 : Status Bits.

As describe in table 4, there are 8 bits in %MB3 which are reserved for future version.

Table 4: %MB3 status bits description

%MB3 Bits	Description
%MX3.0	Motorized valve adjust flag. (see section Motorized valve controller module)
%MX3.1	Reserved
%MX3.2	Reserved
%MX3.3	Reserved
%MX3.4	Reserved
%MX3.5	Reserved
%MX3.6	Reserved
%MX3.7	Reserved

%MB4 : Key Status Bits.

As describe in table 5, there are 8 bits in %MB4 which are updated every PLC cycle.

Table 5: %MB4 status bits description

%MB4 Bits	Description (Read Only)
%MX4.0	Key 0 status bit. During the key 0 is pressing this bit value is read as 1, otherwise read as 0
%MX4.1	Key 1 status bit. During the key 1 is pressing this bit value is read as 1, otherwise read as 0
%MX4.2	Key 2 status bit. During the key 2 is pressing this bit value is read as 1, otherwise read as 0
%MX4.3	Key 3 status bit. During the key 3 is pressing this bit value is read as 1, otherwise read as 0
%MX4.4	Key 4 status bit. During the key 4 is pressing this bit value is read as 1, otherwise read as 0
%MX4.5	Key 5 status bit. During the key 5 is pressing this bit value is read as 1, otherwise read as 0
%MX4.6	Key 6 status bit. During the key 6 is pressing this bit value is read as 1, otherwise read as 0
%MX4.7	Key 7 status bit. During the key 7 is pressing this bit value is read as 1, otherwise read as 0

%MB5 : Key Status Bits.

As describe in table 6, there are 8 bits in %MB5 which are updated every PLC cycle.

Table 6: %MB5 status bits description

%MB5 Bits	Description (Read Only)
%MX5.0	Key 8 status bit. During the key 8 is pressing this bit value is read as 1, otherwise read as 0.
%MX5.1	Key 9 status bit. During the key 9 is pressing this bit value is read as 1, otherwise read as 0.
%MX5.2	Key UP status bit. During the key UP is pressing this bit value is read as 1, otherwise read as 0.
%MX5.3	Key DOWN status bit. During the key DOWN is pressing this bit value is read as 1, otherwise read as 0.
%MX5.4	Key RIGHT status bit. During the key RIGHT is pressing this bit value is read as 1, otherwise read as 0.
%MX5.5	Key LEFT status bit. During the key LEFT is pressing this bit value is read as 1, otherwise read as 0.
%MX5.6	Key ENTER status bit. During the key ENTER is pressing this bit value is read as 1, otherwise read as 0.
%MX5.7	Key ESCAPE status bit. During the key ESCAPE is pressing this bit value is read as 1, otherwise read as 0.

%MB6 : Key Status Bits.

As describe in table 7, there are 8 bits in %MB6 which are updated every PLC cycle. The last 4 bits are reserved for future version.

Table 7: %MB6 status bits description

%MB6 Bits	Description
%MX6.0	Key SIGN/POINT status bit. During the key SIGN/POINT is pressing this bit value is read as 1, otherwise read as 0.
%MX6.1	Key DELETE status bit. During the key DELETE is pressing this bit value is read as 1, otherwise read as 0.
%MX6.2	Variable value changing screen active bit. This bit value is read as a 1, when the PLC visu screen is show the variable value chaning screen, otherwise read as a 0.
%MX6.3	Keypad status bit. If this bit's value is 1, Keys reading are disable in variable value changing screen, otherwise keys reading are enable.
%MX6.4	Reserved.
%MX6.5	Reserved.
%MX6.6	Reserved.
%MX6.7	Reserved.

%MB7 : Status Bits.

As describe in table 8, there are 8 bits in %MB7 which are reserved for future version.

Table 8: %MB7 status bits description

%MB7 Bits	Description
%MX7.0	Reserved
%MX7.1	Reserved
%MX7.2	Reserved
%MX7.3	Reserved
%MX7.4	Reserved
%MX7.5	Reserved
%MX7.6	Reserved
%MX7.7	Reserved

%MB8.....%MB10 Reserved Marker Memory
Marker memory area between the %MB8 and %MB10 are reserved for future version.

Table 9: %MB8....%MB10 Reserved marker memory.

