EURO



User Manual

Version: 0.64 Build ID:20170224 Last change:February 23, 2017.

jPLCPro User Manual Table of Contents

1	Intr	oductior	n to jPLCPro	1
	1.1	Featur	es Overview	1
	1.2	Systen	n Requirements	1
	1.3	Websit	te	1
2	Inst	talling th	ne software	2
	2.1	Startin	g Installation	2
	2.2	First W	/indow	3
	2.3	Licens	e Agreement	3
	2.4	Install	Location	4
	2.5	Installa	ation	5
	2.6	End		6
3	Uni	nstalling	g the software	7
	3.1	Startin	g Uninstallation	7
	3.2	First W	/indow	8
	3.3	Uninst	allation	8
	3.4	End		9
4	Use	er Interfa	ace Features	10
	4.1	Closab	ble Tabs	10
	4.2	Dockin)g	10
	4.2	.1 Doo	cking features	10
	4.2	.2 Doo	cking actions	11
	4	.2.2.1	Docking Shortcuts	11
	4.3	Toolba	ars	12
5	Use	er Interfa	ace	13
	5.1	Work E	Environment	13
	5.2	Drop-d	lown menus	13
	5.2	.1 File	e Menu	14
	5.2	.2 Edi	it Menu	14
	5.2	.3 Pro	oject Menu	15
	5	.2.3.1	Symbols Table	15
	5	.2.3.2	Constants Table	15
	5	.2.3.3	Variables Table	16
	5	.2.3.4	Data Types Tables	17
	5	.2.3.5	Configuration	17
		5.2.3.5	5.1 Project Properties	18
		5.2.3.5	5.2 PLC Settings	18
				Page i

jPLCPro User Manual 5.2.3.5.3 Libraries	10
5.2.4 Tools Menu	
5.2.4.1 C Series Configurator	
5.2.4.2 Options	
5.2.4.2.1 Communication Settings	
5.2.4.2.2 Language Settings	
5.2.4.2.3 Advanced Settings	
5.2.5 Compile Menu	
5.2.6 Online Menu	
5.2.7 View Menu	
5.2.8 Help Menu	
5.3 ToolBars	
5.4 Editors View	
5.5 Project View	
5.6 Console View	
5.7 Properties View	
5.7.1 FB Properties Layout	
5.7.2 IO Elements Layout	
6 Memory Organization	27
6.1 System Types	
6.1.1 BIT Type	27
6.1.2 BYTE Type	27
6.1.3 WORD Type	27
6.1.4 DWORD Type	27
6.1.5 LWORD Type	27
6.1.6 FLOAT Type	27
6.1.7 STRING Type	27
6.1.8 VOID Type	28
6.2 Memory Zones	29
6.2.1 Memory Zone families	29
6.2.2 Memory Zone types:	29
6.2.3 Accessing a Memory Zone	29
6.2.4 Memory Zones List	29
6.3 Additional Memory	
7 FB Programming	31
7.1 Creating FBs	31
7.2 Deleting FBs	
	Page ii

7.3 7.3.1 7.3.2 7.3.3 7.3.4 7.4 7.4.1 7.4.1.1 7.4.2 7.4.3 7.4.4 7.5 7.5.1 7.5.2 7.5.3 7.5.4 7.5.5 7.5.6 7.6 8 BACnet Objects40 9 Compiling......41 10 10.1 Downloading the program43 10.1.1 Write To PLC dialog......43 10.1.2 Monitoring44 10.1.3 FB Monitoring44 10.1.4 Input/Output Monitoring44 10.1.5 Forcing Values......44 10.1.6 Deforcing Values......45 11 11.1.1 46 46 11.1.2 11.1.3 11.1.4 11.1.5 11.1.6

