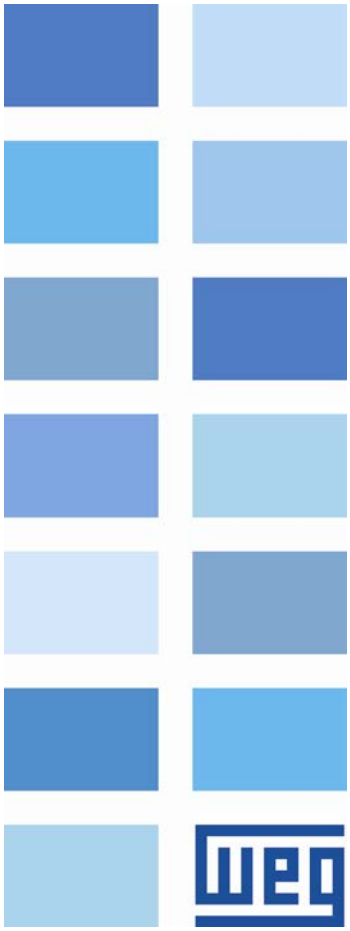
SoftPLC

CFW700

User's Manual

Language: English



SoftPLC Manual

Series: CFW700

Language: English

Document Number: 10001124052 / 02

Publication Date: 10/2013

CONTENTS

ABOUT THIS MANUAL 5
ABBREVIATIONS AND DEFINITIONS 5
NUMERICAL REPRESENTATION 5
1 INTRODUCTION TO THE SOFTPLC 6
1.1 SYMBOLS AND DATA TYPES 6
2 SOFTPLC MEMORY 7
2.2 DATA MEMORY 7
2.2.1 Constants 7
2.2.2 Physical Inputs and Outputs (Hardware) 7
2.2.3 Volatile Markers (Variables) 8
2.2.4 System Markers 8
2.2.5 Parameters 10
2.3 MODBUS 11
2.3.1 Modbus protocol SoftPLC addresses 11
2.3.2 Protocol 11
3 RESUME OF THE FUNCTION BLOCKS 12
3.1 CONTACTS 12
3.1.1 Normally Open Contact – NO CONTACT 12
3.1.2 Normally Closed Contact – NC CONTACT 12
3.1.3 AND Logic with Contacts 12
3.1.4 OR Logic with Contacts 12
3.2 COILS 13
3.2.1 Normal Coil – COIL 13
3.2.2 Negated Coil – NEG COIL 13
3.2.3 Set Coil – SET COIL 13
3.2.4 Reset Coil – RESET COIL 13
3.2.5 Positive Transition Coil – PTS COIL 13
3.2.6 Negative Transition Coil – NTS COIL 13
3.3 MOVEMENT BLOCKS 14
3.3.1 Speed and/or Torque Reference – REF 14
3.4 POSITIONING BLOCKS 14
3.4.1 Stop in Position – POSITION0 14
3.5 CLP BLOCKS 14
3.5.1 Timer – TON 14
3.5.2 Incremental Counter– CTU 15
3.5.3 Proportional-Integral-Derivative Controller – PID 15
3.5.4 Low-pass or High-pass Filter – FILTER 15
3.6 CALCULATION BLOCKS 16
3.6.1 Comparator – COMP 16
3.6.2 Math Operation – MATH 16
3.6.3 Math Function – FUNC 17
3.6.4 Saturator – SAT 17
3.7 TRANSFER BLOCKS 18
3.7.1 Data Transfer – TRANSFER 18
3.7.2 Conversion from Integer (16 bit) to Floating Point – INT2FL 18
3.7.3 User Fault or Alarm Generator – USERERR 18
3.7.4 Converts from Floating Point to Integer (16 bit) – FL2INT 19
3.7.5 Indirect Data Transfer– IDATA 19
3.7.6 Multiplexer – MUX 19
3.7.7 Demultiplexer – DMUX 20

4 INVERTER PARAMETER SETTINGS 21

4.1 SYMBOLS FOR THE PROPERTIES DESCRIPTION 21
4.2 CFW700 CONFIGURATION PARAMETERS 21
4.3 SOFTPLC EXCLUSIVE PARAMETERS 22
P1000 – SOFTPLC STATUS 22
P1001 – SOFTPLC COMMAND 22
P1002 – SCAN CYCLE TIME 22
P1003 – SOFTPLC APPLICATIVE SOFTWARE 22
P1010 TO P1059 – SOFTPLC PARAMETERS 23

5 RESUME OF THE WLP MAIN FUNCTIONS 24

5.1 PROJECT – NEW 24
5.2 PROJECT – OPEN 24
5.3 PROJECT – PROPERTIES 24
5.4 VIEW – COMPILATION INFO 25
5.5 VIEW – USER PARAMETER CONFIGURATION 25
5.6 CONSTRUCT – COMPILE 26
5.7 COMMUNICATION – CONFIGURATION 26
5.8 COMMUNICATION – DOWNLOAD 27

6 FAULTS, ALARMS, AND POSSIBLE CAUSES 28

ABOUT THIS MANUAL

This manual provides the necessary description for the operation of the CFW700 frequency inverter using the user programming module denominated SoftPLC. This manual must be used together with the CFW700 user manual and with the WLP software manual.

ABBREVIATIONS AND DEFINITIONS

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

NUMERICAL REPRESENTATION

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

COMPATIBILITY

NOTE!
Use the WLP V9.50 or higher for SoftPLC programs in firmware version V2.01.

NOTE!
SoftPLC programs from firmware version V2.01 are incompatible with programs from previous firmware versions.

1 INTRODUCTION TO THE SOFTPLC

The SoftPLC is a feature that incorporates to the CFW700 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 CFW700 I/O's and parameters.
- 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.

