Motors | Automation | Energy | Transmission & Distribution | Coatings

SoftPLC

CTW900

User’s Manual

SoftPLC User's Manual

Series: CTW900

Language: English

Document n°: 10002443955 / 00

Publication date: 11/2013

Revision	Description	Chapter
00	First edition	-

CONTENTS

1 ABOUT THE MANUAL

This manual provides the necessary description for the operation of the CA/CC Converter CTW900 using the user programming module denominated SoftPLC. This manual must be used together with the CTW900 User's Manual and with the WLP Software Manual.

1.1 ABBREVIATIONS AND DEFINITIONS

PLC:	Programmable Logic Controller.
CRC:	Cycling Redundancy Check.
RAM:	Random Access Memory.
WLP:	Ladder Language Programming Software.
USB:	Universal Serial Bus.

1.2 NUMERICAL REPRESENTATION

Decimal numbers are represented by means of digits without suffix. Hexadecimal numbers are represented with the letter 'h' after the number.

2 INTRODUCTION TO THE SOFTPLC

The SoftPLC is a feature that incorporates to the CTW900 the functionalities of a PLC, adding flexibility to the product and allowing the user to develop applicative software (user programs).

The SoftPLC main features are:

- Ladder language programming, by using the WLP software.
- Access to all the CTW900 parameters and I/Os.
- 50 configurable user parameters.
- PLC, mathematical and control blocks.
- Applicative software transfer and on-line monitoring via USB.
- Transfer of the installed applicative software to the PC conditioned to a password.
- Storage of the applicative software in the FLASH memory board.
- Execution directly in the RAM memory.

2.1 SYMBOLS AND DATA TYPES

%KW	word type constants (16 bits).
%KF	float type constants (32 bits, floating point).
%MX	bit marker.
%MW	word marker (16 bits).
%MF	float marker (32 bits, floating point).
%SX	system bit marker.
%SW	system word marker (16 bits).
%IX	digital inputs.
%IW	analog inputs (16 bits).
%QX	digital outputs.
%QW	analog outputs (16 bits).

3 SOFTPLC MEMORY

The total size of the SoftPLC memory is 15360 bytes, divided between program memory and data memory. This amount can be reduced depending on the Trace function use.

3.1 MEMORY DIVISION

- Trace function: $15360 \times \frac{P0560}{100}$
- SoftPLC function: $15360 \times \frac{100 - P0560}{100}$

✓ **NOTE!**
P0560 = "Trace Memory", given in percentage. 100.0% is equal to 15360 bytes.

3.2 DATA MEMORY

In the SoftPLC, the data memory area (user variables) is shared with the program memory area. Therefore, the total size of an applicative may vary as a function of the number of variables applied by the user.

The bit, word and float markers are allocated according to the LAST address used in the applicative, i.e., the higher the last address, the bigger the allocated area. Therefore, it is recommended to use the markers in a SEQUENTIAL manner.

The word and float constants do also use the program memory area.

3.2.1 Constants

Table 3.1: *Constant Memory Map*

Sym.	Description	Bytes
%KW	Word Constants (16 bits)	It depends on the quantity of different word constants. E.g.: If there were used: - %KW: 327 = 2 bytes - %KW: 5; 67 = 4 bytes - %KW: 13; 1000; 4 = 6 bytes
%KF	Float Constants (32 bits – IEEE)	It depends on the quantity of different float constants. E.g.: If there were used: - %KF: -0,335 = 4 bytes - %KF: 5,1; 114,2 = 8 bytes - %KF: 0,0; 115,3; 13,333 = 12 bytes

3.2.2 Physical Inputs and Outputs (Hardware)

Table 3.2: *I/O Memory Map*

Sym.	Description	Range	Bytes
%IX	Digital inputs	1 ... 8	1
%QX	Digital outputs	1 ... 5	1
%IW	Analog inputs	1 ... 4	8
%QW	Analog outputs	1 ... 4	8

NOTE!
The analog input (%IW) and analog output (%QW) values, respectively read and written via the SoftPLC, respect their gains (P0232, P0237, P0242, P0247: %IW1–%IW4 and P0252, P0255, P0258, P0261: %QW1–%QW4) and offsets (P0234, P0239, P0244, P0249: %IW1–%IW4).

NOTE!
The values read or written via SoftPLC obey the following rules, respecting the parameters related to the analog input and output signal types (P0233, P0238, P0243, P0248: %IW1–%IW4 and P0253, P0256, P0259, P0262: %QW1–%QW4):

- Option: 0 to 10 V / 20 mA
  0 V or 0 mA = 0
  10 V or 20 mA = 32767
- Option: 4 to 20 mA
  4 mA = 0
  20 mA = 32767
- Option: 10 V / 20 mA to 0
  10 V or 20 mA = 0
  0 V or 0 mA = 32767
- Option: 20 to 4 mA
  20 mA = 0
  4 mA = 32767
- Option: -10 to +10 V
  -10 V = -32768 (or 32768 for a parameter without sign)
  -5 V = -16384 (or 49152 for a parameter without sign)
  0 = 0
  +10 V = 32767
- Option: 20 to 0 mA
  20 mA = 0
  0 mA = 32767

3.2.3 Volatile Markers (Variables)

They consist of variables that can be applied by the user to execute the applicative logics. They can be bit markers (1 bit), word markers (16 bit) or float markers (32 bit – IEEE).

Table 3.3: *Volatile Marker Memory Map*

Sym.	Description	Range	Quantity of Allocated Bytes
%MX	Bit markers	5000 ... 6099	It depends on the last used marker. They are organized in byte pairs. E.g.: - last marker: %MX5000 = 2 bytes - last marker: %MX5014 = 2 bytes - last marker: %MX5016 = 4 bytes - last marker: %MX5039 = 6 bytes
%MW	Word markers	8000 ... 8199	It depends on the last used marker. E.g.: - last marker: %MX8000 = 2 bytes - last marker: %MX8001 = 4 bytes - last marker: %MX8007 = 16 bytes
%MF	Float markers	9000 ... 9199	It depends on the last used marker. E.g.: - last marker: %MX9000 = 4 bytes - last marker: %MX9001 = 8 bytes - last marker: %MX9007 = 32 bytes

NOTE!
In order to minimize the applicative size, use the markers in a sequential manner.
E.g.:

- Bit markers: %MX5000, %MX5001, %MX5002.
- Word markers: %MW8000, %MW8001, %MW8002.
- Float markers: %MF9000, %MF9001, %MF9002.