iPLCPro User Manual

jPLCPro Use		
11.1.7	GEQ	48
11.1.8	EQUALS	48
11.1.9	NOT_EQUALS	49
11.1.10	BITWISE_AND	49
11.1.11	BITWISE_OR	
11.2 Arith	metic	50
11.2.1	ADD	
11.2.2	SUBTRACT	50
11.2.3	MULTIPLY	51
11.2.4	DIVIDE	51
11.2.5	ABS	51
11.2.6	SQRT	51
11.2.7	NTH_SQRT	51
11.2.8	POW	52
11.2.9	LOG	52
11.2.10	LOG10	52
11.2.11	SHIFT_LEFT	52
11.2.12	SHIFT_RIGHT	52
11.3 Trig	onometric	53
11.3.1	SIN	53
11.3.2	COS	53
11.3.3	TAN	53
11.3.4	CTAN	54
11.3.5	ARCSIN	54
11.3.6	ARCCOS	54
11.3.7	ARCTG	54
11.3.8	ARCCTG	54
11.4 Time	ers	55
11.4.1	TON	55
11.4.2	TOF	55
11.4.3	ТР	56
11.5 Edg	e Detection	56
11.5.1	EDRE	56
11.5.2	EDFE	57
11.6 Bista	able	57
11.6.1	BESD	57
11.6.2	BERD	58
		Page iv

, 11.7 Stati	stic	59
11.7.1	AVERAGE	59
11.7.2	MINIMUM	59
11.7.3	MAXIMUM	59
11.7.4	MINIMUM_ANALOG	60
11.7.5	MAXIMUM_ANALOG	60
11.8 Sele	ction	60
11.8.1	SEL	61
11.8.2	MUX	61
12 FB Pro	gramming Examples	62
12.1 Bina	ry Input to Binary Output	62
12.1.1	Lessons Learned	63
12.2 OnC	ounter FB	64
12.2.1	Lessons Learned	65
12.3 Dela	yedOut FB	66
12.3.1	Lessons Learned	67
13 Shortc	uts Overview	68

1 Introduction to jPLCPro

In this manual you will find all the information necessary for working with jPLCPro. jPLCPro is integrated work environment at Sun Microsystem **Java**(tm) platform for program development in Functional Block (FB henceforth) language for EPLC. Definition of FB language is according to the IEC61131 standard and it is given in the separate manual.

jPLCPro represents work environment for design, development and translation of FB, and it is supported by all OS which support JAVA virtual machine (Windows XP/NT).

1.1 Features Overview

jPLCPro allows:

- Programming in FB language
- Writing special Functional Blocks in C language
- Downloading executable files to EPLC controllers
- Real-Time online monitoring
- Creation of symbols, constants and variables
- Creation of users and groups
- Creation of new data types
- BACnet objects management
- Configuration of C-Series EPLC controllers
- Configuration of BIOS parameters

1.2 System Requirements

jPLCPro requirements are:

- CPU 1GHz
- RAM 64 MB
- HDD 120MB
- OS All supporting Java Virtual Machine (JVM)
- Java 1.8 or newer

Sun Java jre1.8 or newer version must be installed on the computer. If there is no JVM on the computer, installation of JVM will be offered during the jPLCPro installation. For all the information on JVM click <u>here</u>.

1.3 Website

www.euroicc.com

2 Installing the software

The installation of jPLCPro is started by running the setup file "jPLCPro X.Y – yyyymmdd-xxxx.Setup.exe".

The installation can be stopped at any moment by left-clicking the "Cancel" button.

2.1 Starting Installation

Double-click the setup file.



Left-click on "Next" button to proceed.



2.3 License Agreement

Review the License Agreement and left-click on "Next" button to proceed.