%M Memory	Explanation
%MB8	Reserved
%MB9	Reserved
%MB10	Reserved

%MB11 Lcd Contrast Value :
User can be adjust LCD contrast value as writing a value in this marker memory area.
Minumum value for this variable data is 30 (low lcd contrast .)
Maximum value for this variable data is 60 (high lcd contrast .)

Table 10: %MB11 LCD contrast value

%M Memory	Description(Read/Write)
%MB11	LCD contrast value. The user can write and read this variable data area.

%MD12 PLC Cycle Time Value (msn):
User can be read the last PLC cycle time value in this marker memory area. This value is updated every PLC cycle.

Table 11: %MD12 PLC cycle time

%M Memory	Description(Read only)
%MD12	Last plc cycle time

%MD16 10 msn Time-Based Register:
This register value is incremented every 10 msn after first powered on the PLC unit.

Table 12: %MD16 10msn time base resigter

%M Memory	Description(Read only)
%MD16	10msn time base register

%MB20 Timer interrupt interval(msn):
User can describe a timer interrupt for PLC program. Interrupt interval is can be describe in this marker memory area. User can adjust this value between 1(msn) and 255(msn) value.
If this value is 0 then timer interrupt function is not actived in PLC.

Table 13: %MB20 Timer interrupt interval

%M Memory	Description(Read/Write)
%MB20	Timer interrupt interval value.

%MB21.....%MB27 Reserved Marker Memory
Marker memory area between the %MB21 and %MB27 are reserved for future version.

Table 14: %MB21....%MB27 Reserved marker memory.

%M Memory	Description
%MB21	Reserved
%MB22	Reserved
%MB23	Reserved
%MB24	Reserved
%MB25	Reserved
%MB26	Reserved
%MB27	Reserved

MODBUS

MODBUS enables you to establish master-slave communications with any connected device that supports the MODBUS protocol. Any controller in the network may function as either master or slave using any of the controller's existing COM Ports **(RS232, RS485, Ethernet "only slave").**

Before you can run a MODBUS command, you must configure MODBUS parameters for both Master and Slave devices.

EPLC96XX PLC unit currently supports RTU (binary) transmission mode. The supported modbus funciton is listed in table 15.

Table 15: Supported Modbus Master/Slave Protocol Functions.

Function No	Description
1	Read Coils (read digital output).
2	Read Discrete Inputs (read digital input).
3	Read Holding Register (read data memory).
4	Read Input Register (read analogue inputs)
5	Write Single Coils (write one digital output).
6	Write Single Register (write one data memory.)
15	Write Multiple Coils (write multiple digital output.)
16	Write Multiple Registers (write multiple data memory.)

You can define two different data memory and data memory min/max table for modbus function **3, 6** and **16**. The related marker memory for these tables addresses are describe in Table 16.

Table 16: Data memory tables addresses for modbus functions

Function No	Description
%MD28	Data memory table-1 start address
%MD32	Data memory table-2 start address
%MD36	Min/Max table start adress for table-1 data memory
%MD40	Min/Max table start adress for table-2 data memory
%MW44	Data memory table-1 size
%MW46	Data memory table-2 size

RS232 Modbus Settings

EPLC96XX PLC is actions as a master or slave in modbus communication. As describe in table 3. %MX2.1 bit is used as a master/slave selection for RS232 modbus communicaiton. For modbus communication over RS232 port, the Slave ID (%MB49)value is must be defined between 1 and 247.
85 and **170** Slave ID values are reserved for RS232 port, user is can not defined the Slave ID with this values.

Table 17: RS232 Modbus Settings.

%M Memory	Description
%MB49	Slave ID value.
%MW50	Slave Start Address (enable if PLC selected as a MASTER). It is used for determining the datas starting address wihch are reading/writing in slave device with modbus communication.
%MW52	Slave Data Number (enable if PLC selected as a MASTER). It is used for determining the how many datas are reads/writes from slave device with modbus communication.
%MW54	Master Start Address (enable if PLC selected as a MASTER). It is used for determining the datas starting address which are writing in master device with modbus communication.
%MB56	Master Memory Area (enable if PLC selected as a MASTER). It is used for determining the datas memory area which are writing in master device with modbus communication. **0 : Global Memory Area, 1: Input Memory Area, 2 : Output Memory Area**
%MB57	Modbus Function Number (enable if PLC selected as a MASTER). User must be write in this data area with one of supported modbus function number.
%MB58	Modbus Status Register.
%MB70	Time Out : (10 msecs base) a Time out value of 100 is equal to 1 second. This is the amount of time a master device will wait for an answer from a slave.
%MB71	Retries Number. This is the number of times a device will try to send a message
%MW72	Time Out Delay Time: (1msec base). The end of the modbus message time out is delayed with specified number of milliseconds

RS485 Modbus Settings

EPLC96XX PLC is actions as a master or slave in modbus communication. As describe in table 3. %MX2.2 bit is used as a master/slave selection for RS485 modbus communicaiton. For modbus communication over RS485 port, the Slave ID (%MB59)value is must be defined between 1 and 247.