3.2.4 System Markers

They consist of special variables that allow the user to read and change converter data that may or may not be available in the parameters. They can be: system bit markers (1 bit) or system word markers (16 bits).

Table 3.4.a: Memory Map for the Odd System Bits

Sym.	Description	Range	Bytes
Type	System bits	3000 ... 3040	4 bytes
%SX	Writing/Command (odd)		
	3001	General Enable	**0:** It disables the firing pulses of armature, interrupting the supply for the motor. **1:** It enables the firing pulses of armature, allowing the motor operation.
	3003	Run	**0:** It stops the motor by deceleration ramp. **1:** It starts the motor according the acceleration ramp up to reach the speed reference.
	3005	Reverse	**0:** It runs the motor in the forward direction. **1:** It runs the motor in the reverse direction.
	3007	JOG	**0:** It disables the JOG function. **1:** It enables the JOG function.
	3009	Remote	**0:** The converter goes to the LOCAL situation. **1:** The converter goes to the REMOTE situation.
	3011	Fault Reset	**0:** No function. **1:** If in a fault condition, it executes the reset of the converter. **NOTE:** When this command is executed the converter and the SoftPLC applicative are reinitialized. This is also valid for the reset via HMI.
	3013	Quick Stop	**0:** It does not execute the quick stop command. **1:** It executes the quick stop command.
	3021	Second Ramp	**0:** The converter uses the first ramp values, programmed in P0100 and P0101, as the motor acceleration and deceleration ramp times. **1:** The converter is configured to use the second ramp values, programmed in P0102 and P0103, as the motor acceleration and deceleration ramp times. **Obs.:** Program P0105 in 5 to enable the selection by SoftPLC.

***Table 3.4.b:** Memory Map for the Even System Bits*

Sym.	Description	Range	Bytes
Type	System bits	3000 ... 3040	4 bytes
%SX	Reading/Status (even)		
	3000	Enabled (READY)	**0:** The converter is not enabled. **1:** The converter is enabled and ready to run the motor.
	3002	Running	**0:** The converter is not sending the programmed reference to the motor. **1:** The converter is running with the programmed reference.
	3004	Reverse	**0:** The rotation is defined to the forward direction. **1:** The rotation is defined to the reverse direction.
	3006	JOG	**0:** The JOG function is not enabled. **1:** The JOG function is enabled.
	3008	Remote	**0:** The converter is in Local mode. **1:** The converter is in Remote mode.
	3010	With Fault	**0:** The converter is not in a fault condition. **1:** The converter has detected a fault. **Obs.**: The fault number can be read by means of the parameter P0049 – Present Fault.
	3012	Undervoltage	**0:** No undervoltage. **1:** With undervoltage.
	3014	FWD/REV Change	**0:** No function. **1:** The converter is changing the direction of rotation.
	3016	With Alarm	**0:** It is not in alarm condition. **1:** It is in alarm condition. **Note:** The alarm number can be read by means of the parameter P0021 - Present Alarm.
	3020	Second Ramp	**0:** It indicates that the second ramp is not active. **1:** It indicates that the second ramp is active.
	3022	Blocked	**0:** The converter is not blocked by logic stop. **1:** The converter is blocked by logic stop.
	3024	In Acceleration	**0:** The converter is not running the accelerating ramp. **1:** The converter is running the accelerating ramp.
	3026	In Deceleration	**0:** The converter is not running the decelerating ramp. **1:** The converter is running the decelerating ramp.
	3028	Self-tuning	**0:** The converter is not running the self-tuning routine. **1:** The converter is running the self-tuning routine.
	3032	Start key (1)	**0:** Not pressed. **1:** Pressed during 1 scan cycle.
	3034	Stop key (0)	
	3036	Rotation Direction key (↻)	
	3038	Local/Remote key	
	3040	JOG key	**0:** Not pressed. **1:** Pressed.

***Table 3.5:** Memory Map for the Word System Markers*

Sym.	Description	Range	Bytes
%SW	System Words	3300 ... 3320	22 bytes
	Reading/Status (Even)		
	3300	Motor Speed [13 bits]	
	3302	Motor Rated Speed [rpm]	
	3304	Motor Speed [rpm]	
	3306	Speed Reference [rpm]	
	3308	Alarm	
	3310	Fault	

NOTE!

The system Word markers %SW3300 and %SW3301 use a 13 bits resolution (0 to 8191), that represents the rated speed of the motor (P0402). This way, for a motor with rated speed of 1800 rpm, if the speed reference by SoftPLC (%SW3301) is set to 4096, the motor will run at 900 rpm.

NOTE!
Equation to calculate the motor speed in rpm:

$$\text{Speed in rpm} = \frac{\text{Rated Speed in rpm x Speed in 13 bits}}{8192}$$

3.2.5 Parameters

Parameters P1001 to P1059 do only appear on the CTW900 HMI when there is a valid applicative (user program) in the memory, i.e., P1000 > 0.

Table 3.6: Memory Map of the Parameters

Sym.	Description	Range	Bytes
%PW	System parameters (refer to the CTW900 User's Manual)	0... 999	
	SoftPLC Parameters	1000 ... 1059	6 bytes
	P1000: SoftPLC Status [Read-only parameter]	0: No Applicative 1: Installing Applicative 2: Incompatible Applicative. 3: Stopped Applicative 4: Running Applicative	
	P1001: SoftPLC Control	0: Stop Applicative 1: Run Applicative 2: Delete Applicative	
	P1002: Scan Cycle Time [ms] [Read-only parameter]		
%UW	User Parameters	1010 ... 1059	100 bytes

3.3 MODBUS

3.3.1 SoftPLC Addresses in the Modbus Protocol

Table 3.7: SoftPLC x Modbus Addresses Range

Sym.	Description	SoftPLC	Modbus
%IX	Digital inputs	1 ... 8	2201...2208
%QX	Digital outputs	1 ... 5	2401...2405
%IW	Analog inputs	1 ... 4	2601...2604
%QW	Analog outputs	1 ... 4	2801...2804

NOTE!
All the other data types have user addresses (SoftPLC) equal to the Modbus addresses. E.g., %PW100 = Modbus address 100; %MX5000 = Modbus address 5000; %SW3308 = Modbus address 3308.

3.3.2 Protocol

Refer to Modbus protocol chapter in the CTW900 Modbus RTU User's Manual.