1.1 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)

2 SOFTPLC MEMORY

2.1 MEMORY DIVISION

- RAM SoftPLC: 4096 bytes
- FLASH SoftPLC: 32768 bytes

NOTE!
The SoftPLC applicative software stored in the FLASH memory runs in the RAM (Random Access Memory). Therefore, whenever the applicative is larger than 4536 bytes, the scan cycle slows down due to loading time from the FLASH memory to the RAM.

2.2 DATA MEMORY

The SoftPLC data memory area (user variables) is shared with the programming memory. Therefore, the total size of an applicative may vary as function of the amount 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 program memory space.

2.2.1 Constants

Table 2.1: *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

2.2.2 Physical Inputs and Outputs (Hardware)

Table 2.2: *I/O Memory Map*

Sym.	Description	Range	Bytes
%IX	Digital inputs	1 ... 8	2
%QX	Digital outputs	1 ... 5	2
%IW	Analog inputs	1 ... 2	4
%QW	Analog outputs	1 ... 2	4

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–%QW2) and offsets (P0234, P0239, P0244, P0249: %IW1–%IW2).

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–%QW2):

- 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

2.2.3 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 2.3: *Volatile Marker Memory Map*

Sym.	Description	Range	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...

2.2.4 System Markers

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

Table 2.4.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 Enabling		**0:** It disables the inverter, interrupting the supply for the motor. **1:** It enables the inverter allowing the motor operation.
	3003	Run/Stop		**0:** It stops the motor with deceleration ramp. **1:** The motor runs according to the acceleration ramp until reaching the speed reference value.
	3005	Speed Direction		**0:** It runs the motor in the counterclockwise direction. **1:** It runs the motor in the clockwise direction.
	3007	JOG		**0:** It disables the JOG function. **1:** It enables the JOG function.
	3009	LOC/REM		**0:** The inverter goes to the LOCAL situation. **1:** The inverter goes to the REMOTE situation.
	3011	Fault reset		**0:** No function. **1:** If in a fault condition, then it executes the inverter reset. **Note:** When this command is executed the inverter and the SoftPLC applicative are reinitialized. This is also valid for the reset via keypad.
	3021	Activates the Second Ramp		**0:** The values for the motor acceleration and deceleration are those from the first ramp (P0100 and P0101). **1:** The values for the motor acceleration and deceleration are those from the second ramp (P0102 and P0103). **Note:** In order to enable the selection via SoftPLC, program P0105 in 6.

Table 2.4.b: *Memory Map for the Even System Bits*

Sym.	Description		Range	Bytes
Type	System bits		3000 ... 3040	4 bytes
%SX	*Reading/State (Even)*			
	3000	General Enabling		**0:** General Enabling is not active **1:** General enabling is active and the inverter is ready to run the motor.
	3002	Motor Running (RUN)		**0:** The motor is stopped **1:** The inverter is driving the motor at the set point speed, or executing either the acceleration or the deceleration ramp.
	3004	Speed Direction		**0:** The motor is rotating counterclockwise **1:** The motor is rotating clockwise
	3006	JOG		**0:** JOG function inactive **1:** JOG function active
	3008	LOC/REM		**0:** Inverter in LOCAL situation **1:** Inverter in REMOTE situation
	3010	Fault condition		**0:** The inverter is not in a fault condition **1:** Any fault has been registered by the inverter. **Note:** The fault number can be read by means of the parameter P0049 – Current Fault.
	3012	Undervoltage		**0:** No Undervoltage **1:** With Undervoltage
	3014	PID operation mode		**0:** In manual mode (PID function) **1:** In automatic mode (PID function)
	3016	Alarm condition		**0:** The inverter is not in an alarm condition **1:** The inverter is in an alarm condition. **Note:** The alarm number can be read by means of the parameter P0048 – Current Alarm.
	3018	In configuration mode		**0:** Inverter operating normally. **1:** Inverter in configuration mode. It indicates a special condition when the inverter cannot be enabled: ▪ Executing the self-tuning routine. ▪ Executing guided start-up routine. ▪ Executing the keypad copy function. ▪ Executing the flash memory card guided routine. ▪ There is a parameter setting incompatibility. **Note:** It is possible to obtain the exact description of the special operation mode at parameter P0692.
	3020	Active Ramp		**0:** Indicates that the first ramp is active **1:** indicates that the second ramp is active
	3032	Start key (1)		**0:** Not pressed **1:** Pressed during 1 scan cycle
	3034	Stop key (0)		
	3036	Speed direction key (↻)		
	3038	Local/Remote key		
	3040	JOG key		**0:** Not pressed **1:** Pressed

Table 2.5: Memory Map for the System Word Markers

Sym.	Description		Range	Bytes
%SW	System Words		3300 ... 3324	22 bytes
	Reading markers/Status (Even)			
	3300	Motor speed [13 bits]		
	3302	Motor synchronous speed [rpm]		
	3304	Motor speed [rpm]		
	3306	Speed reference [rpm]		
	3308	Alarm		
	3310	Fault		
	3312	Flux Current Id [13 bit]		
	3314	Torque Current Iq [13 bit]		
	3316	Flux Current Reference Id* [13 bit]		
	3318	Torque Current Reference Iq* [13 bit]		
	3320	Inverter Nominal Current (HD) [A x10]		
	3322	Unfiltered motor current (P003) [A x10]		
	3324	Unfiltered motor torque [% x10]		

NOTE!
The system word markers %SW3300 and %SW3301 use a 13 bits resolution (8192 → 0 to 8191), which represents the motor synchronous speed. Thus, if for a VI pole motor (this means a synchronous speed of 1200 rpm) the speed reference via SoftPLC (%SW3301) is 4096; the motor will run at 600 rpm.