Table 18: RS485 Modbus Settings.

%M Memory	Description
%MB59	Slave ID value.
%MW60	Slave Start Address (enable if PLC selected as a MASTER). It is used for determining the datas starting address wihch are reading/writing in slave device with modbus communication.
%MW62	Slave Data Number (enable if PLC selected as a MASTER). It is used for determining the how many datas are reads/writes from slave device with modbus communication.
%MW64	Master Start Address (enable if PLC selected as a MASTER). It is used for determining the datas starting address which are writing in master device with modbus communication.
%MB66	Master Memory Area (enable if PLC selected as a MASTER). It is used for determining the datas memory area which are writing in master device with modbus communication. **0 : Global Memory Area, 1: Input Memory Area, 2 : Output Memory Area**
%MB67	Modbus Function Number (enable if PLC selected as a MASTER). User must be write in this data area with one of supported modbus function number.
%MB68	Modbus Status Register.
%MB74	Time Out : (10 msecs base) a Time out value of 100 is equal to 1 second. This is the amount of time a master device will wait for an answer from a slave.
%MB75	Retries Number. This is the number of times a device will try to send a message
%MW76	Time Out Delay Time: (1msec base). The end of the modbus message time out is delayed with specified number of milliseconds

Modbus Status Message Description.
User can be known the status of the modbus communication function as a reading the status register which are %MB58 for RS232 and %MB68 for RS485 modbus communication.
In table 19 modbus status registers description is listed.

Table 19: Modbus status registers descriptions.

Error No	Description
0	**No Error** Modbus query is ended in succes.
1	**Illegal Function** Modbus function number is not supported in EPLC96XX PLC.
2	**Illegal Data Address** Data request command includes invalid addresses.
3	**Illegal Data Memory Area** Requested data type with modbus query is not suitable with master memory area.
4	**Illegal Data Number** Requested data number is too much or exceed the limit. Max 120 WORDS is reading or writing in one modbus query. Max 1900 BITS is reading or writing in one modbus query.
5	**Illegal Data Value** Data value in Modbus query is exceed the limits.
6	**Synchronization Error** Modbus query is response is incomplete, Frame Error or Data Overrun error is detected.
7	**Illegal ID** the value in %MB49 for RS232 or %MB59 for RS485 is illegal.
8	**CRC Error** Crc Error is detected.
9	**Parity Error** Parity error is detected.
10	**Master Time Out Error** No response from the slave device for modbus query.
11	**Communication Link Error** If master device is not get response from the slave device as a retries number of times.

MOTORIZED VALVE CONTROLLER MODULE

When motorized valve controller module is enable, digital out-2(%QX0.2) is operates for opening the valve, digital out-3(%QX0.3) is operates for closing the valve and this outputs status is calculated automaticaly according to the valve set position(%MW80) and valve current position (%MW78) values.

Table 20: Motorized valve controller module variable

%M Memory	Description
%MX2.7	**Motorized valve controller module enable** Motorized valve controller module can be enable via in this flag.
%MX3.0	**Motorized valve adjust flag** Motorized valve position can be adjust via in this flag. when this flag is set motorized valve is move minimum step value to the closing direction while valve current position value is not changing. This flag is clear automatically.
%MW78	**Valve current position (% ratio)** current valve position can be read via in this register. this register value is between 0 and 1000 0 : is mean motorized valve is %0 position, 1000 : is mean motorized valve is %100.0 position
%MW80	**Valve set position (% ratio)** valve set position is set via in this register value.
%MW82	**Valve boundly/move time (seconds)** It defines after how many seconds valve is completely opened.
%MW84	**Valve dead band(% ratio)** Minimum movement steps of valve while opening or closing are determined as % ratio.

%MB86.....%MB99 Reserved Marker Memory

Marker memory area between the %MB86 and %MB99 are reserved for future version.