2.4 Install Location

Choose the install directory for jPLCPro. This is done by left-clicking on "Browse" button and selecting the appropriate directory:

送 JPLCPro 5.0.1 - 20160603-1498 Setup	
Choose Install Location	6
Choose the folder in which to install JPLCPro.	
Setup will install JPLCPro in the following folder. To install in a and select another folder. Click Install to start the installation	
Destination Folder	
Program Files (x86)\EuroICC\JPLCPro 5.0.1 - 2016060	Browse
Space required: 640.2MB	
Space available: 237.3GB	
EuroICC	
< Back	Install Cancel

Or by typing the entire path:



jPLCPro User Manual Left-click on "Next" button to proceed to installing jPLCPro

JPLCPro 5.0.1 - 20160603-1498 Setup		
Choose Install Location		*
Choose the folder in which to install JPLCPro.		
Setup will install JPLCPro in the following folde and select another folder. Click Install to start		folder, dick Browse
Destination Folder		
:\Program Files (x86)\EuroICC\JPLCPro 5	.0.1 - 20160603-1498	Browse
	.0.1 - 20160603-1498	Browse
Space required: 640.2MB Space available: 237.3GB	.0.1 - 20160603-1498	Browse
Space required: 640.2MB Space available: 237.3GB	0.1 - 20160603-1498	Browse
Space required: 640.2MB		Browse

2.5 Installation

Wait for the installation to complete. Left-click on "Show Details" button to show details concerning the installation process.

JPLCPro 5.0.1 - 20160603-1498 Setup	L	
Installing		•
Please wait while JPLCPro is being installed.		3
Extract: cygwin.exe 55%		
		7
Show details		
uroICC		

jPLCPro is installed. To run jPLCPro after exiting the installation check the "Run jPLCPro" checkbox by left-clicking on it:

JPLCPro 5.0.1 - 20160603-:	1498 Setup	
2	Completing the JPL Wizard	CPro Setup
	JPLCPro has been installed on you	ur computer.
	Click Finish to close this wizard.	
Q	Run JPLCPro	
www.euroicc.com		
	< Back	Finish Cancel
		194013

Left-click the "Finish" button to finish the installation.

<u> </u>	Completing the JPLC Wizard	Pro Setup
	JPLCPro has been installed on you	r computer.
	Click Finish to close this wizard.	
EURO	Run JPLCPro	
www.euroicc.com		

3 Uninstalling the software

Uninstalling jPLCPro is done by running the uninstall file "uninstall.exe". This file is located in the install directory of jPLCPro.

The uninstallation process is canceled by left-clicking the "Cancel" button.

3.1 Starting Uninstallation

Double-click on the uninstallation file "uninstall.exe";



3.2 First Window

Left-click on the "Uninstall" button to start uninstallation of jPLCPro.

JPLCPro Uninstall		
Uninstall JPLCPro Remove JPLCPro fr		
JPLCPro will be unin	stalled from the following folder. Click Uninsta	Il to start the uninstallation.
Uninstalling from:	C:\Program Files (x86)\EuroICC\JPLCPro 5.	0.1 - 20160603-1498\
EuroICC		Uninstall Cancel

3.3 Uninstallation

Wait for the uninstallation to complete installing jPLCPro. Left-click on "Show Details" button to show details concerning the uninstallation process

👶 JPLCPro Uninstall	
Uninstalling Please wait while JPLCPro is being uninstalled.	3
Delete file: C:\Program Files (x86)\EuroICC\JPLCP	ro 5.0.1 - 20160603-1498\bin\arm-gnu-tool
Show details	
EuroICC	< Back Close Cancel

jPLCPro User Manual **3.4 End**

jPLCPro is uninstalled. To show details of the removal process left-click on the "Show Details" button.

🞐 JPLCPro Uninstall	
Uninstallation Complete Uninstall was completed successfully.	3
Completed	
Show details	
EuroICC	
	< Back Close Cancel

To exit the window and finish the uninstallation left-click on the "Close" button.

💈 JPLCPro Uninstall	
Uninstallation Complete Uninstall was completed successfully.	3
Completed	
Show details	
EuroICC	
< Ba	ck Close Cancel

4 User Interface Features

List of User Interface features in jPLCPro:

- Closable Tabs
- Docking
- Toolbars
- Multi-desktop

4.1 Closable Tabs

jPLCPro tabs are closable. Press on "x" to close a tab.

Ed	lito	or														
		MA	۹IN	_F	в	×	-	B	loc	k_M	Jan	ne	X	ŝ		
Γ																
						·		·	·	·	·					
																•
						•		•	•	•						
	-															

At least one tab will always be open.