NOTE!
Equation for the calculation of the motor speed value in rpm:

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

2.2.5 Parameters

The parameters from P1011 to P1059 appear on the keypad only when there is a valid applicative (user program) in the memory, i.e., when P1000 > 0.

Table 2.6: Parameter Memory Map

Sym.	Description	Range	Bytes
%PW	System parameters (refer to the CFW700 manual)	0... 999	
	SoftPLC parameters	P1000...P 1003	6 bytes
	P1000: SoftPLC status (Read-only parameter)	0: No Applicative 1: Install. App. 2: Incompat. App. 3: App. Stopped 4: App. Running	
	P1001: SoftPLC Command	0: Stop Program 1: Run Program 2: Delete Program	
	P1002: Scan Cycle Time [ms] (Read-only parameter)		
	P1003: SoftPLC applicative	0: User 1: PID Regulator 2: Electronic Potentiometer (EP) 3: Multispeed 4: 3-Wire Start/Stop 5: FWD/REV	
%UW	User parameters	P1010...P1059	100 bytes

2.3 MODBUS

2.3.1 Modbus protocol SoftPLC addresses

Table 2.7: SoftPLC x Modbus address range

Sym.	Description	SoftPLC	Modbus
%IX	Digital inputs	1 ... 8	2201...2208
%QX	Digital outputs	1 ... 5	2401...2405
%IW	Analog inputs	1 ... 2	2601...2602
%QW	Analog outputs	1 ... 2	2801...2802

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

2.3.2 Protocol

Refer to the RS232/RS485 Serial Communication Manual, at the Modbus protocol chapter.

3 RESUME OF THE FUNCTION BLOCKS

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

3.1 CONTACTS

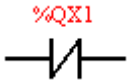
They send to the stack the content of a programmed data (0 or 1), which may be of the type:

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

3.1.1 Normally Open Contact – NO CONTACT

Menu: *Insert – Contacts – Normally Open Contact.*
E.g.: It sends to the stack the content of the bit marker 5000.

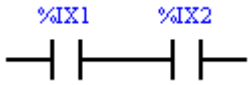
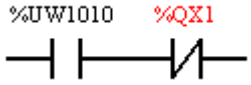
3.1.2 Normally Closed Contact – NC CONTACT



Menu: *Insert – Contacts – Normally Closed Contact.*
E.g.: It sends to the stack the negated content of the digital output 1.

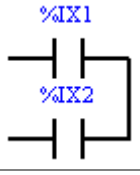
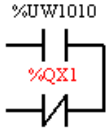
3.1.3 AND Logic with Contacts

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	%IX1	%IX2	Stack
	0 0 1 1	0 1 0 1	0 0 0 1
 %UW1010. (~%QX1)	%UW1010	%QX1	Stack
	0 0 1 1	0 1 0 1	0 0 1 0

3.1.4 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	%IX1	%IX2	Stack
		0 0 1 1	0 1 0 1	0 1 1 1
	%UW1010 + (~%QX1)	%UW1010	%QX1	Stack
		0 0 1 1	0 1 0 1	1 0 1 1

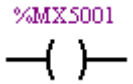
3.2 COILS

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

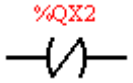
It is allowed to add coils in parallel at the last column.

3.2.1 Normal Coil – COIL



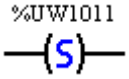
Menu: *Insert – Coils – Coil.*
E.g.: It sets the bit marker 5001 with the stack content.

3.2.2 Negated Coil – NEG COIL



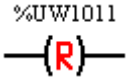
Menu: *Insert – Coils – Negated Coil.*
E.g.: It sets the digital output 2 with the negated content of the stack.

3.2.3 Set Coil – SET COIL



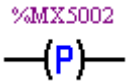
Menu: *Insert – Coils – Set Coil.*
E.g.: It sets the user parameter P1011, provided that the content of the stack is not 0.

3.2.4 Reset Coil – RESET COIL



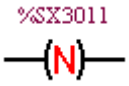
Menu: *Insert – Coils – Reset Coil.*
E.g.: It resets the user parameter P1011, provided that the content of the stack is not 0.

3.2.5 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.

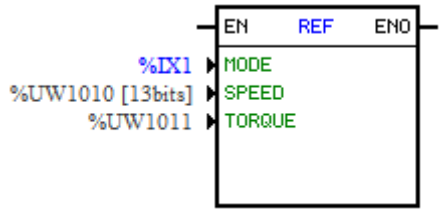
3.2.6 Negative Transition Coil – NTS COIL



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.

3.3 MOVEMENT BLOCKS

3.3.1 Speed and/or Torque Reference – REF

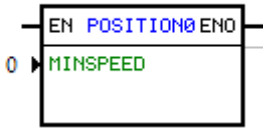


Menu: *Insert - Function Blocks - Movement - REF*
Input:
EN: Enables the block
Output:
ENO: Goes to 1 when EN ≠ 0 and without error.
Properties:
MODE: 0 = Speed mode, 1= Torque mode
SPEED: Speed reference [RPM or13 Bits]
TORQUE: Torque reference [13 Bits]

In the example above, if the EN input is active and the digital input 1 off, then the block will generate a speed reference according to the user parameter P1010 in the 13 bit unit. If there is no error (e.g., disabled inverter), the ENO output goes to 1.

