SoftPLC

CFW100

User Manual

SoftPLC User Manual

Series: CFW100

Language: English

Document Number: 10002965849 / 02

Publication Date: 10/2014

CONTENTS

ABOUT THE MANUAL 5
ABBREVIATIONS AND DEFINITIONS 5
1 INTRODUCTION TO SOFTPLC 6
1.1 SYMBOL OF THE DATA TYPES 6
2 SOFTPLC MEMORY 7
2.1 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 13
3.3.1 Speed and/or Torque Reference – REF 13
3.4 PLC BLOCKS 14
3.4.1 Timer – TON 14
3.4.2 Incremental Counter – CTU 14
3.4.3 Proportional-Integral-Derivative Controller – PID 14
3.4.4 Low-pass or High-pass Filter – FILTER 15
3.5 CALCULATION BLOCKS 15
3.5.1 Comparator – COMP 15
3.5.2 Math Operation – MATH 15
3.5.3 Math Function – FUNC 16
3.5.4 Saturator – SAT 16
3.6 TRANSFER BLOCKS 17
3.6.1 Data Transfer – TRANSFER 17
3.6.2 Conversion from Integer (16 bits) to Floating Point – INT2FL 17
3.6.3 User Fault or Alarm Generator - USERERR 17
3.6.4 Convert from Floating Point to Integer (16 bits) – FL2INT 17
3.6.5 Indirect Data Transfer – IDATA 18
3.6.6 Multiplexer – MUX 18
3.6.7 Demultiplexer – DMUX 19
4 INVERTER PARAMETER SETTINGS 20
4.1 SYMBOLS FOR PROPERTY DESCRIPTION 20
4.2 CFW100 CONFIGURATION PARAMETERS 20
4.3 EXCLUSIVE SOFTPLC PARAMETERS 21

P900 – SoftPLC Status 21
P901 – SoftPLC Command 21
P902 – Scan Cycle Time 21
P910 to P959 – SoftPLC User Parameters 21

5 SUMMARY OF THE WLP MAIN FUNCTIONS 22

5.1 PROJECT – NEW 22
5.2 PROJECT – OPEN 22
5.3 PROJECT – PROPERTIES 23
5.4 VIEW – COMPILATION INFORMATION 23
5.5 VIEW – USER PARAMETER CONFIGURATION 23
5.6 CONSTRUCT – COMPILE 24
5.7 COMMUNICATION – CONFIGURATION 24
5.8 COMMUNICATION – DOWNLOAD 25
5.9 COMMUNICATION – UPLOAD 26

6 ALARMS, FAULTS AND POSSIBLE CAUSES 27

ABOUT THE MANUAL

This manual provides the required information for the operation of the CFW100 frequency inverter using the user's programming module, called SoftPLC. This manual must be used together with the user's manual of the CFW100 and WLP software.

ABBREVIATIONS AND DEFINITIONS

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

1 INTRODUCTION TO SOFTPLC

The SoftPLC is a resource that adds the functionalities of a PLC to the CFW100, making the product flexible and allowing the user to develop its own applications (user's programs).

The main SoftPLC characteristics are:

- Programming in "Ladder Language" using the WLP software.
- Access to all Parameters and I/Os of the CFW100.
- 50 configurable parameters for the user to set.
- PLC, Mathematical and Control Blocks
- On-line transfer and monitoring of the software application via Serial/USB interface.
- Transfer of the application software installed on the CFW100 to the PC with password restriction.
- Storage of the software application on the FLASH memory module.
- Execution directly on the RAM.

1.1 SYMBOL OF THE DATA TYPES

%KW word-type constants (16 bits)
%KF float-type constants (32 bits, floating point)
%MX bit markers
%MW word markers (16 bits)
%MF float markers (32 bits, floating point)
%SX system bit markers
%SW system word markers (16 bits)
%IX digital Inputs
%IW analog inputs (16 bits)
%QX digital outputs
%QW analog outputs (16 bits)
%UW user parameters (16 bits)
%UW system parameters (16 bits)
%PD drive parameter (16 bits)

2 SOFTPLC MEMORY

The total size of the SoftPLC memory is 4712 bytes for programming memory and data memory.

2.1 MEMORY

- SoftPLC function: 4712 bytes
- SoftPLC User Parameter: 408 bytes

2.2 DATA MEMORY

The SoftPLC data memory area (user variables) is shared with the programming memory. Therefore, the total size of an application may vary as a function of the number of variables used by the user.

The bit, word and float markers are allocated according to the LAST address used on the application, that is, the longer the last address, the larger the allocated area. Therefore, it is recommended to use the markers in a SEQUENTIAL manner.

The word and float constants also use programming memory space.

2.2.1 Constants

Table 2.1: *Memory Map of the Constants*

Sym.	Description	Bytes
%KW	Word Constants (16 bits)	It depends on the number of different word constants. E.g.: If it was used: - %KW: 327 = 2 bytes - %KW: 5; 67 = 4 bytes - %KW: 13; 1000; ~~13~~; 4 = 6 bytes
%KF	Float Constants (32 bits – IEEE)	It depends on the number of different float constants. E.g.: If it was used: - %KF: -0.335 = 4 bytes - %KF: 5.1; 114.2 = 8 bytes - %KF: 0.0; 115.3; ~~0,0~~; 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 ... 3	2
%IW	Analog/Frequency Inputs	1 ... 2	4
%QW	Analog Output	1	2

NOTE!
The %IW3 marker corresponds to the frequency input. In order to activate this input, it is necessary to set P246 to 1.

NOTE!
The values of the Analog/Frequency Inputs (%IW) and Analog Output (%QW) read and written via SoftPLC respect their gains (P232, P247: %IW1, %IW3 e P252: %QW1) and offsets (P234, P249: %IW1, %IW3).