4 SUMMARY OF THE FUNCTION BLOCKS

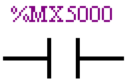
A summary of the function blocks that are available for the user programming will be presented in this chapter.

4.1 CONTACTS

They send to the stack the content of a programmed data (0 or 1), which may be of the type:

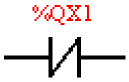
- %MX: Bit Marker.
- %IX: Digital Input.
- %QX: Digital Output.
- %UW: User Parameter.
- %SX: System Bit Marker – Reading.

4.1.1 Normally Open Contact – NO CONTACT



Menu: Insert - Contacts - NO CONTACT.
E.g.: It sends to the stack the content of the bit marker 5000.

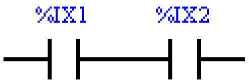
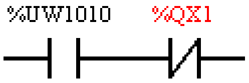
4.1.2 Normally Closed Contact – NC CONTACT



Menu: Insert - Contacts - NC CONTACT.
E.g.: It sends to the stack the negated content of the digital output 1.

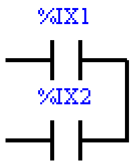
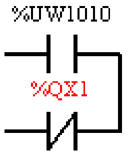
4.1.3 AND Logic with Contacts

When the contacts are in series, an AND logic is executed among them, storing the result in the stack. Examples:

Example	**Truth Table**		
	%IX1	%IX2	Stack
 %IX1.%IX2	0 0 1 1	0 1 0 1	0 0 0 1
	%UW1010	%QX1	Stack
 %UW1010. (~%QX1)	0 0 1 1	0 1 0 1	0 0 1 0

4.1.4 OR Logic with Contacts

When the contacts are in parallel, an OR logic is executed among them, storing the result in the stack. Examples:

Example	Operation	Truth Table		
		%IX1	%IX2	Stack
	%IX1 + %IX2	0	0	0
		0	1	1
		1	0	1
		1	1	1
		%UW1010	%QX1	Stack
	%UW1010 + (~%QX1)	0	0	1
		0	1	0
		1	0	1
		1	1	1

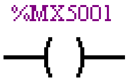
4.2 COILS

They save the stack content (0 or 1) in the programmed element:

- %MX: Bit Marker.
- %QX: Digital Output.
- %UW: User Parameter.
- %SX: System Bit Marker – Writing.

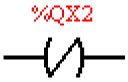
It is allowed to add coils in parallel at the last column.

4.2.1 Normal Coil – COIL



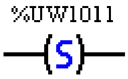
Menu: Insert - Coils - COIL
E.g.: It sets the bit marker 5001 with the stack content.

4.2.2 Negated Coil – NEG COIL



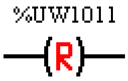
Menu: Insert - Coils - NEG COIL
E.g.: It sets the digital output 2 with the negated content of the stack.

4.2.3 Set Coil – SET COIL



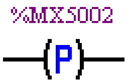
Menu: Insert - Coils - SET COIL
E.g.: It sets the user parameter P1011, provided that the content of the stack is not 0.

4.2.4 Reset Coil – RESET COIL



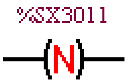
Menu: Insert - Coils - RESET COIL
E.g.: It resets the user parameter P1011, provided that the content of the stack is not 0.

4.2.5 Positive Transition Coil – PTS COIL



Menu: Insert - Coils - PTS COIL
E.g.: It sets the bit marker 5002 during 1 scan cycle, provided that a transition from 0 to 1 in the stack is detected.

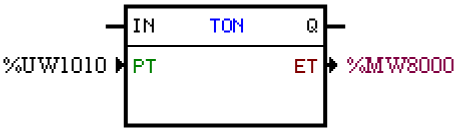
4.2.6 Negative Transition Coil – NTS COIL



Menu: Insert - Coils - NTS COIL
E.g.: It sets the system bit marker 3011 during 1 scan cycle, provided that a transition from 1 to 0 in the stack is detected.

4.3 PLC BLOCKS

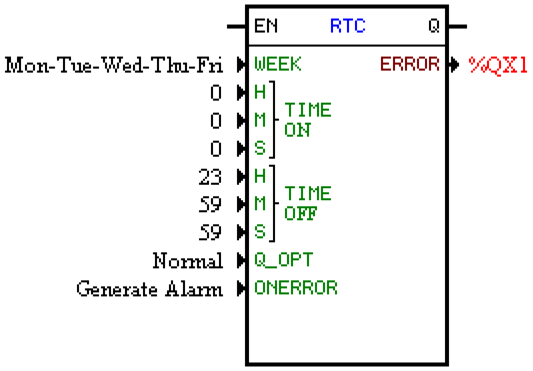
4.3.1 Timer – TON



Menu:	Insert - Function Blocks - PLC - TON
Input:	
IN:	Enables the block
Output:	
Q:	Goes to 1 when IN ≠ 0 and ET ≥ PT
Properties:	
PT:	Programmed Time (Preset Time)
ET:	Elapsed Time

In the example above, if the IN input is active and the content of the word marker 8000 is higher or equal than the content of the user parameter P1010, the output Q is set.

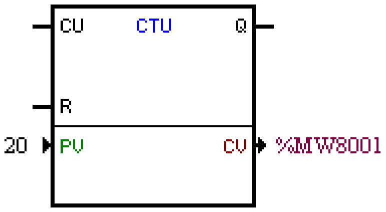
4.3.2 Real Time Clock – RTC



Menu:	Insert - Function Blocks - PLC - RTC
Input:	
EN:	Enables the block
Output:	
Q:	Goes to 1 when IN ≠ 0 and the current time is posterior to the turning on time and anterior to the turning off time.
Properties:	
WEEK:	Days of the week
H-T.ON:	Turning on hour
M-T.ON:	Turning on minute
S-T.ON:	Turning on second
H-T.OFF:	Turning off hour
M-T.OFF:	Turning off minute
S-T.OFF:	Turning off second
Q_OPT:	0: normal Q output, 1: inverted Q output
ONERROR:	0: Generates A700 alarm, 1: Generates F701 fault
ERROR:	It goes to 1 when there is an error in the block.

In the example above, if the EN input is active, the output Q is set every day from Monday through Friday between 7:30 and 9:00.