3.4 POSITIONING BLOCKS

3.4.1 Stop in Position – POSITION0



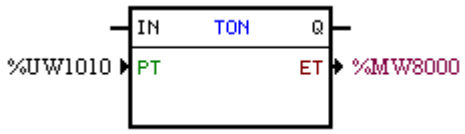
Menu: *Insert-Function Blocks-Positioning-POSITION0*
Entrada:
EN: Enables the block
Saída:
ENO: Goes to 1 when the motor stop
Properties:
MINSPEED:
Minimum speed to stop [13 Bits]

When the EN input is activated, if the drive is not enabled and the P0229 is not set to 1, is generated A702. If the P0202 is not in 5, or any other block POSITION0 was active, the block is not enabled. After this checks, the reference speed is monitored and when it becomes equal or less than the value of MINSPEED, the block is allocated at the current position, with the SoftPLC commands run goes to 1 and the SoftPLC speed reference goes to 0.

In the example above, if the EN input is active, the block is only allocated in position if the speed reference reaches the value 0, and then the ENO output goes to 1.

3.5 CLP BLOCKS

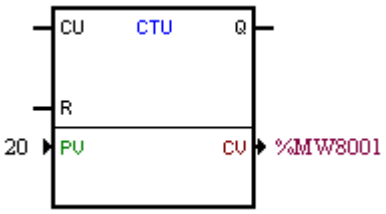
3.5.1 Timer – TON



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.

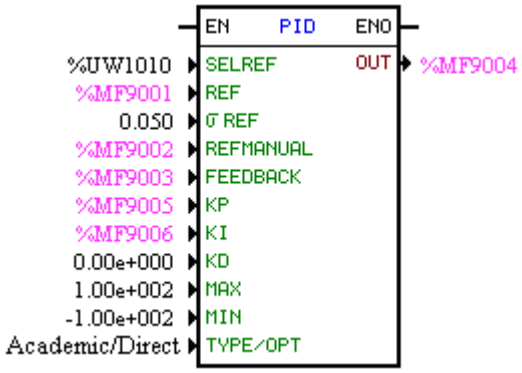
3.5.2 Incremental Counter– CTU



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.

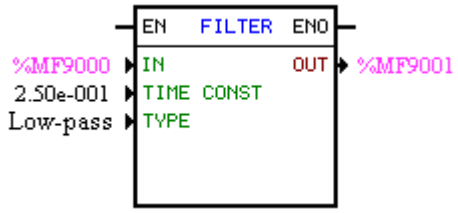
3.5.3 Proportional-Integral-Derivative Controller – PID



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 converter into a PI. The control output OUT, represented by the float marker 9004, has the maximum and minimum limits of 100 and -100.

3.5.4 Low-pass or High-pass Filter – FILTER

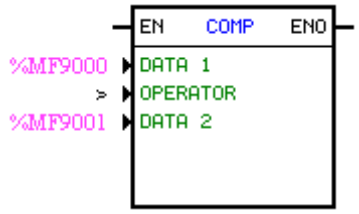


Menu: *Insert - Function Blocks – PLC-FILTER.*
Inputs:
EN: Enables the block.
Output:
ENO: EN Input image.
Properties:
TS: Sampling time.
IN: Input data.
TIMECONST: Filter time constant.
TYPE: Low-pass/High-pass.
OUT: Input data filtered value.

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.

3.6 CALCULATION BLOCKS

3.6.1 Comparator – COMP

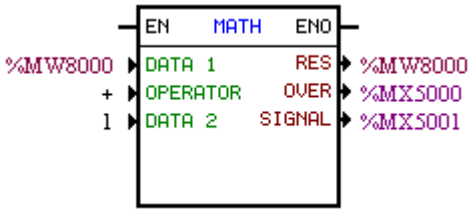


Menu: *Insert - Function Blocks – Calculation-COMP.*
Input:
EN: Enables the block.
Output:
ENO: Goes to 1 when the comparison condition is fulfilled.
Properties:
FORMAT: Integer or floating point.
DATA 1: Comparison data 1.
OPERATOR: Comparison operator.
DATA 2: Comparison data 2.

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).

3.6.2 Math Operation – MATH

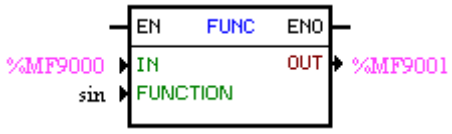


Menu: *Insert - Function Blocks – Calculation-MATH.*
Input:
EN: Enables the block.
Output:
ENO: Indicates if the calculation has been executed.
Properties:
FORMAT: Integer or floating point.
DATA1: Calculation data 1. It may also appear as DATA1H and DATA1L (representing the high and low parts of the 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 if 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).

3.6.3 Math Function – FUNC

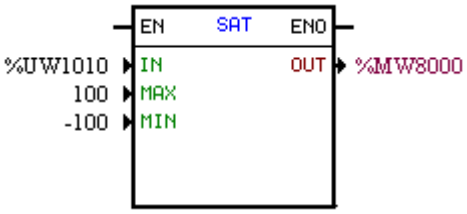


Menu: *Insert - Function Blocks – Calculation-FUNC.*
Input:
EN: Enables the block.
Output:
ENO: Indicates if the calculation has been executed.
Properties:
FORMAT: Integer or floating point.
IN: Data to be calculated.
FUNCTION: Mathematic function (sin, cos, etc).
OUT: Calculation result.

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).

3.6.4 Saturator – SAT



Menu: *Insert - Function Blocks – Calculation-SAT.*
Input:
EN: Enables the block.
Output:
ENO: Indicates if 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.

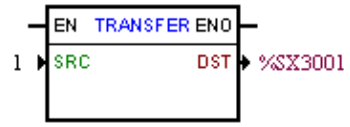
In the example above, when the EN input is active, the word marker 8000 contains the user parameter P1010 value, limited however, between the maximum of 100 and the minimum of -100.

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, the outputs OUT and ENO are reset to zero.