NOTE!
The values read and written via SoftPLC obey the following rules, respecting the parameters related to the analog input and output signal types (P233: %IW1 and P253: %QW1):

- 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

2.2.3 Volatile Markers (Variables)

They consist of variables that can be applied by the user to execute the logics of the application. 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 application 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 may not be available in the parameters. They may be: system bit markers (1 bit) or system word markers (16 bits).

Table 2.4.a: *Memory Map for the Odd System Bits – Writing/Command*

Sym.	Description		Range	Bytes
Type	System Bits		3000 ... 3040	4 bytes
%SX	*Writing/Command (odd)*			
	3001	General Enable		**0:** It disables the inverter completely, interrupting the motor supply. **1**: It enables the inverter, allowing the motor operation.
	3003	Run/Stop		**0**: Stops the motor by deceleration ramp. **1:** The motor runs according to the acceleration ramp until reaching the speed reference value.
	3005	Speed Direction		**0**: The motor runs in the reverse direction. **1:** The motor runs in the forward 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 fault condition, it resets the inverter. **NOTE:** When this command is executed, the inverter and the SoftPLC Application are reinitialized. This is also true for reset via keypad.
	3021	Activate the Second Ramp		**0:** The values for the motor acceleration and deceleration are those from the first Ramp (P100 and P101). **1:** The values for the motor acceleration and deceleration are those from the first Ramp (P102 and P103). **Note:** Set P105 to 6 in order to enable the selection via SoftPLC.

Table 2.4.b: *Memory Map for the Even System Bit – Reading/State*

Sym.	Description		Range	Bytes
Type	System Bits		3000 ... 3020	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 running reverse. **1:** The motor is running forward.
	3006	JOG		**0:** JOG function inactive. **1:** JOG function active.
	3008	LOC/REM		**0:** Inverter in local mode. **1:** Inverter in remote mode.
	3010	In Fault		**0:** Inverter is not in fault state. **1:** Some fault registered by the inverter. **Note:** The number of the fault can be read through parameter P049 – Current Fault.
	3012	Undervoltage		**0:** No undervoltage. **1:** With undervoltage.
	3016	Alarm Condition		**0:** Inverter is not in an alarm condition. **1:** Inverter is in an alarm condition. **Note:** The number of the alarm can be read by means of parameter P048 – Current Alarm.
	3018	In Configuration Mode		**0:** Inverter operating normally. **1:** Inverter in configuration state. It indicates a special condition when the inverter cannot be enabled: - There is a parameter setting incompatibility. **Note:** Parameter P047 shows the cause of the parameter setting incompatibility.
	3020	Ramp Active		**0:** It indicates that the first Ramp is active. **1:** It indicates that the second Ramp is active.
	3032	Start key ( )		**0:** Not pressed. **1:** Pressed during 1 scan cycle. **Note:** Only use to visualize the key status with Run/Stop Selection via HMI keys (P224 = 0 or P227 = 0).
	3034	Stop key ( )		
	3046	Up key ( )		**0:** Not pressed. **1:** Pressed during 1 scan cycle.
	3048	Down key ( )		

Table 2.5: *Memory Map for the Even System Word Markers*

Sym.	Description		Range	Bytes
%SW	System Words		3300 ... 3324	48 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		
	3320	Inverter rated current [A x10]		
	3322	Unfiltered motor current (P003) [A x10]		
	3324	Unfiltered motor torque [% x10]		

NOTE!
The system word marker %SW3300 uses a 13-bit resolution (8192 ⟶ to 8191), which represents the motor synchronous speed. Thus, if the speed reference via "Reference" block (%SW3301) is 4096 for a VI pole motor (this means a synchronous speed of 1200 rpm), the motor will run at 600 rpm.

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

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

NOTE!
Equation for the calculation of the motor speed in Hz:

$$\text{Speed in Hz} = \frac{\text{synchronous frequency in Hz (P403) x speed in 13 bits}}{8192}$$

2.2.5 Parameters

The parameters from P910 to P959 appear on the keypad of the CFW100 only when there is a valid application (user program) in the memory, i.e., when P900 > 0.

Table 2.6: *Parameter Memory Map*

Sym.	Description	Range	Bytes
%PD	System Parameters (refer to the CFW100 manual)	0... 999	
%PW	SoftPLC Parameters	900 ... 959	6 bytes
	P900: SoftPLC Status [Read-only parameter]	0: No application 1: Install. Applic. 2: Applic. Incomp. 3: Applic. Stopped 4: Applic. Running	
	P901: SoftPLC Command	0: Stop Program 1: Run Program 2: Delete Program	
	P902: Scan Cycle Time [ms] [Read-only parameter]		
%UW	User Parameters	910 ... 959	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 ... 3	2401...2403
%IW	Analog/Frequency Inputs	1 ... 2	2601...2602
%QW	Analog Output	1	2801

NOTE!
The %IW3 marker corresponds to the frequency input. In order to activate this input, it is necessary to set P246 to 1.

NOTE!
All the other data types have the user addresses (SoftPLC) equal to the Modbus addresses. E.g.: %PD0100 = Modbus address 100; %MX5000 = Modbus address 5000; %SW3308 = Modbus address 3308.

2.3.2 Protocol

Refer to the Modbus RTU (CFW100) User Manual, Modbus Protocol chapter.

3 RESUME OF THE FUNCTION BLOCKS

This chapter contains a summary of the function blocks that are available for the user programming.