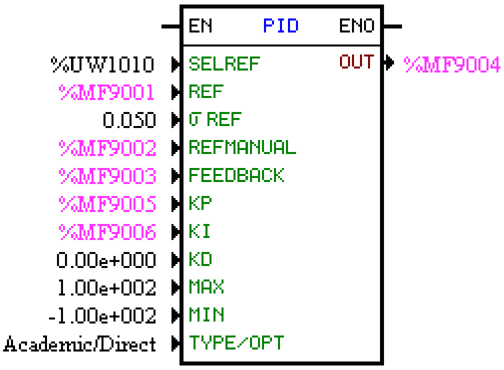
4.3.3 Incremental Counter – CTU



Menu:	Insert - Function Blocks - PLC - CTU
Inputs:	
CU:	Captures the transitions from 0 to 1 at this input (Counter Up)
R:	Resets CV
Output:	
Q:	Goes to 1 when CV ≥ PV
Properties:	
PV:	Programmed Value (Preset Value)
CV:	Counter Value

In the example above, if the content of the word marker 8001 is higher or equal than 20, the output Q is set.

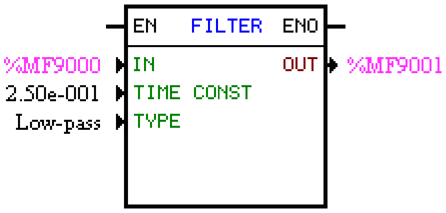
4.3.4 Proportional-Integral-Derivative Controller – PID



Menu:	Insert - Function Blocks - PLC - PID
Inputs:	
EN:	Enables the block
Output:	
ENO:	EN input image
Properties:	
TS:	Sampling Time
SELREF:	Automatic/manual reference
REF:	Automatic reference
δREF:	Automatic reference filter time constant
REFMANUAL:	Manual reference
FEEDBACK:	Process feedback
KP:	Proportional gain
KI:	Integral gain
KD:	Derivative gain
MAX:	Maximum output value
MIN:	Minimum output value
TYPE:	Academic/parallel
OPT:	Direct/reverse
OUT:	Controller output

In the example above, if the EN input is active, the controller starts its operation. The content of the user parameter P1010 selects the reference that is active, i.e., whether it is the float marker 9001 (automatic reference) or the 9003 (manual reference). There is a 0.05s filter for the automatic reference. Since the derivative gain is fixed in 0, this indicates that the PID was converted into a PI. The control output OUT, represented by the float marker 9004, has the maximum and minimum limits of 100 and -100.

4.3.5 Low-Pass or High-Pass Filter – FILTER

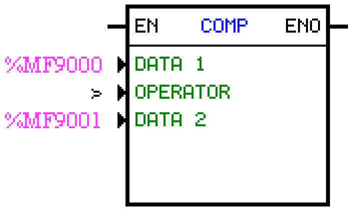


Menu:	Insert - Function Blocks - PLC - FILTER
Inputs:	
EN:	Enables the block
Output:	
ENO:	EN input image
Properties:	
TS:	Sampling Time
IN:	Input data
TIMECONST:	Filter time constant
TYPE:	Low-pass/High-pass
OUT:	Input data filtered value

In the example above, if the EN input is active, the content of the float marker 9000 will be filtered with a time constant of 0.25s by means of a low-pass filter and will be transferred to the float marker 9001.

4.4 CALCULATION BLOCKS

4.4.1 Comparator – COMP

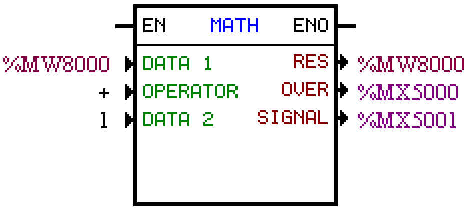


Menu: Insert - Function Blocks – Calculation - COMP
Input:
EN: Enables the block
Output:
ENO: Goes to 1 when the comparison condition is fulfilled
Properties:
FORMAT: Integer or floating point
DATA 1: Comparison data 1
OPERATOR: Comparison operator
DATA 2: Comparison data 2

In the example above, if the EN input is active and the content of the float marker 9000 is higher than the content of the float marker 9001, then the output ENO is set.

NOTE!
If the FORMAT is integer, all the numeric data are considered words of 15 bits + sign (-32768 to 32767).

4.4.2 Mathematical Operation – MATH

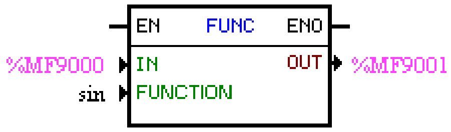


Menu: Insert - Function Blocks - Calculation - MATH
Input:
EN: Enables the block
Output:
ENO: Indicates whether the calculation has been executed
Properties:
FORMAT: Integer or floating point
DATA1: Calculation data 1. It may also appear as DATA1H and DATA1L (representing the high and low parts of the data 1)
OPERATOR: Mathematic operator (+, -, *, etc)
DATA2: Calculation data 2. It may also appear as DATA2H and DATA2L (representing the high and low parts of the data 2)
RES: Calculation result. It may also appear as RESH and RESL (representing the high and low parts of the result) and also as QUOC and REM (representing the quotient and the reminder of a division)
OVER: Indicates whether the result exceeded its limit
SIGNAL: Result sign

In the example above, if the EN input is active, the value of the word marker 8000 is incremented at each scan cycle. When the bit marker 5000 goes to 1, it indicates overflow and the word marker 8000 remains in 32767.

NOTE!
If the FORMAT is integer, all the numeric data are considered words of 15 bits + sign (-32768 to 32767).

4.4.3 Mathematical Function – FUNC

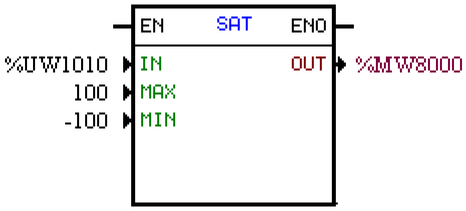


Menu: Insert - Function Blocks - Calculation - FUNC
Input:
EN: Enables the block
Output:
ENO: Indicates whether the calculation has been executed
Properties:
FORMAT: Integer or floating point
IN: Data to be calculated
FUNCTION: Mathematic function (sin, cos, etc)
OUT: Calculation result

In the example above, if the EN input is active, the float marker 9001 presents the result of the float marker 9000 sine calculation.

NOTE!
If the FORMAT is integer, all the numeric data are considered words of 15 bits + sign (-32768 to 32767).