3.7 TRANSFER BLOCKS

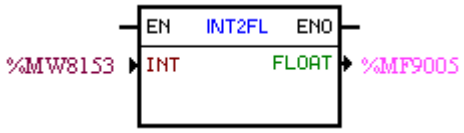
3.7.1 Data Transfer – TRANSFER



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: Destine 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).

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

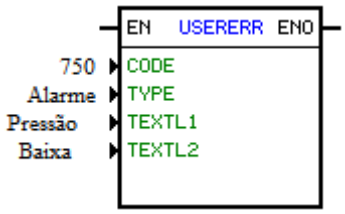


Menu: *Insert - Function Blocks- Transfer -INT2FL.*
Input:
EN: Enables the block.
Output:
ENO: Indicates that the transfer has been done.
Properties:
INT: Integer data.
FLOAT: Data converted into floating point.

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 bit + sign (-32768 to 32767).

3.7.3 User Fault or Alarm Generator – USERERR

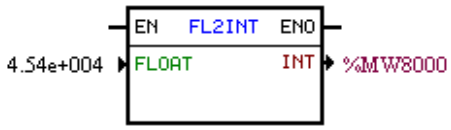


Menu: *Insert - Function Blocks - Transfer - USERERR*
Input:
EN: Enables the block
Output:
ENO: It indicates 1 when EN = 1 and the alarm or error has been effectively generated.
Properties:
CODE: Alarm or fault code
TYPE: 0: Generates alarm, 1: Generates fault
TEXTL1: HMI line 1 text
TEXTL2: HMI line 2 text

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

NOTE!
If the block is configured for Fault, then it will be necessary to reset the drive in order to be able to enable it again.

3.7.4 Converts from Floating Point to Integer (16 bit) – FL2INT

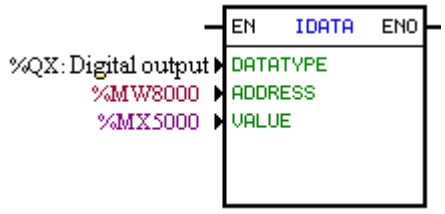


Menu: *Insert - Function Blocks- Transfer -FL2INT.*
Input:
EN: Enables the block.
Output:
ENO: Indicates that the transfer has been done.
Properties:
FLOAT: Floating point data.
INT: Data converted into integer.

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 bit + sign (-32768 to 32767).

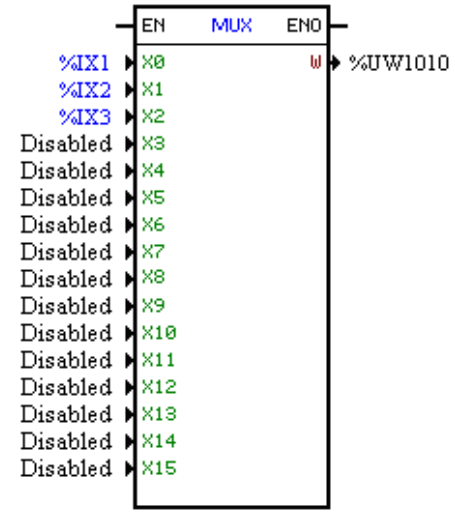
3.7.5 Indirect Data Transfer– IDATA



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.

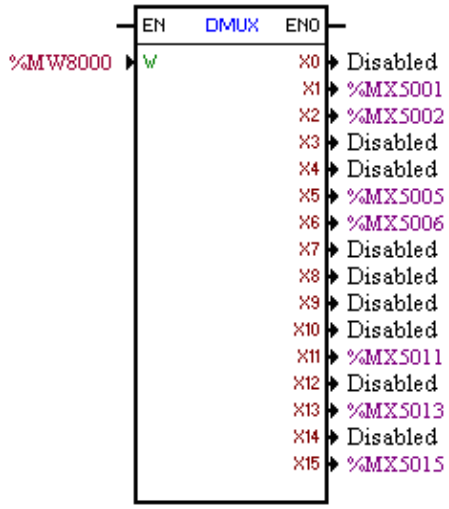
3.7.6 Multiplexer – MUX



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.

3.7.7 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.

4 INVERTER PARAMETER SETTINGS

In the continuation, only the parameters of the CFW-700 frequency inverter that are related to the SoftPLC will be presented.

4.1 SYMBOLS FOR THE PROPERTIES DESCRIPTION

RO	Read-only parameter.
CFG	Parameter that can be changed only with a stopped motor.
Net	Parameter visible on the keypad if the inverter has a network interface installed– RS232, RS485, CAN, Anybus-CC, Profibus – or if the USB interface is connected.
Serial	Parameters visible on the keypad if the inverter has the RS232 or the RS485 interface installed.

4.2 CFW700 CONFIGURATION PARAMETERS

P0100 – Acceleration Time

P0101 – Deceleration Time

P0220 – LOCAL/REMOTE Selection Source

P0221 – Speed Reference Selection – LOCAL Situation

P0222 – Speed Reference Selection – REMOTE Situation

P0223 – FORWARD/REVERSE Selection - LOCAL Situation

P0226 – FORWARD/REVERSE Selection - REMOTE Situation

P0224 – Run/Stop Selection – LOCAL Situation

P0227 – Run/Stop Selection - REMOTE Situation

P0225 – JOG Selection – LOCAL Situation

P0228 – JOG Selection - REMOTE Situation

P0251 – AO1 Function

P0254 – AO2 Function

P0275 – DO1 Function (RL1)

P0276 – DO2 Function (RL2)

P0277 – DO3 Function (RL3)

P0278 – DO4 Function

P0279 – DO5 Function

NOTE!
For further information, please refer to the CFW700 Programming Manual.