4.2 Docking

Application panels are called "Dockables".

- Dockables are positioned by drag-and-drop gesture
- Dockables can be grouped into tabbed panes
- Dockables can be closed

4.2.1 Docking features

Every Dockable has a title bar with a set of common functions (and their corresponding shortcuts).

- Maximize/Restore Target takes all the available space on the desktop
- Detach/Attach Target is separated/added to parent window
- Close Target is closed
- Contextual pop-up menu Target specific options

a Example - IPLCPro Development version							um 000 . 33
Ele Edit Project Tools Compile Online View							
9999964 494 mg •							
12 Projects	Editor					25	Properties -
E 🖶 Example	🛃 MADIL FB 🔹 🛋 Block Name 📧						
Kobijes Kobijes Kock, Name Korane		fil tab	1 ET	111	141		
opt services Constantions Co	100 Parts 11						
Configuration			d de la composición de la comp				
	L IEI I	Console					
100							
22 Monitoring Tree 🗠							
	4						
	Design Definition						
	15- Watch Window	L					* B 20 ×
		e m 📁 🏚 🖪 🗸					
	Tarable	and a state of the state of the		Value			Status

Contextual pop-up menu example:

4	Minimize	Ctrl+F6
Ø	Detach	Ctrl+F5
вя 29	Maximize	Shift+Escape
×	Close	Ctrl+F4

4.2.2 Docking actions

Dockables have their actions palette, depending on their type and context.

Action is performed by doing one of the following:

- Left-clicking on an icon in upper right corner of the target
- Right-clicking on targets tittle bar to open a popup menu. Left-click on an icon
- Using a shortcut

4.2.2.1 Docking Shortcuts

All dockables have the following shortcuts:

lcon	Shortcut	Action
	Ctrl+F6	Minimize
	Ctrl+F6	Restore (after Minimize)

SHIFT+Esc	Maximize
CTRL+Esc	Restore (after Maximize)
CTRL+F5	Detach
CTRL+F6	Attach (after Detach)
CTRL+F4	Close

4.3 Toolbars

jPLCPro tools are hide-able. To toggle the visibility of a toolbar, press the appropriate icon.

Hiding a toolbar example:

🛎 🕼 🗒	1 🗘 🖏 🕫 🗈	1 •	
e 🖫	199 9 of 12	💼 🕴 🎡 🎕	

5 User Interface

jPLCPro consists of several bars and dockables (views).

- Work Environment
- Drop-down menus
- Toolbar menu
- Editor View
- Project View
- Monitoring View
- Console View
- Properties View
- Watch View

5.1 Work Environment

Work Environment is a place where all drop-down menus, bars and dockables are displayed.

TE Example - IPLCPro Development version		And the second		
File Edit Project Tools Compile Onli	ne View Help			
BBBB64 \$\$\$4				
12 Projects	a Idter		12	C Properties
Forgets Forgets Model Forgets For	📥 MAIN_FB 211 📥 Block_Name 16			
Majh FD				
diock_Name				
E @ Constants				
(a) Variables Parameters Users & Groups				
 Users & Groups Ø 8ACnet objects 				
- ⁴⁴ 5 Data Types				
S Data Types D Universal inputs S Configuration				
A consideration	Contraction of the second s			
Monitoring Tree				
	1 m		1	
	Desgn Definition			
	Contract and the second of the second s			n
	Console	医胆管炎	S- Watch Window	※ 開 站 ※
	H.		M + X + 1/6 + 1/9 + 1/1 P P	8 V
			Variable	Value Status
12				

5.2 Drop-down menus

There are eight drop-down menus:

- File Menu
- Edit Menu
- Project Menu
- Tools Menu
- Compile Menu

- Online Menu
- View Menu
- Help Menu

Each drop-down menu is opened by left-clicking the appropriate button.