4.4.4 Saturator – SAT



Menu: Insert - Function Blocks - Calculation - SAT
Input:
EN: Enables the block
Output:
ENO: Indicates whether saturation has occurred, provided that EN ≠ 0
Properties:
FORMAT: Integer or floating point
IN: Input data
MAX: Maximum allowed value
MIN: Minimum allowed value
OUT: Output data

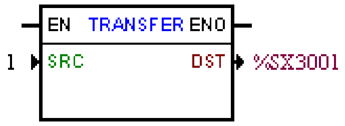
In the example above, when the EN input is active, the word marker 8000 contains the user parameter P1010 value, limited however, between the maximum of 100 and the minimum of -100.

NOTE!
If the FORMAT is integer, all the numeric data are considered words of 15 bits + sign (-32768 to 32767).

NOTE!
If the MIN value is higher than the MAX value, the outputs OUT and ENO are zeroed.

4.5 TRANSFER BLOCKS

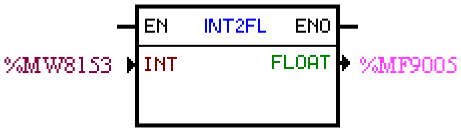
4.5.1 Data Transfer – TRANSFER



Menu: Insert - Function Blocks - Transfer - TRANSFER
Input:
EN: Enables the block
Output:
ENO: Indicates that the transfer has been done
Properties:
SRC: Source data
DST: Destination data

In the example above, if the EN input is active, the word constant 1 is transferred to the system bit marker 3001 (general enable).

4.5.2 Conversion from Integer (16 bit) to Floating Point – INT2FL

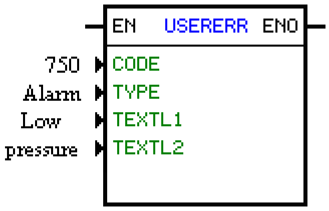


Menu: Insert - Function Blocks - Transfer - INT2FL
Input:
EN: Enables the block
Output:
ENO: Indicates that the transfer has been done
Properties:
INT: Integer data
FLOAT: Data converted into floating point

In the example above, if the EN input is active, the content of the word marker 8153 (taking into account its sign) is converted into floating point to the float marker 9005.

NOTE!
INT is treated as a word of 15 bits + sign (-32768 to 32767).

4.5.3 User Fault or Alarm Generator – USERERR

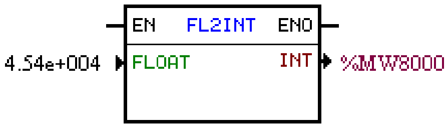


Menu: Insert - Function Blocks - Transfer - USERERR
Input:
EN: Enables the block
Output:
ENO: Indicates 1 when EN = 1 and the alarm or the fault was actually generated.
Properties:
CODE: Alarm or fault code
TYPE: 0: Generates alarm, 1: Generates fault
TEXTL1: HMI line 1 text
TEXTL2: HMI line 2 text

In the example above, if the EN input is active, A750 will appear with the “Low pressure” text on the HMI.

NOTE!
If this block is configured as Fault, it is necessary to reset the converter to remove the text from the HMI and be able to enable it again.

4.5.4 Conversion from Floating Point to Integer (16 bit) – FL2INT

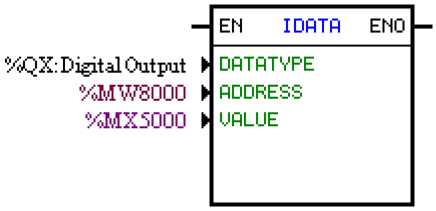


Menu:	Insert - Function Blocks - Transfer - FL2INT
Input:	
EN:	Enables the block
Output:	
ENO:	Indicates that the transfer has been done
Properties:	
FLOAT:	Floating point data
INT:	Data converted into integer

In the example above, if the EN input is active, the float constant 4.54 x 10^4 is converted into an integer with sign via the word marker 8000. However, after the conversion, the word marker 8000 will remain with the value of 32767, because this is the positive limit of a word.

NOTE!
INT is treated as a word of 15 bits + sign (-32768 to 32767).

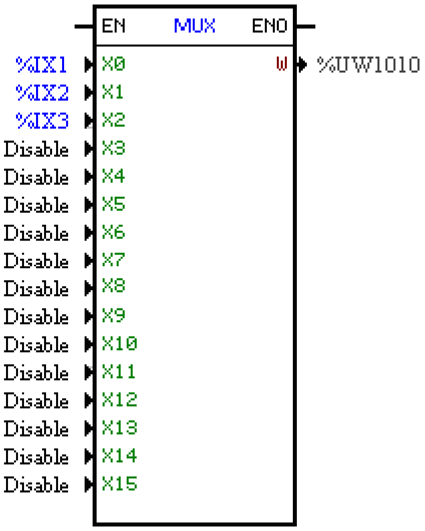
4.5.5 Indirect Data Transfer – IDATA



Menu:	Insert - Function Blocks - Transfer - IDATA.
Input:	
EN:	Enables the block.
Output:	
ENO:	Indicates that the transfer has been done.
Properties:	
CMD:	Read/Write command
DATATYPE:	Data type
ADDRESS:	User address
VALUE:	Read content/Value to be written

In the example above, if the EN input is active, the content of the bit marker 5000 is written to the digital output whose address is the content of the word marker 8000.

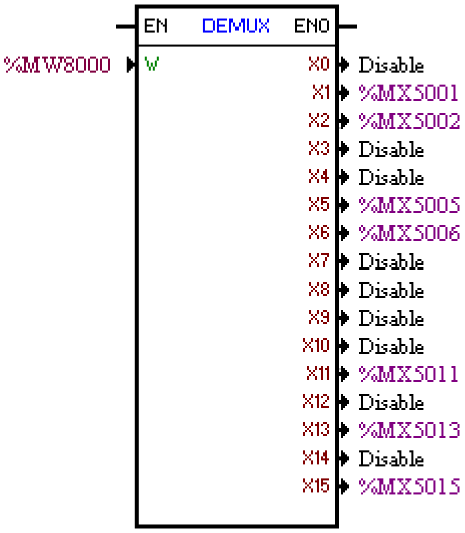
4.5.6 Multiplexer – MUX



Menu:	Insert - Function Blocks - Transfer - MUX
Input:	
EN:	Enables the mathematic operation
Output:	
ENO:	Indicates that the transfer has been done
Properties:	
X0-X15:	Binary data vector
W:	Resulting word

In the example above, when the EN input is active, the digital inputs 1, 2 and 3 transfer their content to the bits 0, 1 and 2 of the user parameter P1010.