4.3 SOFTPLC EXCLUSIVE PARAMETERS

P1000 – SoftPLC Status

Adjustable Range:	0 = No Applicative 1 = Install. App.. 2 = Incompat. App. 3 = App. Stopped 4 = App. Running	**Factory Setting:** 0
Properties:	RO	
Access groups via HMI:	SPLC	

Description:
It allows the user to visualize the SoftPLC status. If there is no installed applicative, the parameters from P1001 to P1049 will not be showed on the keypad.

If this parameter presents the option 2 (“Incompat. App.”), it indicates that the version that has been loaded in the flash memory board is not compatible with the current CFW700 firmware.

In this case it is necessary to recompile the project in the WLP, considering the new CFW700 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 be known or that the password be not enabled.

P1001 – SoftPLC Command

Adjustable Range:	0 = Stop Program. 1 = Run Program. 2 = Delete Program.	**Factory Setting:** 0
Properties:	CFG	
Access groups via HMI:	SPLC	

Description:
It allows stopping, running or excluding the installed applicative, for that reason, the motor must be disabled.

P1002 – Scan Cycle Time

Adjustable Range:	0.0 to 999.9 s	**Factory Setting:** -
Properties:	CFG	
Access groups via HMI:	SPLC or READ	

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

P1003 – SoftPLC Applicative Software

Adjustable Range:	0 = User 1 = PID Regulator 2 = Electronic Potentiometer (EP) 3 = Multispeed 4 = 3-Wire Start/Stop 5 = FWD/REV	**Factory Setting:** 0
Properties:	CFG	
Access groups via HMI:	SPLC	

Description:
Allows the user to select the applicative included in the CFW700 inverter.

P1003	Description
0	The applicative that will run in the SoftPLC is that loaded by the user via ladder programming.
1	The applicative that will run in the SoftPLC is the PID regulator. It can be used to control a closed loop process. This applicative sets proportional, integral and derivative regulator superimposed to the regular speed control of the CFW700 inverter.
2	The applicative that will run in the SoftPLC is the electronic potentiometer. It allows the motor speed reference settings via two digital inputs, one for speeding up the motor and another to slow down the motor.
3	The applicative that will run in the SoftPLC is the multispeed. It allows speed reference settings based on to the values defined in some parameters (P1011 to P1018) with a logical combination of the digital inputs DI4, DI5 and DI6, limited to 8 pre-programmed speed references. Advantages such as stability of fixed pre-programmed references and electrical noise immunity (isolated digital inputs DIX) are noted in this kind of application.
4	The applicative that will run in the SoftPLC is the 3-Wire Start/Stop. It allows the inverter to start/stop as with a retention contact and an emergency button.
5	The applicative that will run in the SoftPLC is the FWD/REV command. It gives the user the combination of two inverter commands in a single digital input (forward/reverse and start/stop).

NOTE!
For additional information refer to the chapter 19 of the CFW700 programming and troubleshooting manual.

P1010 to P1059 – SoftPLC Parameters

Adjustable Range: 0 to 65535 **Factory Setting:** 0
Properties: CFG
Access groups via HMI: SPLC

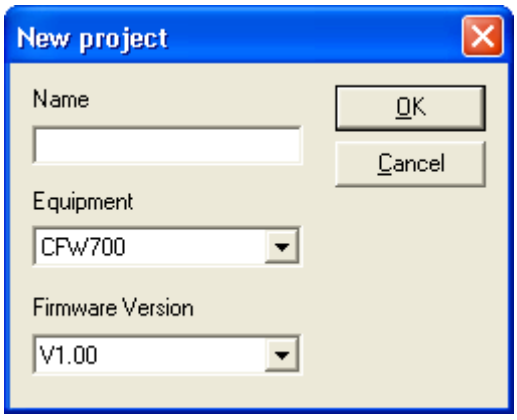
Description:
They consist of parameters with functions defined by the user by means of the WLP software. It is also possible for the user to configure these parameters as described in the item 5.5.

5 RESUME OF THE WLP MAIN FUNCTIONS

This chapter brings basic information about the operations done with the WLP software for the CFW700 inverter programming. More information can be obtained in the manual or in the help of the WLP software.