Table 21: %MB86....%MB99 Reserved marker memory.

%M Memory	Description
%MB86	Reserved
%MB87	Reserved
%MB88	Reserved
%MB89	Reserved
%MB90	Reserved
%MB91	Reserved
%MB92	Reserved
%MB93	Reserved
%MB94	Reserved
%MB95	Reserved
%MB96	Reserved
%MB97	Reserved
%MB98	Reserved
%MB99	Reserved

HIGH SPEED COUNTER (HSC)

High-speed counters count high speed events that cannot be controlled at EPLC96XX scan rates.
The maximum counting frequency of a high-speed counter is 20kHz for Encoder inputs and 30kHz for counter inputs.

There are five basic types of counters: single-phase counter with internal direction control, single-phase counter with external direction control, two-phase counter with 2 clock inputs, A/B phase quadrature counter and frequency measurement type. Note that every mode is not supported by every counter. You can use each type except the frequency measurement type: without reset or start inputs, with reset and without start, or with both start and reset inputs.

* When you activate the reset input, it clears the current value and holds it clear until you deactivate reset.

* When you activate the start input, it allows the counter to count. While start is deactivated, the current value of the counter is held constant and clocking events are ignored.

* If reset is activated while start is inactive, the reset is ignored and the current value is not changed. If the start input becomes active while the reset input is active, the current value is cleared.

Table 22: HSC modes.

Mode	Description	Inputs			
	(For G and H type Input)HSC0	%IX0.0	%IX0.1	%IX0.2	
	(For C and E type Input)HSC0	%IX0.0	%IX0.1	%IX0.2	%IX0.3
	(For A and B type Input)HSC0	%IX0.0	%IX0.1	%IX0.2	%IX0.5
	(For A and B type Input)HSC1	%IX0.3	%IX0.4	%IX0.6	%IX0.7
0	Single phase counter with internal direction	Clock			
1		Clock		Reset	
2		Clock		Reset	Start
3	Single phase counter with external direction	Clock	Direction		
4		Clock	Direction	Reset	
5		Clock	Direction	Reset	Start
6	Two phase counter with 2 clock input	Clock Up	Clock Down		
7		Clock Up	Clock Down	Reset	
8		Clock Up	Clock Down	Reset	Start
9	A/B phase Encoder counter	Clock A	Clock B		
10		Clock A	Clock B	Reset	
11		Clock A	Clock B	Reset	Start
12	Frequency Measurement (with 100 msn sampling time)	Frequency input			
13	Frequency Measurement (with 500 msn sampling time)	Frequency input			

Note:
* Mode 12 and mode 13 is not supported for HSC1 counter.
* HSC1 is not supported for C, E, G and H type input card.
* Mode 2,5,8 and 11 is not supported for G and H type input card.

HSC Parameter:
After you define the counter and the counter mode, you can program the dynamic parameters of the counter. Each high-speed counter has a control byte that allows the following actions:
* Enabling or disabling the counter
* Controlling the direction (modes 0, 1, and 2 only), or the initial counting direction for all other modes
* Loading the current value

Examination of the control byte and associated current is invoked by the execution of the HSC instruction. Table 24 describes each of these control bits.

Table 23: HSC control parameter description

%M Memory	Description
%MD100	New current value for HSC0.
%MD104	New current value for HSC1.
%MB132	Configuration value for HSC0.
%MB133	Configuration value for HSC1.

Table 24: HSC configuration byte description

HSC0	HSC1	Description
%MX132.0	%MX133.0	**Active level control bit for Reset:** 0 = Reset is active low 1 = Reset is active high
%MX132.1	%MX133.1	**Active level control bit for Start:** 0 = Start is active low 1 = Start is active high
%MX132.2	%MX133.2	**Counting direction control bit:** 0 = Count down 1 = Count up
%MX132.3	%MX133.3	**Write the new current value to the HSC:** 0 = No update 1 = Update current value
%MX132.4	%MX133.4	**Enable the HSC:** 0 = Disable the HSC 1 = Enable the HSC
%MX132.5	%MX133.5	
%MX132.6	%MX133.6	
%MX132.7	%MX133.7	

%MB108.....%MB131 Reserved Marker Memory
Marker memory area between the %MB108 and %MB131 are reserved for future version.

Table 25: %MB108....%MB131 Reserved marker memory.