4.5.7 Demultiplexer – DMUX



Menu: Insert - Function Blocks - Transfer - DMUX

Input:

EN: Enables the mathematic operation

Output:

ENO: Indicates that the transfer has been done

Properties:

W: Source word

X0-X15: Resulting binary data vector

In the example above, when the EN input is active, the bits 1, 2, 5, 6, 11, 13 and 15 of the word marker 8000 are transferred respectively to the bit markers 5001, 5002, 5005, 5006, 5011, 5013 and 5015.

5 CTW900 PARAMETER SETTINGS

Next, only the CTW900 parameters related to the SoftPLC will be presented.

5.1 SIMBOLS FOR PROPERTIES DESCRIPTION

RO:	Read-only parameter.
CFG1:	Configuration parameter level 1: modification not allowed with converter running.

5.2 CTW900 CONFIGURATION PARAMETERS

P0100 – Acceleration Ramp

P0101 – Deceleration Ramp

P0220 – LOCAL/REMOTE Mode Selection

P0221 – Speed Reference Selection in LOCAL Mode

P0222 – Speed Reference Selection in REMOTE Mode

P0223 – Forward/Reverse Selection in LOCAL Mode

P0226 – Forward/Reverse Selection in REMOTE Mode

P0224 – Run/Stop Selection in LOCAL Mode

P0227 – Run/Stop Selection in REMOTE Mode

P0225 – JOG Selection in LOCAL Mode

P0228 – JOG Selection in REMOTE Mode

P0251 – AO1 Function

P0254 – AO2 Function

P0257– AO3 Function

P0260 – AO4 Function

P0275 – DO1 Function

P0276 – DO2 Function

P0277 – DO3 Function

P0278 – DO4 Function

P0279 – DO5 Function

P0560 – Trace Available Memory

✓ **NOTE!**
For further information, refer to the CTW900 User’s Manual.

5.3 SOFTPLC EXCLUSIVE PARAMETERS

P1000 – SoftPLC Status

Adjustable Range: 0 = No Applicative
1 = Installing Applicative
2 = Incompatible Applicative
3 = Stopped Applicative
4 = Running Applicative

Factory Setting: -

Properties: RO

Access groups via HMI: 01 PARAMETER GROUPS
∟ 40 SoftPLC

Description:
It allows the user to visualize the SoftPLC status. If there is no installed applicative, the parameters from P1001 to P1059 will not be showed on the HMI.

If this parameter presents the option 2 (Incompatible App.), it indicates that the version that has been loaded in the flash memory board is not compatible with the current CTW900 firmware.

In this case, it is necessary to recompile the project in the WLP considering the new CTW900 version, and to download it again. If this is not possible, the upload of this applicative with the WLP can be done, provided that the applicative password is known or that the password is not enabled.

P1001 – SoftPLC Control

Adjustable Range: 0 = Stop Applicative
1 = Run Applicative
2 = Delete Applicative

Factory Setting: 0

Properties: CFG1

Access groups via HMI: 01 PARAMETER GROUPS
∟ 40 SoftPLC

Description:
It allows stopping, running or deleting an installed applicative. Therefore, the converter cannot be Running.

P1002 – Scan Cycle Time

Adjustable Range: 0 to 65535 ms

Factory Setting: -

Properties: RO

Access groups via HMI: 01 PARAMETER GROUPS
∟ 40 SoftPLC

Description:
It consists in the applicative scanning time. The bigger the applicative, the longer the scanning time will be.

P1010 to P1059 – SoftPLC Parameters

Adjustable Range: -32768 to 32767 **Factory Setting:** 0

Properties: CFG1

Access groups via HMI: 01 PARAMETER GROUPS
∟ 40 SoftPLC

Description:
They consist of parameters with functions defined by the user by means of the WLP software. It is also possible for the user to configure these parameters as described in the section 6.5.

NOTE!
Parameters P1010 to P1019 can be visualized in the monitoring mode.

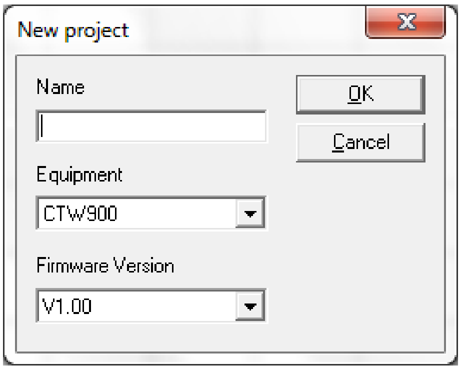
NOTE!
When P1011 is a writing parameter and it is programmed in P0205, P0206 or P0207, then its content can be changed with the HMI ▲ and ▼ keys.

6 SUMMARY OF THE WLP MAIN FUNCTIONS

This chapter brings basic information on the operations performed with the WLP software for the CTW900 programming. Further information can be obtained in the WLP software help.

6.1 PROJECT – NEW

It creates a new project. Besides defining the project name, it is also necessary to configure the equipment and the respective firmware version.



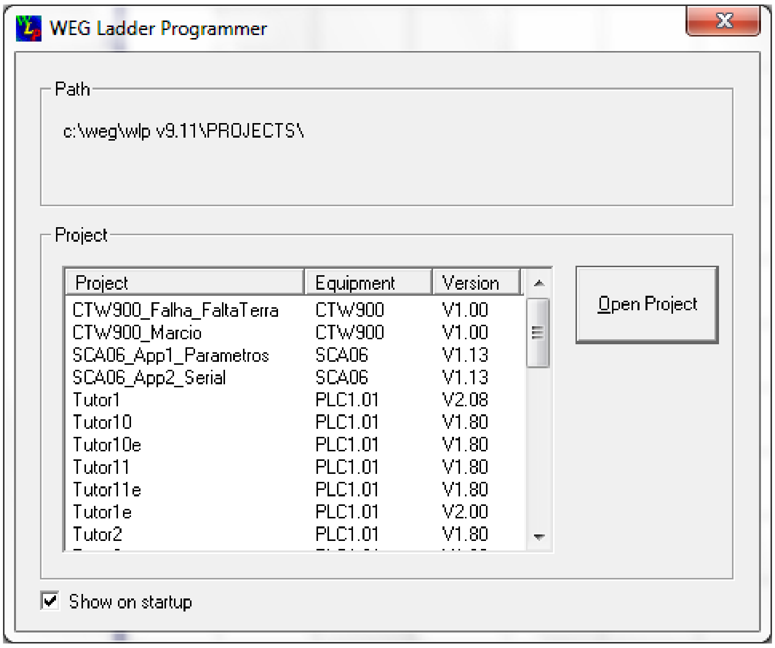
Name: |

Equipment: CTW900

Firmware Version: V1.00

6.2 PROJECT – OPEN

It opens the selected project.