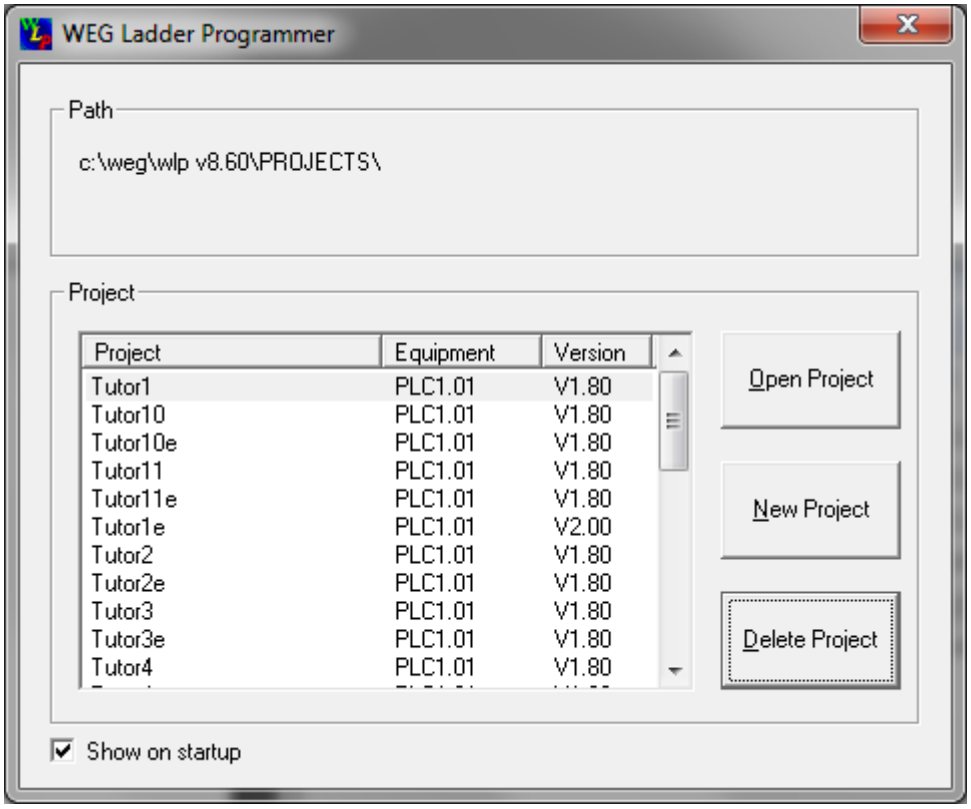
5.1 PROJECT – NEW

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



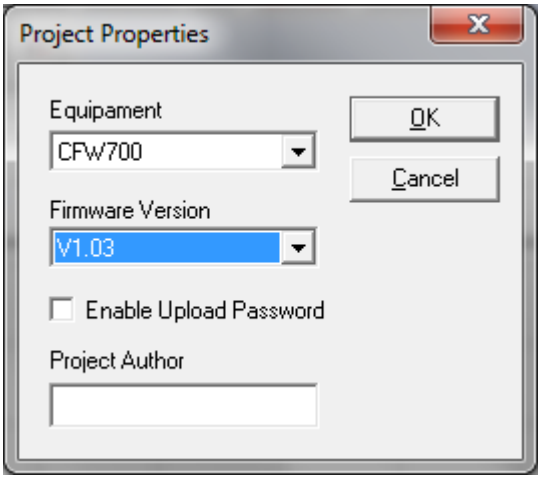
5.2 PROJECT – OPEN

It opens the selected project.



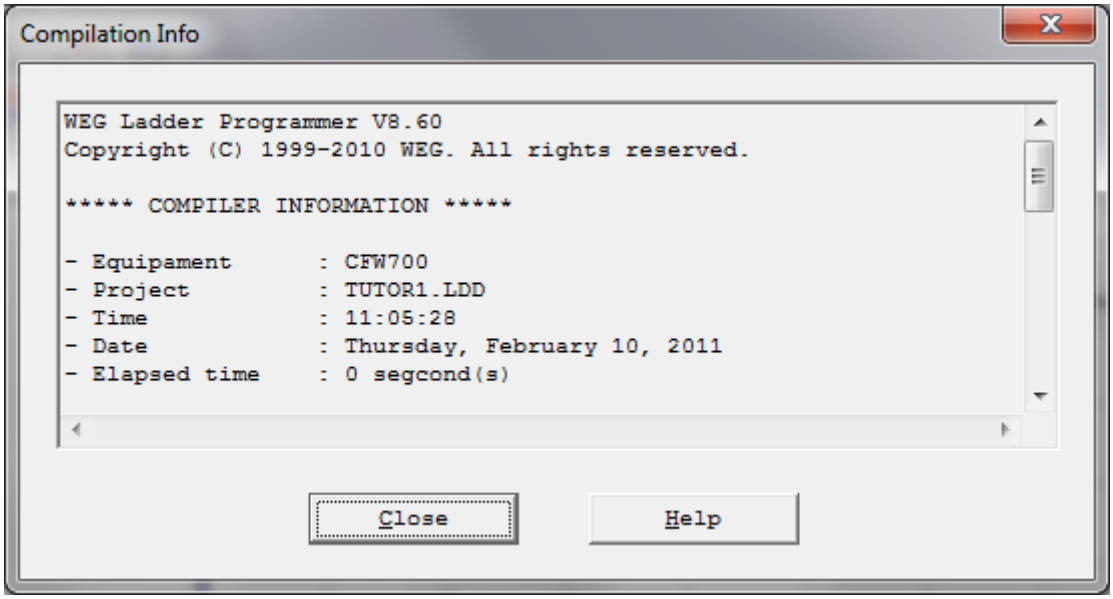
5.3 PROJECT – PROPERTIES

It allows the user to redefine the equipment and the firmware version. In this box it is also configured whether or not the project will have upload password.