3.1 CONTACTS

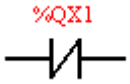
They send to the stack the content of a programmed data (0 or 1), which can be of the following type:
- %MX: Bit Marker
- %IX: Digital Input
- %QX: Digital Output
- %UW: User Parameter
- %SX: System Bit Marker – Reading

3.1.1 Normally Open Contact – NO CONTACT



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

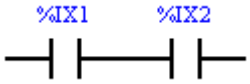
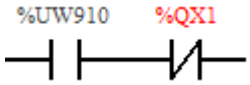
3.1.2 Normally Closed Contact – NC CONTACT



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

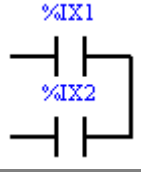
3.1.3 "AND" Logic with Contacts

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	0
	0	1	0
	1	0	0
	1	1	1
 %UW910. (~%QX1)	%UW910	%QX1	Stack
	0	0	0
	0	1	0
	1	0	1
	1	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	0
		0	1	1
		1	0	1
		1	1	1
	%UW910 + (~%QX1)	%UW910	%QX1	Stack
		0	0	1
		0	1	0
		1	0	1
		1	1	1

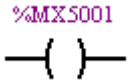
3.2 COILS

They save the content of the stack in the programmed data (0 or 1), which can be of the following type:

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

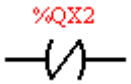
It is allowed to add coils in parallel in the last column.

3.2.1 Normal Coil - COIL



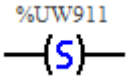
Menu: *Insert-Coils-COIL*
E.g.: It sets the bit marker 5001 with the stack content

3.2.2 Negated Coil – NEG COIL



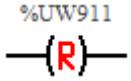
Menu: *Insert-Coils-NEG COIL*
E.g.: It sets the digital output 2 with the negated content of the stack

3.2.3 Set Coil – SET COIL



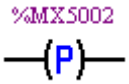
Menu: *Insert-Coils-SET COIL*
E.g.: It sets the user parameter 911, provided that the content of the stack is not 0

3.2.4 Reset Coil – RESET COIL



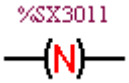
Menu: *Insert-Coils-RESET COIL*
E.g.: It resets the user parameter 911, provided that the content of the stack is not 0

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.

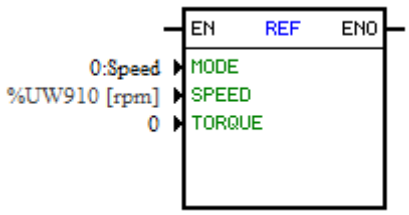
3.2.6 Negative Transition Coil – NTS COIL



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

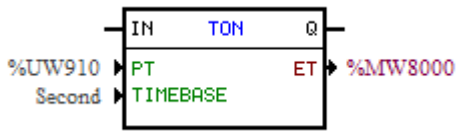


Menu: *Insert-Function Blocks-Movement-REF*
Input:
EN: Enables the block
Output:
ENO: Goes to 1 when EN ≠ 0 and without error
Properties:
MODE: 0=Speed Mode
SPEED: Speed reference [RPM,13 Bits, Hz (x10)]
TORQUE: Not available

In the example above, if the EN is active, and the digital input 1 is off, then the block will generate a speed reference according to the user parameter 910 in the rpm unit. If there is no error (example: disabled inverter), the ENO output goes to 1.

3.4 PLC BLOCKS

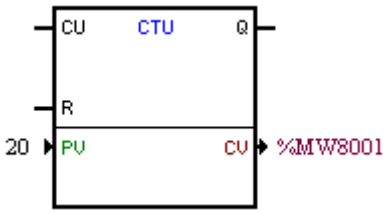
3.4.1 Timer – TON



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

In the example above, if the IN input is active and the content of the word marker 8000 is greater than or equal to the content of the user parameter 910, the output Q goes to set1.

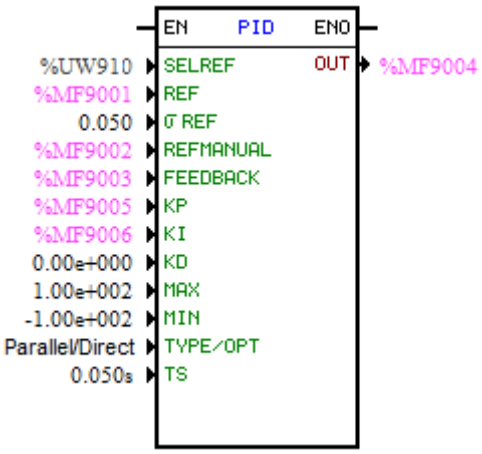
3.4.2 Incremental Counter – CTU



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: Preset value
CV: Counter Value

In the example above, if the content of the word marker 8001 is greater than or equal to 20, the output Q goes to 1.

3.4.3 Proportional-Integral-Derivative Controller – PID

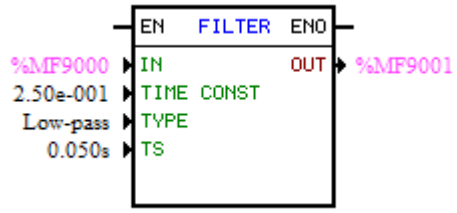


Menu: *Insert-Function Blocks-PLC-PID*
Inputs:
EN: Enables the block
Output:
ENO: EN input image
Properties:
TS: Sampling period
SELREF: Automatic/manual reference
REF: Automatic reference
δREF: Time constant of the automatic reference filter
REFMANUAL: Manual reference
FEEDBACK: Process feedback
KP: 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 input EN is active, the controller starts its operation. The content of the user parameter 910 selects the reference that is active, that is, whether it is the float marker 9001 (automatic reference) or 9002 (manual reference). There is a 0.05s filter for the automatic reference. Since the derivative

gain is fixed in 0, this indicates that de PID was converted into a PI. The value of the control output OUT, represented by the float marker 9004, has the maximum and minimum limits of 100 and -100 and the sampling time of 0.050s.

3.4.4 Low-pass or High-pass Filter – FILTER

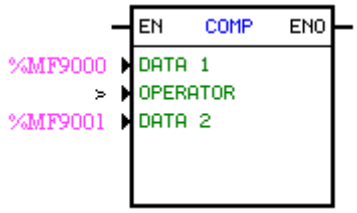


Menu: *Insert-Function Blocks-PLC-FILTER*
Inputs:
EN: Enables the block
Output:
ENO: EN input image
Properties:
TS: Sampling period
IN: Input data
TIMECONST: Filter time constant
TYPE: Low-pass/High-pass
OUT: Input data filtered value

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 a low-pass filter and it will be transferred to the float marker 9001.