%M Memory	Description
%MB108	Reserved
%MB109	Reserved
%MB110	Reserved
%MB111	Reserved
%MB112	Reserved
%MB113	Reserved
%MB114	Reserved
%MB115	Reserved
%MB116	Reserved
%MB117	Reserved
%MB118	Reserved
%MB119	Reserved

%MB108.....%MB131 Reserved Marker Memory

Marker memory area between the %MB108 and %MB131 are reserved for future version.

Table 25: %MB108....%MB131 Reserved marker memory.

%M Memory	Description
%MB120	Reserved
%MB121	Reserved
%MB122	Reserved
%MB123	Reserved
%MB124	Reserved
%MB125	Reserved
%MB126	Reserved
%MB127	Reserved
%MB128	Reserved
%MB129	Reserved
%MB130	Reserved
%MB131	Reserved

%MB134.....%MB179 Reserved Marker Memory

Marker memory area between the %MB134 and %MB179 are reserved for future version.

Table 26: %MB134....%MB179 Reserved marker memory.

%M Memory	Description
%MB134	Reserved
%MB135	Reserved
%MB136	Reserved
%MB137	Reserved
%MB138	Reserved
%MB139	Reserved
%MB140	Reserved
%MB141	Reserved
%MB142	Reserved
%MB143	Reserved
%MB144	Reserved
%MB145	Reserved
%MB146	Reserved
%MB147	Reserved
%MB148	Reserved
%MB149	Reserved
%MB150	Reserved
%MB151	Reserved
%MB152	Reserved
%MB153	Reserved
%MB154	Reserved
%MB155	Reserved
%MB156	Reserved
%MB157	Reserved
%MB158	Reserved
%MB159	Reserved
%MB160	Reserved
%MB161	Reserved
%MB162	Reserved

%MB134.....%MB179 Reserved Marker Memory

Marker memory area between the %MB134 and %MB179 are reserved for future version.

Table 26: %MB134....%MB179 Reserved marker memory.

%M Memory	Description
%MB163	Reserved
%MB164	Reserved
%MB165	Reserved
%MB166	Reserved
%MB167	Reserved
%MB168	Reserved
%MB169	Reserved
%MB170	Reserved
%MB171	Reserved
%MB172	Reserved
%MB173	Reserved
%MB174	Reserved
%MB175	Reserved
%MB176	Reserved
%MB177	Reserved
%MB178	Reserved
%MB179	Reserved

PWM/PTO OUTPUT

PTO provides a square wave (50% duty cycle) output with user control of the cycle time and the number of pulses. PTO can produce either a single train of pulses or multiple trains of pulses (using a pulse profile). You specify the number of pulses and the cycle time (in either microsecond or millisecond increments):

PWM provides a continuous, variable duty cycle output with user control of the cycle time and the pulse width.

The EPLC96XX PWM module can be configurated as a two PWM generator (PWM period must be same but duty cycle value can be different) or one PTO generator.

One PWM generator is assigned to digital output Q0.0, and the other PWM generator is assigned to digital output Q0.1. PTO generator is assigned to the digital output Q0.0

Table 27: PWM module control parameter description

%M Memory	Description	
%MW180	PWM0/1 or PTO cycle time value	range: 2 to 65,535
%MW182	PWM0 pulse width and PTO pulse count value	range: 0 to 65,535
%MW184	PWM1 pulse width	range: 0 to 65,535
%MB186	Total number of segments	Multiple-segment PTO operation only
%MB187	Number of the segment in progress	Multiple-segment PTO operation only
%MD188	Starting location of the profile table	Multiple-segment PTO operation only
%MB192	PWM / PTO modul control byte	

Table 28: PWM / PTO control byte description

Error No	Description
%MX192.0	**PTO/PWM0/1 update the cycle time** 0 = no update 1 = update cycle time
%MX192.1	**PWM0update the pulse width time / PTO update the pulse count value** 0 = no update 1 = update pulse
%MX192.2	**PWM1update the pulse width time** 0 = no update 1 = update pulse
%MX192.3	**PTO/PWM time base** 0 = 1 s/tick 1 = 1 msn/tick
%MX192.4	**PTO/PWM mode select** 0 = PTO 1 = PWM
%MX192.5	**PTO single/multiple segment operation** 0 = single 1 = multiple
%MX192.6	**PTO/PWM0 enable** 0 = disable 1 = enable
%MX192.7	**PWM1 enable** 0 = disable 1 = enable