5.4 VIEW – COMPILATION INFO

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

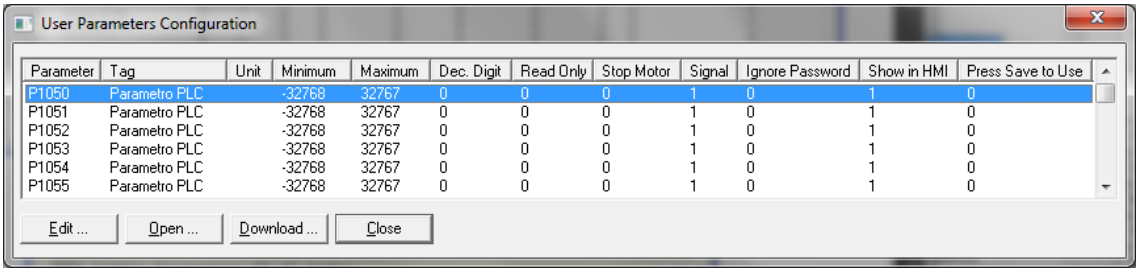


5.5 VIEW – USER PARAMETER CONFIGURATION

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

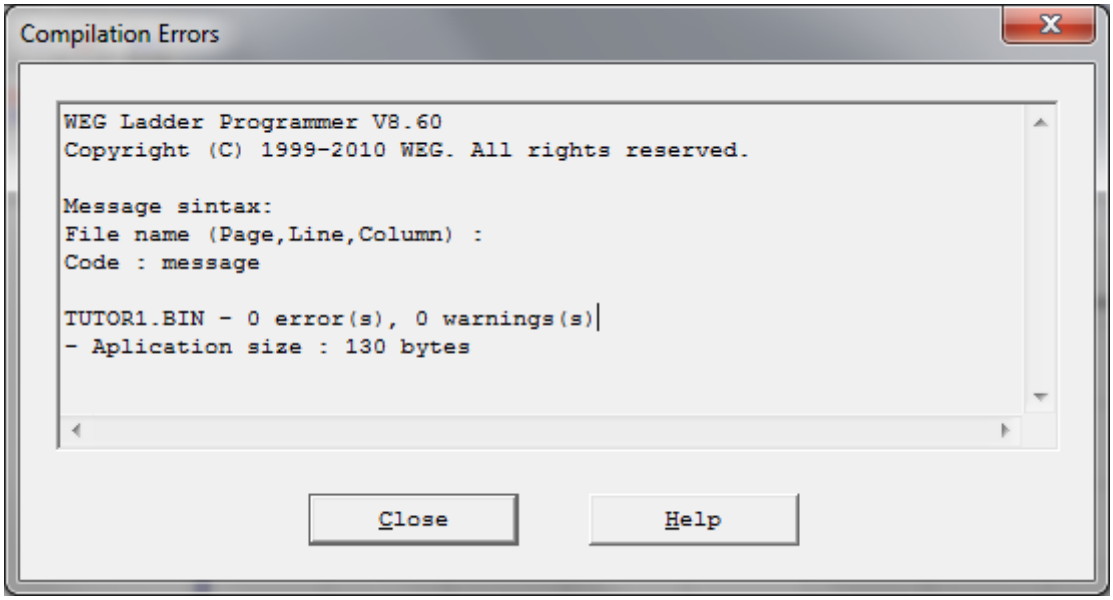
- Units.
- Minimum and maximum limit.
- Number of decimal positions.
- Hexadecimal or normal format.
- Reading or writing only.
- Parameter changing: no confirmation, stopped motor or stopped motor + save.
- With or without sign.
- Ignores the password (allows modification regardless of P0005) or normal.
- Password level: always view and ignores the password, always view and enables the password, only view or never view the password.
- Allows saving the parameter value (retentive), when it is used in some blocks (PLC, Calculations and Transfers) on power down.

Those configurations can be transferred to the CFW700 with the “Download” button.



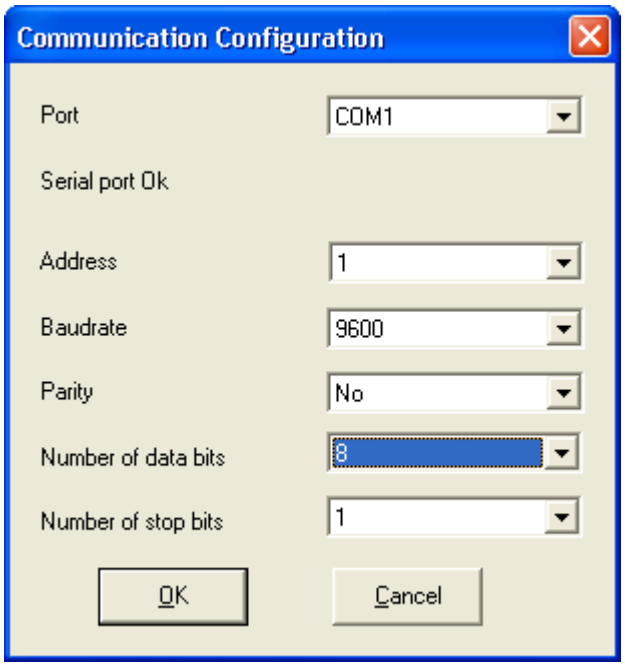
5.6 CONSTRUCT – COMPILE

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



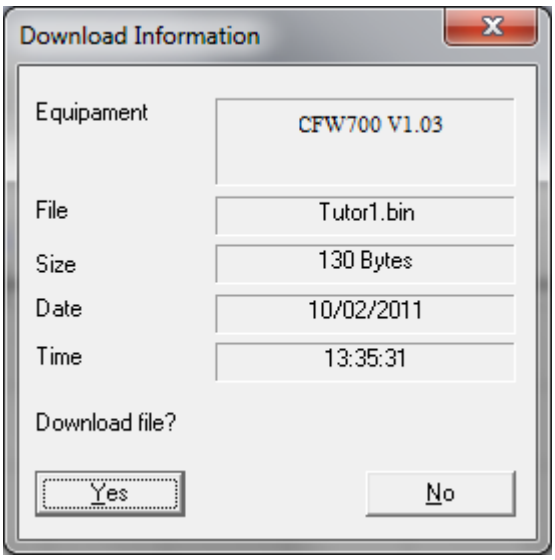
5.7 COMMUNICATION – CONFIGURATION

The USB port is used for the CFW700. Therefore, the USB driver must be installed. The driver is found in the DRIVER_USB folder, inside the WLP V7.2X.



5.8 COMMUNICATION – DOWNLOAD

This command allows downloading the applicative and/or the user parameter configurations to the CFW700.



6 FAULTS, ALARMS, AND POSSIBLE CAUSES

Table 6.1: *"Faults", "Alarms", and Possible Causes*

Fault/Alarm	Description	Possible Causes
A702: Inverted Disabled	It occurs when the movement block (REF block) is active and the drive general enabling command is not active.	■ Verify if the general enabling command of the drive 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 references in the local and/or remote modes (P0221 and P0222).
F711: Fault in the execution of the SoftPLC	Fault in the execution of the SoftPLC.	■ Incompatible user program. ■ Failure during loading the user program. ■ WLP version incompatible with the inverter firmware version.