Path

c:\weg\wlp v9.11\PROJECTS\

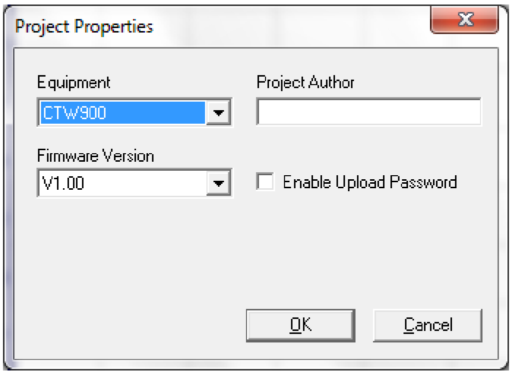
Project

Project	Equipment	Version
CTW900_Falha_FaltaTerra	CTW900	V1.00
CTW900_Marcio	CTW900	V1.00
SCA06_App1_Parametros	SCA06	V1.13
SCA06_App2_Serial	SCA06	V1.13
Tutor1	PLC1.01	V2.08
Tutor10	PLC1.01	V1.80
Tutor10e	PLC1.01	V1.80
Tutor11	PLC1.01	V1.80
Tutor11e	PLC1.01	V1.80
Tutor1e	PLC1.01	V2.00
Tutor2	PLC1.01	V1.80

☑ Show on startup

6.3 PROJECT – PROPERTIES

It allows the user to redefine the equipment and the firmware version. In this box it is also configured whether the project will have an upload password.



Equipment: CTW900

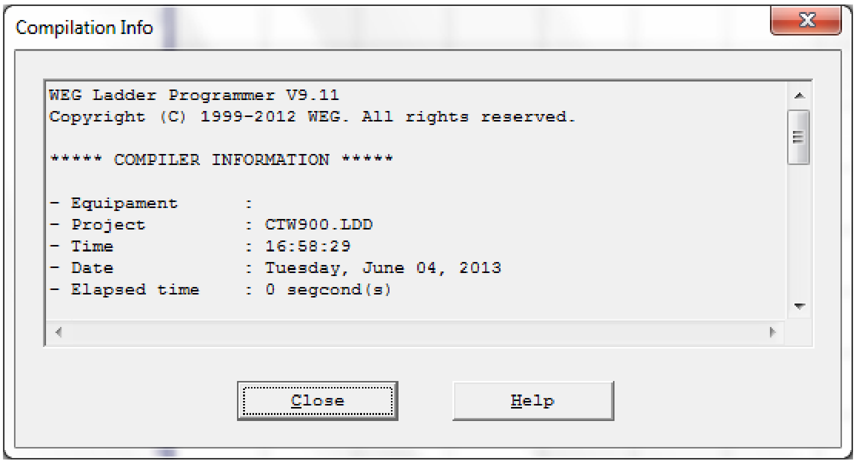
Project Author:

Firmware Version: V1.00

□ Enable Upload Password

6.4 VIEW – COMPILATION INFORMATION

It allows the user to know the size in bytes of the compiled applicative (<projectname>.bin) to be sent to the equipment.



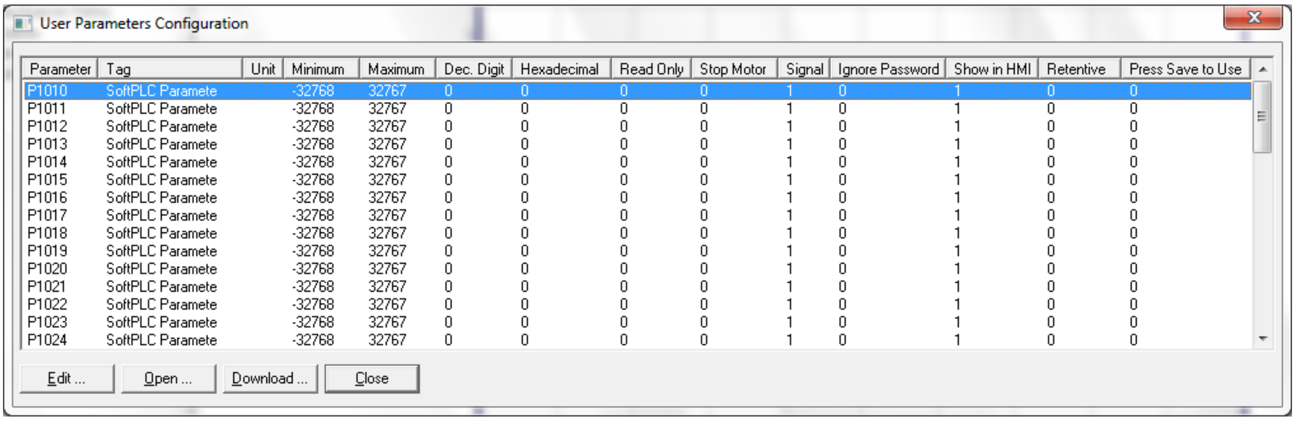
```
WEG Ladder Programmer V9.11
Copyright (C) 1999-2012 WEG. All rights reserved.

***** COMPILER INFORMATION *****

- Equipament      :
- Project         : CTW900.LDD
- Time            : 16:58:29
- Date            : Tuesday, June 04, 2013
- Elapsed time    : 0 segcond(s)
```

6.5 VIEW – USER PARAMETER CONFIGURATION

It opens an attribute visualization window for all the user parameters. With a double click on the parameter, it is possible to configure these attributes, which include:

- Parameter descriptive text on the HMI (up to 21 characters).
- Text for the units (up to 5 characters).
- Maximum and minimum limits.
- Number of decimal positions.
- Hexadecimal or decimal format.
- Writing or read-only.
- Modifications only with a stopped motor, or online.
- With or without sign.
- Ignores the password (allows modification regardless of P0000) or normal.
- Visualizes or hides the parameter.
- Allows saving the parameter value on power down (retentive), when it is used in some blocks (PLC, Calculation and Transfer).
- Configuration parameter that allows modification with the motor running.