Table 29: Profile Table Format for Multiple Segment PTO Operation

Word Offset	Segment	Description
0		**Initial cycle time (2 to 32767)**
1	**1**	**Cycle time delta per pulse (-32767 to 32767)**
2		**Pulse count (1 to 32767)**
3		**Initial cycle time (2 to 32767)**
4	**2**	**Cycle time delta per pulse (-32767 to 32767)**
5		**Pulse count (1 to 32767)**
(Continues)	**3**	**Initial cycle time (2 to 32767)**

Table 30: PWM / PTO Control byte description

Control Byte	PWM module status							
	PWM-1 Enable	PTO PWM-0 Enable	PTO Segment Selection	Mode Selection	Time Base	PWM1 Pulse Width Update	PWM-0 Pulse Width / PTO Pulse Count Update	Cycle Time Update
65	-	Yes	Single	PTO	sn	-	-	Yes
66	-	Yes	Single	PTO	sn	-	Yes	-
67	-	Yes	Single	PTO	sn	-	Yes	Yes
73	-	Yes	Single	PTO	ms	-	-	Yes
74	-	Yes	Single	PTO	ms	-	Yes	-
75	-	Yes	Single	PTO	ms	-	Yes	Yes
96	-	Yes	Multiple	PTO	sn	-	-	-
104	-	Yes	Multiple	PTO	ms	-	-	-
81	-	Yes	-	PWM	sn	-	-	Yes
82	-	Yes	-	PWM	sn	-	Yes	-
83	-	Yes	-	PWM	sn	-	Yes	Yes
89	-	Yes	-	PWM	ms	-	-	Yes
90	-	Yes	-	PWM	ms	-	Yes	-
91	-	Yes	-	PWM	ms	-	Yes	Yes
145	Yes	-	-	PWM	sn	-	-	Yes
148	Yes	-	-	PWM	sn	Yes	-	-
149	Yes	-	-	PWM	sn	Yes	-	Yes
153	Yes	-	-	PWM	ms	-	-	Yes
156	Yes	-	-	PWM	ms	Yes	-	-
157	Yes	-	-	PWM	ms	Yes	-	Yes
209	Yes	Yes	-	PWM	sn	-	-	Yes
210	Yes	Yes	-	PWM	sn	-	Yes	-
211	Yes	Yes	-	PWM	sn	-	Yes	Yes
212	Yes	Yes	-	PWM	sn	Yes	-	-
213	Yes	Yes	-	PWM	sn	Yes	-	Yes
215	Yes	Yes	-	PWM	sn	Yes	Yes	Yes
217	Yes	Yes	-	PWM	ms	-	-	Yes
218	Yes	Yes	-	PWM	ms	-	Yes	-
219	Yes	Yes	-	PWM	ms	-	Yes	Yes
220	Yes	Yes	-	PWM	ms	Yes	-	-
221	Yes	Yes	-	PWM	ms	Yes	-	Yes
223	Yes	Yes	-	PWM	ms	Yes	Yes	Yes

EEPROM READ/WRITE FUNCTION

You can save a value stored in data memory to permanent memory (EEPROM) under the control of your program. You can multiple (maximum 255 value) read from the EEPROM but you can write only one value in each PLC cycle.

For reading value from EEPROM, load the address of the data memory to be saved in %MD196 and load the address of the EEPROM location where the data memory saved in %MW194 and load the read size value in %MB179. Then, load %MB193 with the command to read the value.

For writing value to EEPROM , load the address of the data memory to be saved in %MD196 and load the address of the EEPROM location where the data memory saved in %MW194. Then, load %MB193 with the command to write the value.

Once you have loaded the command to save the value to EEPROM, you do not change the value in data memory until the EPLC96XX clear the bit **%M193.7**, indicating that the save operation is complete.

Once you have loaded the command to read the value from EEPROM, you do not change the value in data memory until the EPLC96XX clear the bit **%M193.6**, indicating that the read operation is complete.

As described in Table 31, %MB193 defines the size of the data to be saved to EEPROM memory or to be readed from EEPROM memory and provides the command that initiates a read/write operation.

There is **60Kbyte** EEPROM memory area for user, so you can not write the value bigger than 61440 in %MW194 memory area.