Fle	Edit Project Tools	Compile Online	en <u>H</u> elp
	New		i i i i i i i i i i i i i i i i i i i
-	Open Project	Ctrl+O	tor
in)	Save Project	Ctrl +S	MAIN FR*
6	Save Project As	Ctrl+Shift+S	Minut 200
-72	Close Project		
	Open project folder		les wid 10 kini sen se
44	Export Function Blocks	Ctrl+Shift+E	ise nation the state see se
	Close	Ctrl+F4	TOT THERE SHE WERE TOTAL TO
	Recent Projects		
	Exit	Ctrl +O	

Fle	Edit Project Tools	Compile Online	View		Help								
	New		画	N	lew Pr	ojec	t		Ctr	I+St	nift	+N	1
0	Open Project	Ctrl+0	1	N	lew M	bdu	e		Ctr	+1			H
圆	Save Project	Ctrl+S	1. M	AI	1 18	X							
6	Save Project As.	Ctrl+Shift+S	a formation	0.00		-	-	-					
	Close Project												
	Open project folder												
٠	Export Function Blocks	Ctrl+Shift+E	13										
	Close	Ctrl+F4	10.7										
	Recent Projects	3											
	Exit	Ctrl+Q											

Some elements of a drop-down menu have sub-menus. These sub-menus are expanded by leftclicking on them.

5.2.1 File Menu

File menu allows manipulation of the project as a whole:

- New Menu Create menus
 - New Project Create new project
 - New Module Create new module (FB or FB-C)
- Open Project open a project
- Save Project save project
- Save Project As save project as a different project from current
- Close Project close project
- Open Project Folder open project location
- Export Function Blocks export FBs
- Close Closes the current project
- Recent Projects Select one of the recent projects to open it
- Exit close jPLCPro

5.2.2 Edit Menu

Edit menu allows the editing of FBs and FB-Cs:

- Undo undo last action (except Function Blocks)
- Redo redo last action
- Cut cut selected content to clipboard
- Copy copy selected content to clipboard
- Paste paste the content from clipboard
- Delete delete selected content
- Clear clear everything from current FB
- Find find FB by name or address

File	Edit	Project	Tools	<u>C</u> ompile	Online
	New				•
œ	Open F	Project		Ctrl	ю
R	Save P	⊦S			
8.	Save P	roject As		Ctrl-	+Shift+S
	Close F	Project			
	Open p	project fold	ler		
₽	Export	Function E	Blocks	Ctrl	+Shift+E
	Close			Ctrl	⊦F4
	Recent	t Projects			•
	Exit			Ctrl	ŀQ
	Exit			Ctrl-	ŀQ



jPLCPro User Manual **5.2.3 Project Menu**

Project menu allows the opening of various tables and configuration of project:

- Symbols Table
- Constants Table
- Variables Table
- Data Types Table
- Configuration

5.2.3.1 Symbols Table

Symbols Table contains symbols and allows editing symbols.

A symbol represents a certain part of memory. It consists of:

- Name String by which the symbol is known
- Address Memory Zone and Index in it
- Init Value Initial Value of the symbol
- Comment

Symbols Table allows:

- Browsing symbols
- Creating new symbols
- Changing existing symbols
- Removing existing symbols

There are two types of Symbols tables, User Symbols Table and System Symbol Table.

User Symbols Table can be edited. System Symbols Table is a table defined by project settings, and cannot be edited.

Editor	Editor and a second				
📲 MAIN_FB 💥 🛷 User Symbols* 💥 🛛 🛷 System Symbols 💥					
+ -	🦻 🛫 Filter: All 🔻 (
	Symbol name	Address	Init Value	Comment	
1	RHC_roomNum	MB100		Room Number from Card Reader	
2	RHC_cardType	MB101		Card Type (guest, maid,) from Card R	
3	RHC_startDate	MB102		Card Date from Card Reader	
4	RHC_endDate	MB103		Expiry Date from Card Reader	
5	CRH_roomNum	MB104		Room Number from Card Holder	
6	CRH_cardType	MB105		Card Type (guest, maid) from Card Ra	
7	CRH_card	мхо		Card in Range from Card Reader	

Note: Symbols must have unique names.