3.5 CALCULATION BLOCKS

3.5.1 Comparator – COMP

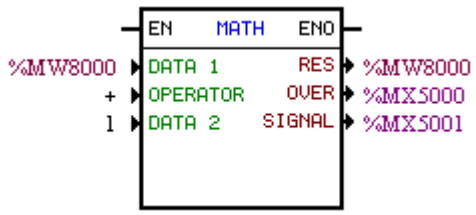


Menu: *Insert-Function Blocks-Calculation-COMP*
Input:
EN: Enables the block
Output:
ENO: Goes to 1 when the comparison condition is fulfilled
Properties:
FORMAT: Integer or floating point
DATA 1: Comparison data 1
OPERATOR: Comparison operator
DATA 2: Comparison data 2

In the example above, if the EN input is active and the content of the float marker 9000 is greater than the float marker 9001, then the output ENO goes to 1.

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

3.5.2 Math Operation – MATH



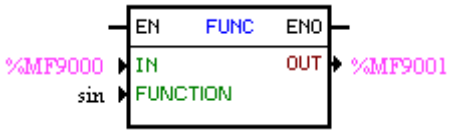
Menu: *Insert-Function Blocks-Calculation-MATH*
Input:
EN: Enables the block
Output:
ENO: Indicates if the calculation has been executed
Properties:
FORMAT: Integer or floating point
DATA1: Calculation Data 1. It may also appear as DATA1H and DATA1L (representing the high and low parts of data 1)
OPERATOR: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

In the example above, when the input EN 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 FORMAT is integer, all the numerical data are considered words of 15 bits + signal (-32768 to 32767).

3.5.3 Math Function – FUNC



Menu: *Insert-Function Blocks-Calculation-FUNC*
Input:

EN:	Enables the block

Output:

ENO:	Indicates if the calculation was executed

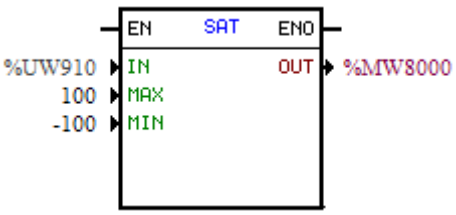
Properties:

FORMAT:	Integer or floating point
IN:	Data to be calculated
FUNCTION:	Mathematic function (sin, cos, etc.)
OUT:	Calculation result

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

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

3.5.4 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

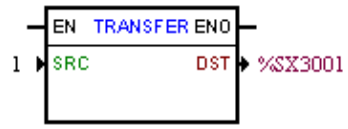
In the example above, when the EN input is active, the word marker 8000 contains the user parameter 910 value, but limited between the maximum of 100 and the minimum of -100.

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

NOTE!
If the value of MIN is greater than the MAX, the outputs OUT and ENO are reset to zero.

3.6 TRANSFER BLOCKS

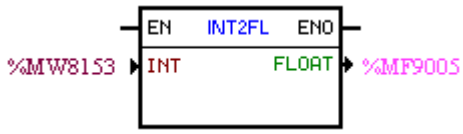
3.6.1 Data Transfer – TRANSFER



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).

3.6.2 Conversion from Integer (16 bits) to Floating Point – INT2FL

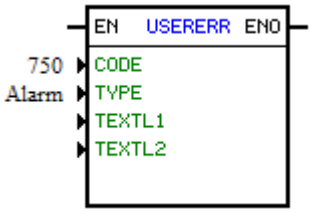


Menu: *Insert-Function Blocks-Transfer-INT2FL*
Input:
EN: Enables the block
Output:
ENO: Indicates that the transfer has been done
Properties:
INT: Integer data
FLOAT: Data converted into floating point

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

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

3.6.3 User Fault or Alarm Generator - USERERR

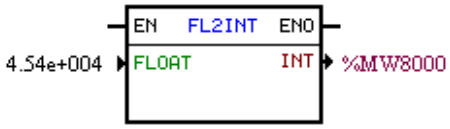


Menu: *Insert-Function Blocks-Transfer-USERERR*
Input:
EN: Enables the block
Output:
ENO: It indicates 1 when EN = 1 and the alarm or error has been effectively generated.
Properties:
CODE: Fault or alarm code.
TYPE: 0: Generates alarm, 1: Generates fault
TEXTL1: HMI line 1 text (Not available)
TEXTL2: HMI line 2 text (Not available)

In the example above, if the EN input is active, A750 will appear.

NOTE!
If this block is configured as Fault, it is necessary to reset the drive so as to enable it again.

3.6.4 Convert from Floating Point to Integer (16 bits) – FL2INT



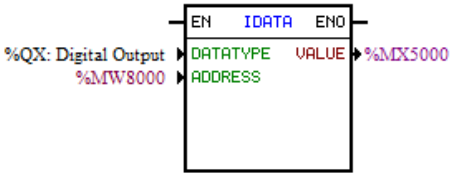
Menu: *Insert-Function Blocks-Transfer-FL2INT*
Input:
EN: Enables the block
Output:

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 $4.54x10^4$ is converted into an integer with signal via word marker 8000. However, after the conversion, the word marker 8000 will remain with the value 32767, because this is the positive limit for a Word.