Internal EEPROM module life is 1,000,000 erase/write cycles, so you must not write the EEPROM less than 5 minute interval for minimum 10 years expire date. There is no restriction for reading from EEPROM.

Table 31: EEPROM read/write operation control parameter description

%M Memory	Description
%MB193 EEPROM Command Format	**MSB** w r 0 0 0 0 c1 c0 **LSB**
%MX193.0 and %MX193.1	**c1c0: Size of data** **00 = reserved** **10 = word** **01 = byte** **11 = dword**
%MX193.6	**r: Read from EEPROM** **0 = No reguest for read data from EEPROM** **1 = PLC program request read data from EEPROM** EPLC96XX PLC reset this bit after read operation completed.
%MX193.7	**w: Write to EEPROM** **0 = No reguest for write data to EEPROM** **1 = PLC program request write data to EEPROM** EPLC96XX PLC reset this bit after read operation completed.
%MB179	**Read data size value (between 1 to 255)** This memory area is use only for reading data from the EEPROM. It defines the read size from EEPROM.
%MW194	**EEPROM Address (between 0 to 61440)** This memory area is use for determining the address where data is read from or write to in EEPROM.
%MD196	**Data address** This memory area is use for determinig the data address in PLC data memory which is read from EEPROM or write to EEPROM.

Table 32: EEPROM read/write operation control byte description.

Control Byte	Description
65	Byte read command
66	WORD (2 byte variable) read command
67	DWORD (4 byte variable) read command
129	Byte write command
130	WORD (2 byte variable) write command
131	DWORD (4 byte variable) write command

ETHERNET COMMUNICATION SETTINGS

The Ethernet communication module is used for connecting the EPLC96XX PLC system to Industrial Ethernet (IE). The EPLC96XX PLC can be remotely configured, programmed and diagnosed via Ethernet using CoDeSys V2.3.x. The EPLC96XX PLC is also communicate with an OPC server via Ethernet.

The Ethernet communication module is delivered with a preset, unique worldwide MAC address that cannot be changed.

Table 33: Ethernet module parameters.

%M Memory	Description
%MB200	PLC MAC ADDRESS[0] (read only)
%MB201	PLC MAC ADDRESS[1] (read only)
%MB202	PLC MAC ADDRESS[2] (read only)
%MB203	PLC MAC ADDRESS[3] (read only)
%MB204	PLC MAC ADDRESS[4] (read only)
%MB205	PLC MAC ADDRESS[5] (read only)
%MB206	PLC IP ADDRESS[0] (read only)
%MB207	PLC IP ADDRESS[1] (read only)
%MB208	PLC IP ADDRESS[2] (read only)
%MB209	PLC IP ADDRESS[3] (read only)
%MB210	PLC NET MASK[0] (read only)
%MB211	PLC NET MASK[1] (read only)
%MB212	PLC NET MASK[2] (read only)
%MB213	PLC NET MASK[3] (read only)
%MB214	PLC GATEWAY[0] (read only)
%MB215	PLC GATEWAY[1] (read only)
%MB216	PLC GATEWAY[2] (read only)
%MB217	PLC GATEWAY[3] (read only)
%MW218	PLC Ethernet port number for TCP communication with CoDeSys(read/write)
%MW220	PLC Ethernet port number for Modbus communication with master device

%MB224.....%MB255 User Defined Bit Area

Marker memory area between the %MB224 and %MB255 are reserved for the user.

Table 34: %MB224....%MB255 user defined bit area.

%M Memory	Explanation
%MB224	Reserved for user
%MB225	Reserved for user
%MB226	Reserved for user
%MB227	Reserved for user
%MB228	Reserved for user
%MB229	Reserved for user
%MB230	Reserved for user
%MB231	Reserved for user
%MB232	Reserved for user
%MB233	Reserved for user
%MB234	Reserved for user
%MB235	Reserved for user
%MB236	Reserved for user
%MB237	Reserved for user
%MB238	Reserved for user
%MB239	Reserved for user
%MB240	Reserved for user
%MB241	Reserved for user
%MB242	Reserved for user
%MB243	Reserved for user
%MB244	Reserved for user
%MB245	Reserved for user
%MB246	Reserved for user
%MB247	Reserved for user
%MB248	Reserved for user
%MB249	Reserved for user
%MB250	Reserved for user
%MB251	Reserved for user
%MB252	Reserved for user
%MB253	Reserved for user
%MB254	Reserved for user
%MB255	Reserved for user