5.2.3.2 Constants Table

Constants Table contains constants and allows editing constants.

Proj	ect	Tools	<u>C</u> ompile	<u>O</u> nline	Viev
۰	Sy	mbols	(Ctrl+Shift+	M
٩	Co	nstants	C	Ctrl+Shift+	-C
(× <u></u> ≇	Var	riables	C	Ctrl+Shift+	-V
4\$	Da	ta Types	C	Ctrl+Shift+	Ð
≪	Co	nfiguratio	on (Ctrl+Shift+	P

A constant is a value that can be used by calling its name. Constants value cannot be changed during runtime. It consists of:

- Name String by which the constant is known
- Type Type of constant
- Value Value of constant.
- Comment

Constants Table allows:

- Browsing constants
- Creating new constants
- Changing existing constants
- Removing existing constants

Similar to Symbols Tables, there are two kinds of Constants Table: User Constants Table and System Constants Table.

ditor						
着 MAIN_FB 🕺 🥔 System Symbols 💥 💣 System Constants 💥 💣 User Constants 🕱						
	Name	Туре	Value	Comment		
1	FLASH_W_P1	BYTE	1	For FlashWriteCmd: Command for storin		
2	FLASH_W_P2	BYTE	2	For FlashWriteCmd: Command for storin		
3	FLASH_W_P3	BYTE	3	For FlashWriteCmd: Command for storin		
4	FLASH_W_P4	BYTE	4	For FlashWriteCmd: Command for storin		
5	FLASH_W_P5	BYTE	5	For FlashWriteCmd: Command for storin		
6	FLASH_W_P6	BYTE	6	For FlashWriteCmd: Command for storin		
7	FLASH_W_P7	BYTE	7	For FlashWriteCmd: Command for storin		
8	FLASH_W_P8	BYTE	8	For FlashWriteCmd: Command for storin		
9	SYS_PAGE	BYTE	62	For FlashWriteCmd: Command for storin		
10	FLASH_W_ERR	BYTE	100	For FlashWriteCmd: There was an error		
11	NO_COMM	BYTE	0	For Com_x_Type: Communication port di		
12	BACnet_MSTP	BYTE	1	For Com_x_Type: BACnet protocol selec		
13	Modbus_RTU	BYTE	2	For Com_x_Type: Modbus RTU protocol		
14	Modbus ASCII	BYTE	3	For Com x Type: Modbus ASCII protoc		

User Constants Table can be edited. System Constants Table is a table defined by project settings, and cannot be edited.

Note: Constants must have unique names.

5.2.3.3 Variables Table

Variables Table contains variables and allows editing variables.

A variable is a value that has a name and type. It can be changed during runtime. Values consist of the same elements as constants:

- Name String by which the variable is known
- Type Type of variable
- Value Value of variable
- Comment

- Browsing variables
- Creating new variables
- Changing existing variables
- Removing existing variables

Note: Variables must have unique names.

5.2.3.4 Data Types Tables

Data Types Tables contain data types and allow editing variables.

Data type can be:

- Arrays array of a certain type and length
- Enumerated array of integers where a position is marked by a certain name
- Structures complex structures that contain one or more other data types

For each data type there is a corresponding Data Type Table.

Data Types Tables allow:

- Browsing data types
- Creating new data types
- Changing existing data types
- Removing existing data types

Note: Data types must have unique names.

5.2.3.5 Configuration

Configuration allows the configuration of the project. It has three main sections:

- Project Properties
- PLC Settings
- Libraries

5 User	Interface
--------	-----------

Version: 10 Description: No description PLC settings PLC Device: C2 Series • Device type: C2.FCC.02 Compler settings P Path: C:\Program Files (v86)\EuroICCUPLCPro 5.0.1 - 20160613-1501\bin\arm-gnu-toolchain_4_9_3\bin Compler options:			
LC settings PLC Device: C2 Series PLC Device: C2 Series Device type: C2.FCC.02 ompiler settings Path: C:\Program Files (x86)\EuroICC\JPLCPro 5.0.1 - 20160613-1501\bin\arm-gnu-toolchain_4_9_3\bin Browse Compiler options: Linker options: Linker options: System libraries User libraries	Version: 1.0		
PLC Device: C2 Series Device type: C2.FCC.02 ompiler settings Path: C:\Program Files (x86)\EuroICC\JPLCPro 5.0.1 - 20160613-1501\bin\arm-gnu-tooldhain_4_9_3\bin Browse Compiler options: Linker options: System libraries User libraries Add	Description: No description		
PLC Device: C2 Series Device type: C2.FCC.02 path: C:\Program Files (x86)\EuroICC\\PLCPro 5.0.1 - 20160613-1501\bin\arm-gnu-toolchain_4_9_3\bin Browse Compiler options: Linker options: System libraries User libraries Add	.C settings		
Device type: C2.FCC.02		•	
ompiler settings Path: [C:\Program Files (x86)\EuroICC\JPLCPro 5.0.1 - 20160613-1501\bin\arm-gnu-toolchain_4_9_3\bin Browse Compiler options: Linker options: Draries System libraries User libraries Add			
Path: [C:\Program Files (x86)\EuroICC\JPLCPro 5.0.1 - 20160613-1501\bin\arm-gnu-toolchain_4_9_3\bin Browse Compiler options: Linker options: braries System libraries User libraries Add			
Compiler options:		95)\EuroTCC\101CBro 5.0.120160612-1501\bio\arm.opu.toolchain 4.9.2	bia Browse
inker options: praries System libraries User libraries Add		50/E0101CC (PECPT0 5.0.1 - 20100015-1301 (pin (ann-gina-toolchain_4_9_5)	DIOWSE
System libraries User libraries Add			
System libraries User libraries Add			
System libraries User libraries Add			
Add			
Kemove			
			Add

5.2.3.5.1 Project Properties

Project Properties are project version and project description.

Both are defined as a text, so any value is acceptable.

5.2.3.5.2 PLC Settings

PLC Settings are PLC Device and Device Type. Both are chosen from a drop-down menu.

PLC Device is a family of EPLCs, such as "C Series".

Device Type is an EPLC device, such as "GRU21".

These settings define the compiler, method of communication and so on. They should be set with appropriate device type.

jPLCPro User Manual **5.2.3.5.3 Libraries**

This section allows management of libraries. There are "System libraries" which come with jPLCPro and "User libraries" that users create with "Export Function Block".

Libraries contain FBs which can be used in FB programming as any other.

Libraries are added by left-clicking the "Add" button and removed by selecting the library by leftclicking on its name and left-clicking the "Remove" button.

5.2.4 Tools Menu

Tool menu contains tools for various devices and options.

5.2.4.1 C Series Configurator

Used for configuring devices from C Series, such as GRU21.

See C Series Configurator Documentation.

5.2.4.2 Options

To save changes left-click the "Ok" button. To discard changes left-click the "Cancel" button.

Options allow changing various settings of a jPLCPro and there are three main sections:

- Communication Settings
- Language Settings
- Advanced Settings

5.2.4.2.1 Communication Settings

In Communication Settings communication parameters are set.

Depending on the device some parameters will be disabled.

Communication Settings allow editing:

- Protocol
- Communication Port
- Baud Rate
- Communication Type
- IP Address
- PLC Address
- Send Time

Communication se	ttings 🚺 Language 🔯 Advanced	
Protocol:	СОМ	
Communication port:	COM1	▼ Refresh
Baud rate:	115200	
Communication type:	8E1 💌	
IP address:	10.0.0.1	
PLC address:	1 🚖	
Send time:	2 🐥 x 50ms (2 - 99)	
end time:	2 🚖 x 50ms (2 - 99)	

Tool	s	Compile	<u>O</u> nline	View	<u>H</u> elp
	C	C Series Cor	nfigurator		
୍ତି	c	Options		(Ctrl+T

jPLCPro User Manual **5.2.4.2.2 Language Settings**

Language is selected by choosing the wanted language.