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

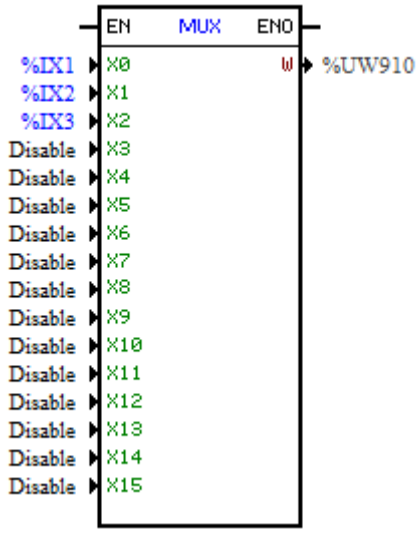
3.6.5 Indirect Data Transfer – IDATA



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.

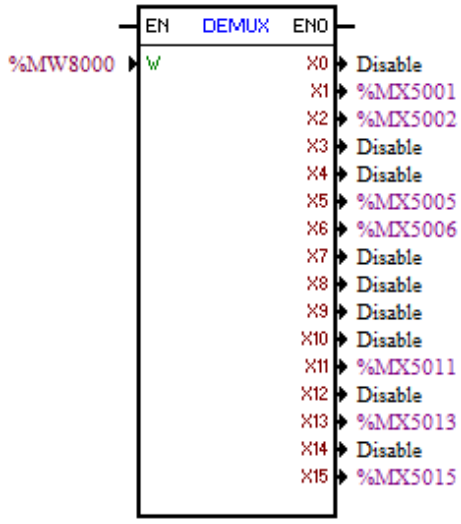
3.6.6 Multiplexer – MUX



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 P910.

3.6.7 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 above example, 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

Below, only the parameters of the CFW100 frequency inverter that are related to the SoftPLC will be presented.

4.1 SYMBOLS FOR PROPERTY DESCRIPTION

ro Read only parameter
cfg Parameter that can be changed only with a stopped motor

4.2 CFW100 CONFIGURATION PARAMETERS

P100 – Acceleration Time

P101 – Deceleration Time

P220 – Selection of the LOCAL/REMOTE Source

P221 – Selection of the Speed Reference – LOCAL Status

P222 – Selection of the Speed Reference – REMOTE Status

P223 – Selection of the Direction of Rotation – LOCAL Status

P224 – Selection of Run/Stop (Local Status)

P225 – Selection of JOG – LOCAL Status

P226 – Selection of the Direction of Rotation – REMOTE Status

P227 – Selection of Run/Stop (Remote Status)

P228 – Selection of JOG – REMOTE Status

P246 – Frequency Input FI

P251 – Analog Output AO1 Function

P263 – Digital Input DI1 Function

P264 – Digital Input DI2 Function

P265 – Digital Input DI3 Function

P266 – Digital Input DI4 Function

P267 – Digital Input DI5 Function

P268 – Digital Input DI6 Function

P269 – Digital Input DI7 Function

P270 – Digital Input DI8 Function

P271 – Digital Input Signal

P275 – DO1 Output Function (RL1)

P276 – DO2 Output Function (RL2)

P277 – DO3 Output Function (RL3)

NOTE!
The resources of available inputs and outputs depend on the used accessory.
For further information, please, refer to the CFW100 Programming Manual.

4.3 EXCLUSIVE SOFTPLC PARAMETERS

P900 – SoftPLC Status

Adjustable Range:	0 = No Application 1 = Install. Applic. 2 = Applic. Incomp. 3 = Applic. Stopped 4 = Applic. Running	**Factory Setting:**	0
Properties:	ro		

Description:
It allows the user to view the SoftPLC status. If there are no applications installed, the parameters P910 to P959 will not be shown on the keypad.

If this parameter presents the option 2 (“Incomp. Applic.”), it indicates that the version that has been loaded in the flash memory module is not compatible with the current CFW100 firmware.

In this case, the user needs to recompile its project in the WLP, considering the new CFW100 version and download it. If that is not possible, the upload of this application can be done with the WLP, provided that the application password is known or is not enabled.

P901 – SoftPLC Command

Adjustable Range:	0 = Stop Applic. 1 = Run Applic 2 = Delete Applic.	**Factory Setting:**	0
Properties:	cfg		

Description:
It allows stopping, running or excluding an installed application, but the motor must be disabled in order to do so.

P902 – Scan Cycle Time

Adjustable Range:	0.00 to 9.999 s	**Factory Setting:**	Not applicable
Properties:	ro		

Description:
It consists of the application scanning time. The larger the application, the longer the scanning time.

P910 to P959 – SoftPLC User Parameters

Adjustable Range:	-9999 to 9999	**Factory Setting:**	0
Properties:	-		

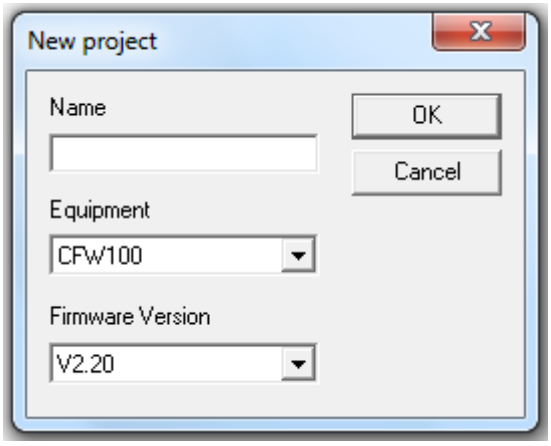
Description:
They consist of use parameters defined by the user via WLP software, as described in item 5.5.