These configurations can be transferred to the CTW900 with the “Download key”.

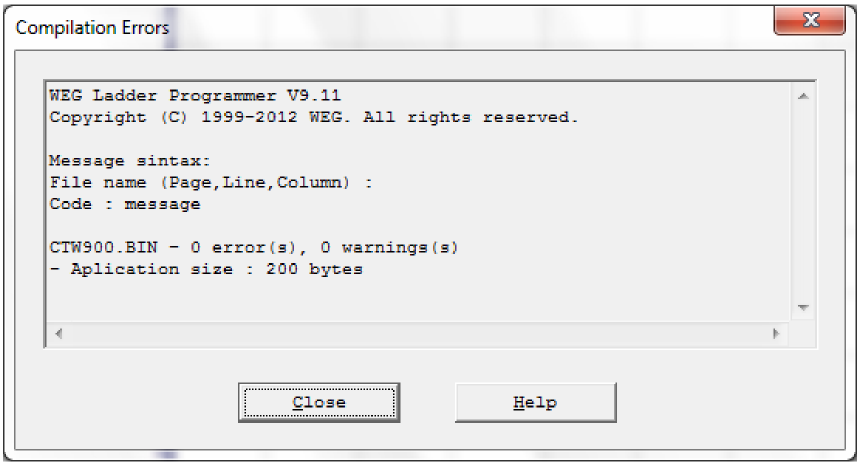


User Parameters Configuration

Parameter	Tag	Unit	Minimum	Maximum	Dec. Digit	Hexadecimal	Read Only	Stop Motor	Signal	Ignore Password	Show in HMI	Retentive	Press Save to Use
P1010	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1011	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1012	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1013	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1014	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1015	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1016	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1017	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1018	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1019	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1020	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1021	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1022	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1023	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0
P1024	SoftPLC Paramete		-32768	32767	0	0	0	0	1	0	1	0	0

6.6 CONSTRUCT – COMPILE

It analyses the applicative and generates the code for the specified equipment.

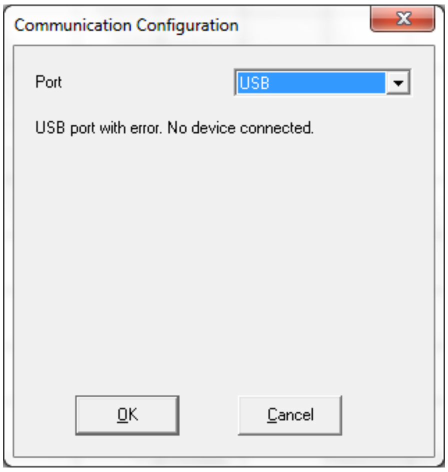


Compilation Errors

```
WEG Ladder Programmer V9.11
Copyright (C) 1999-2012 WEG. All rights reserved.

Message sintax:
File name (Page,Line,Column) :
Code : message

CTW900.BIN - 0 error(s), 0 warnings(s)
- Aplication size : 200 bytes
```

6.7 COMMUNICATION – CONFIGURATION

The USB port is used with the CTW900. Therefore, the USB driver must be installed. The driver is found in the DRIVER_USB folder, inside the WLP V8.XX.



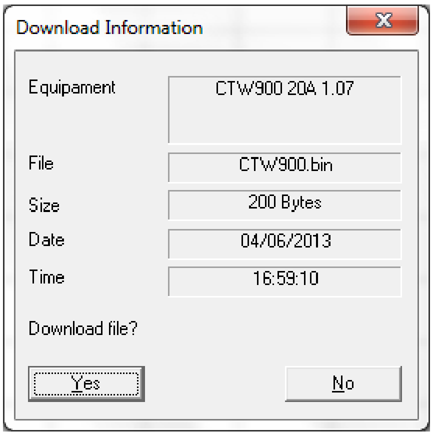
Communication Configuration

Port: USB

USB port with error. No device connected.

6.8 COMMUNICATION – DOWNLOAD

This command allows downloading the applicative and/or the user parameter configurations to the CTW900.



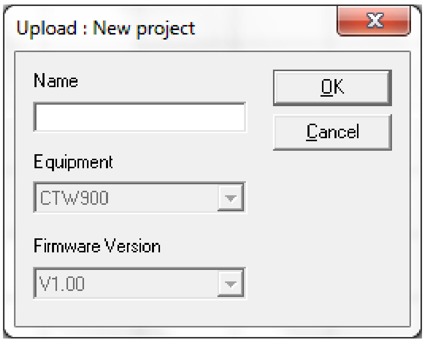
Download Information

Equipament	CTW900 20A 1.07
File	CTW900.bin
Size	200 Bytes
Date	04/06/2013
Time	16:59:10

Download file?

6.9 COMMUNICATION – UPLOAD

This command makes it possible to upload and open the applicative that is installed in the CTW900, provided that the password is valid.



Upload : New project

Name

Equipment: CTW900

Firmware Version: V1.00

7 FAULTS, ALARMS AND POSSIBLE CAUSES

Table 7.1: *Faults, alarms and possible causes*

Fault/Alarm	Description	Possible Causes
A700: Disconnected HMI	It is the alarm that occurs when the RTC block is active, the “Alarm” option is programmed and the HMI is not connected to the converter.	Verify if the HMI is connected to the converter. Search for unplugged connector, broken cable etc.
F701: Disconnected HMI	It is the fault that occurs when the RTC block is active, the “Fault” option is programmed and the HMI is not connected to the converter.	Verify if the HMI is connected to the converter. Search for unplugged connector, broken cable etc.
A702: Converter Disabled	It occurs when the movement block (REF block) and the general enabling command of the converter is not active.	Verify if the general enabling command of the converter is active.
A704: Two Movem. Enabled	It occurs when 2 or more movement blocks (REF block) are enabled simultaneously.	Verify the user program logic.
A706: Not Program. Refer. SPLC	It occurs when a movement block is enabled and the speed reference is not programmed for the SoftPLC.	Verify the programming of the reference in the local and/or remote modes (P0220 and P0221).

WEG Drives & Controls - Automação LTDA.
Jaraguá do Sul - SC – Brazil
Phone 55 (47) 3276-4000 - Fax 55 (47) 3276-4020
São Paulo - SP – Brazil
Phone 55 (11) 5053-2300 - Fax 55 (11) 5052-4212
automacao@weg.net
www.weg.net