5 SUMMARY OF THE WLP MAIN FUNCTIONS

This chapter brings basic information about the operations done with the WLP software for the use of the CFW100 inverter. More information can be obtained in the help of the WLP software

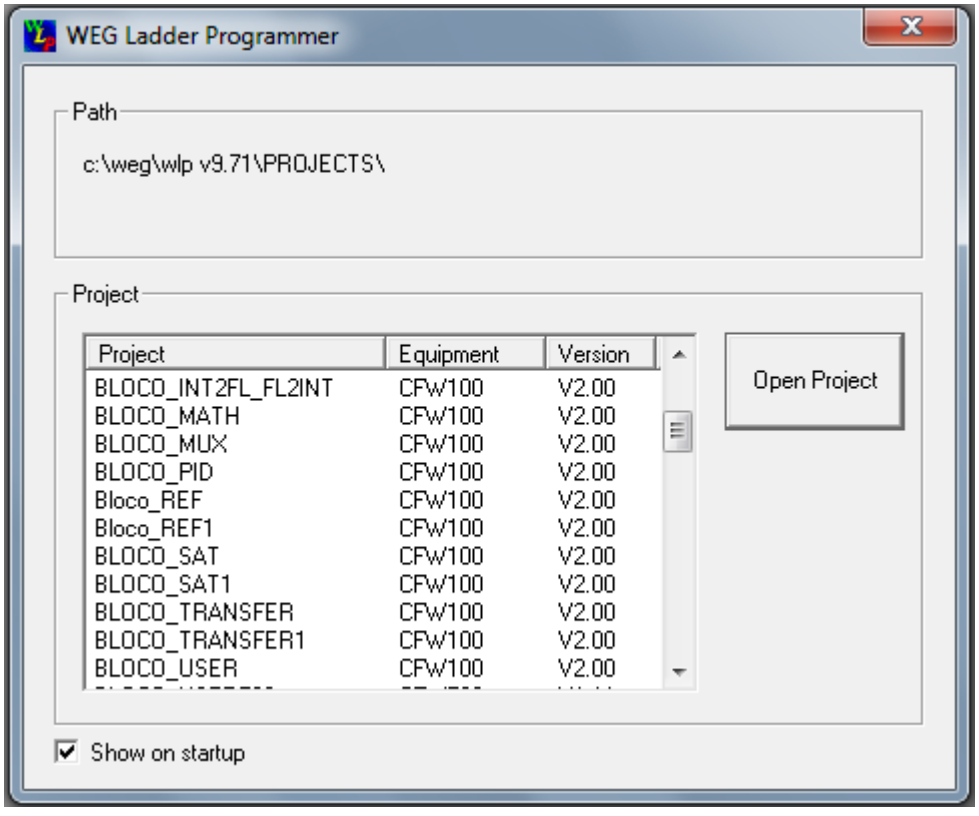
5.1 PROJECT – NEW

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



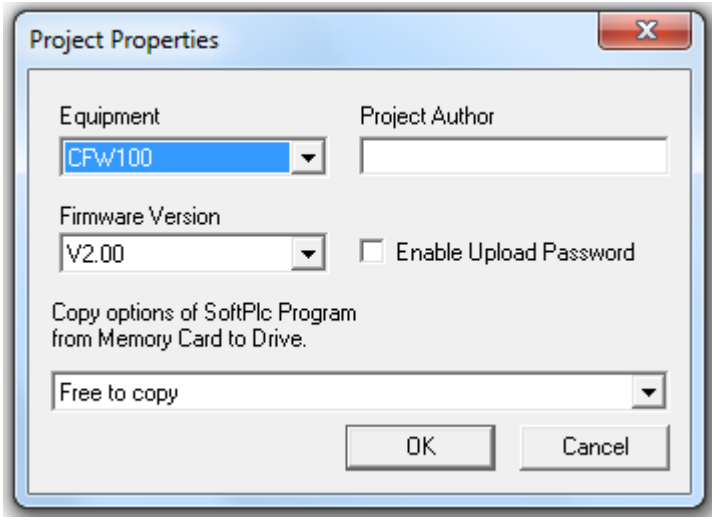
5.2 PROJECT – OPEN

It opens the selected project.



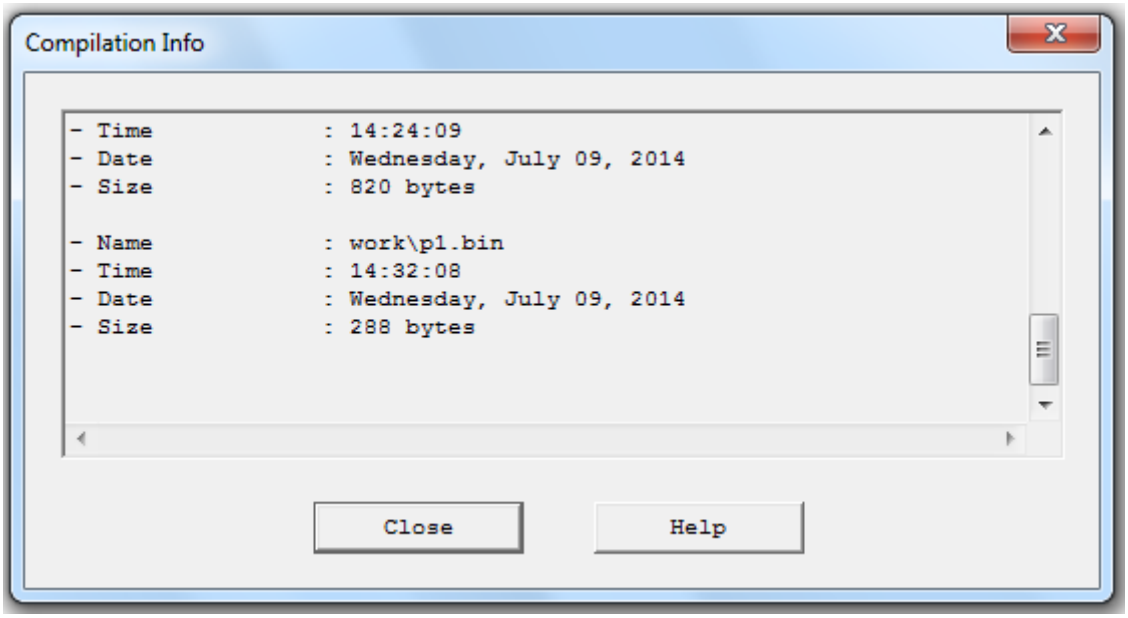
5.3 PROJECT – PROPERTIES

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



5.4 VIEW – COMPILATION INFORMATION

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



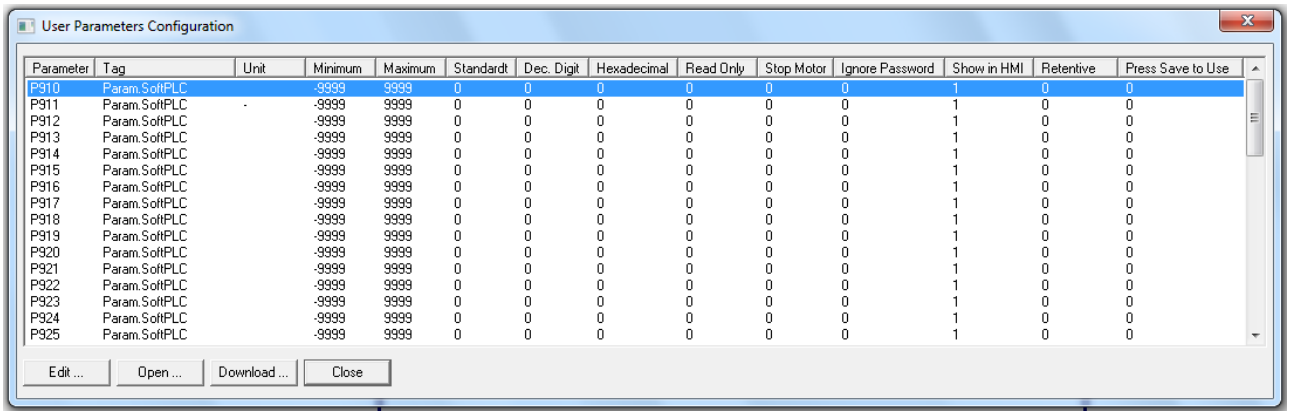
5.5 VIEW – USER PARAMETER CONFIGURATION

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

- Parameter descriptive text (up to 21 characters);
- Unit selection;
- Minimum and maximum limit;
- Default value;
- Number of decimal positions;
- Hexadecimal or regular format;
- Reading or writing only;
- Modification only with a stopped motor or online;
- Ignore the password (allow modification regardless of the inverter password (P000)) or normal;

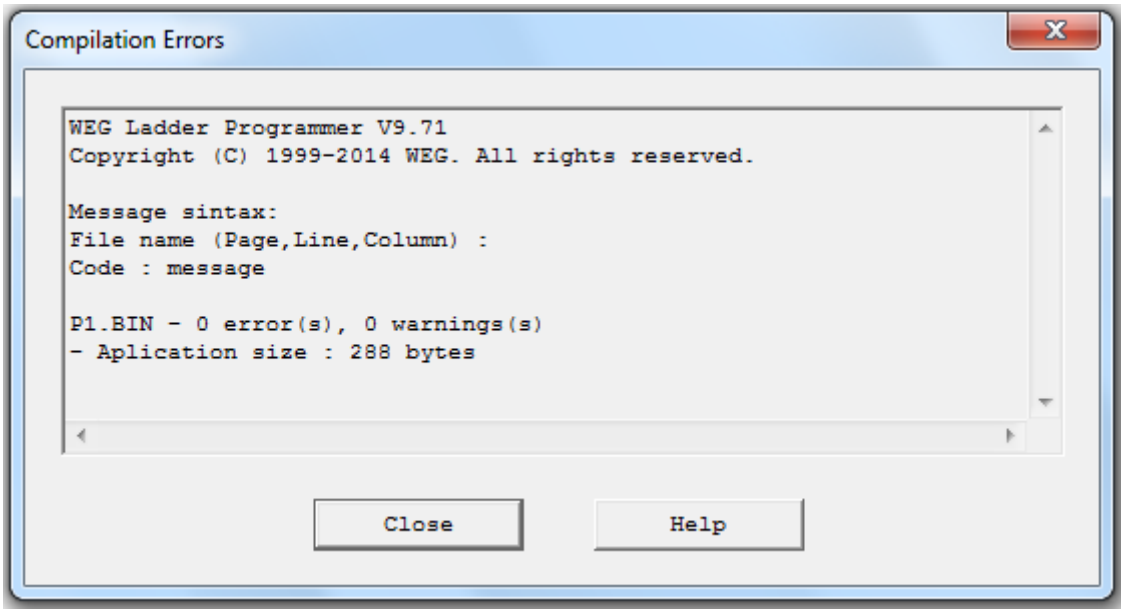
- Display or hide the parameter;
- Allow saving the parameter value (retentive), when it is used in some blocks (PLC, Calculations and Transfers) on power down;
- Configuration parameter which allows modification with the motor running.

Those configurations can be transferred to the CFW100 with the “Download” button.



5.6 CONSTRUCT – COMPILE

It analyzes the application and generates the compiled code for the specified equipment.

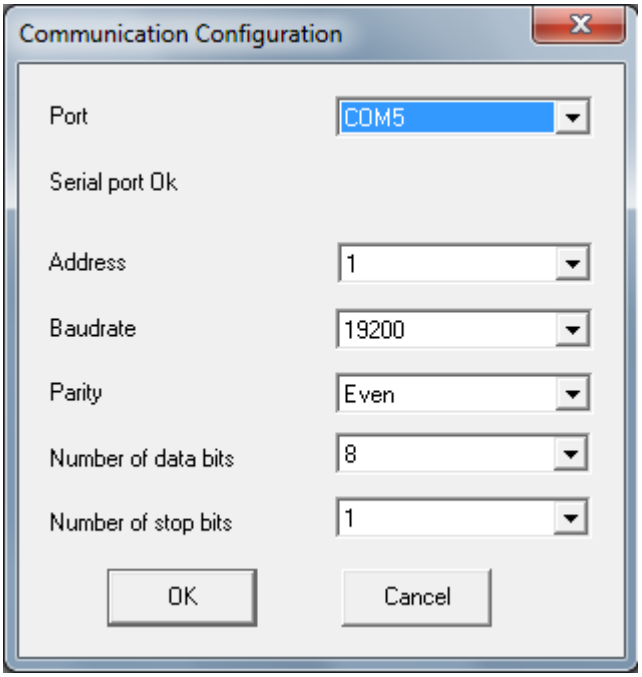


5.7 COMMUNICATION – CONFIGURATION

The Serial port is used for the CFW100.

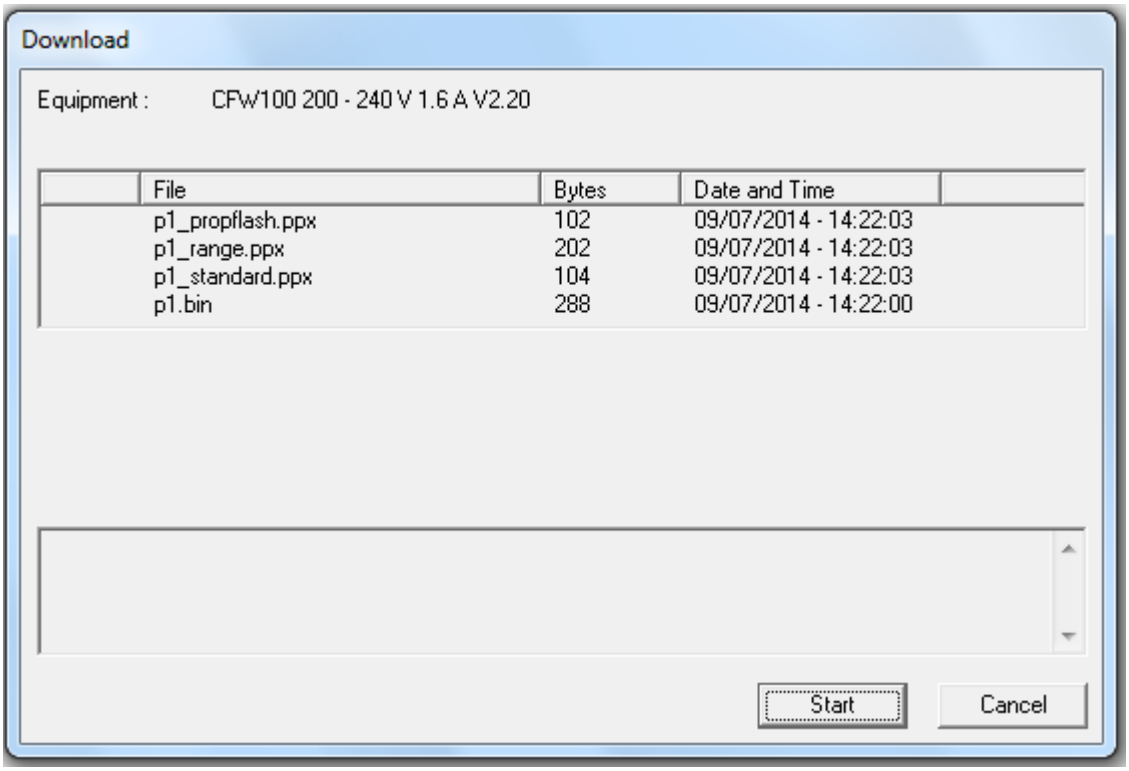
NOTE!

It is recommended to use the CFW100-CRS485 and CFW100-CUSB accessories to establish the communication of the CFW100 with the WLP.



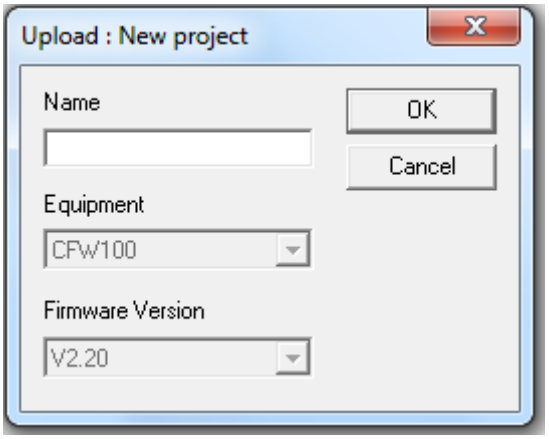
5.8 COMMUNICATION – DOWNLOAD

This command allows downloading the application and/or user parameter configurations to the CFW100.



5.9 COMMUNICATION – UPLOAD

This command makes it possible to upload and open the application installed on the CFW100, provided that the password is valid.



6 ALARMS, FAULTS AND POSSIBLE CAUSES

Table 6.1: *"Alarms", "Faults" and possible causes*

Fault/Alarm	Description	Possible Causes
A702: Inverter Disabled	It occurs when a SoftPLC movement block (REF Block) is not active.	▪ Check if the drive general enable command is active.
A704: Two Movem. Enabled	It occurs when 2 or more movement blocks (REF Block) are enabled simultaneously.	▪ Check the user program logic.
A706: SPLC Ref. Not Prog.	It occurs when a movement block is enabled and the speed reference is not programmed for the SoftPLC	▪ Check the programming of the references in the local and/or remote modes (P221 and P222).
F711: The upload of the SoftPLC application failed	The upload or the SoftPLC application failed.	▪ Fault in the SoftPLC boot by the CPU. ▪ Loaded application incompatible (P900=2) and Command for Application (P901=0);
A712: SoftPLC protected against copy	It occurs when there is an attempt to copy the SoftPLC application protected against copies.	▪ Attempt to copy WLP application protected against copies ("never permit to copy"). ▪ Attempt to copy the WLP from a protected copy ("never permit to copy from a copy");