The default language is English.

Communication settings Language Advanced
Atmel SAM-BA install directory: Browse
OK Cancel

5.2.4.2.3 Advanced Settings

Advanced settings allow setting some of the advanced features of jPLCPro:

- Autosave project before compiling
- Choosing the Atmel SAM-BA directory

5.2.5 Compile Menu

Compile Menu allows compiling features:

<u>C</u> ompile		<u>O</u> nline	View	<u>H</u> elp	
÷	Compile			Ctrl+R	
÷	Compile & Run		n	Ctrl+Shift+R	

- Compile Compiles FB program
- Compile and Run Compiles FB program, downloads it to the device and runs it automatically

Compilation progress and status are displayed in the Console Dockable. jPLCPro needs to be connected to a device for "Compile and Run" feature to run successfully. Depending of the device type some options are unsupported and are disabled.

5.2.6 Online Menu

Online menu allows access to features that include some sort of communication between jPLCPro and a PLC.

- Connect to PLC Connects to a device.
- Disconnect Disconnects from a device
- Write to PLC Write to PLC dialog is presented
- Upgrade Firmware
- Monitor On Turns on monitoring
 - Device must be connected
- Force Forces a Value
 - Monitor must be On
- Deforce Deforces a value
 - Monitor must be On
- Force On Forces a Bit Value
 - o Monitor must be On
 - Force Off Deforces a Bit Value
 - o Monitor must be On
- Synchronize device date and time
- IO Config
- Write IO Config
- Update IO Config
- BIOS Parameters
- IO Parameters
- Slot Parameters
- Device Info

5.2.7 View Menu

View menu allows access to Workspace look features.

- Docking View opens the "Docking Configuration" window.
 - This window allows users to choose which Dockable will be shown.
- ToolBars opens the "ToolBars Configuration" window.
 - o This window allows users to choose which ToolBars will be shown
- Status Bar
 - Allows users to choose whether the Status Bar should be shown. To toggle between states (showing/hidden) press this menu item.
- Reset Workspace
 - Restores default Workspace configuration (Docking Configuration, ToolBar Configuration and Status Bar)

5.2.8 Help Menu

In this menu a user can find the jPLCPro User Manual and some information considering the application.

- Help Contents
 - o Opens a pdf file of jPLCPro User Manual
- About jPLCPro
 - Opens a "About jPLCPro" window that displays various technical information about the software

Upgrade firmware Monitor On Ctrl+M Force... Ctrl+E12 Monitor On Ctrl+F11 Force on F12 Force off F11 Synchronize device date and time IO Config... Write IO Config Update IO Config BIOS Parameters... Ctrl+B IO Parameters... Slot Parameters... Device Info...

Online View Help Write To PLC...

View	v <u>H</u> elp
6	Docking View
***	ToolBars
✓	Status Bar
	Reset Workspace

Help		
	Help Contents	F1
	About JPlcPro	

5	User	Interface
-		

Ctrl+W

5.3 ToolBars

ToolBars are a way to complete certain actions faster. Instead of going through the menus and searching for a wanted action, users can simply one-click on an icon and get the desired effect.

ToolBars are separated by a separator that also functions as hide/show button. Separator on the left of a ToolBar group hides/shows that ToolBar group.

ToolBar groups are logical groups of actions that are grouped based on context.

Mouse-hovering over an icon of a ToolBar will give a brief description of that icons action.

5.4 Editors View

This Dockable is reserved for FB and FB-C editors.

FB editors are in a form of a diagram

Editor							R R B N
🔀 Configuration 🛛	🛷 User Symbols 🕅	ᡖ Klima 🔀 📫 Pristup	×				
		kartice					
			EQUALS	1			
			LQUALS				
		<u> </u>				<u></u>	
		RHC_cardType	· · · · a	out	gostKartica	P	
							
	() () () () () () () () () ()	2	b				
		<u> </u>					
				2			·
			EQUALS				1
			. <u> </u>	out	sobKartica	[−] 4 [−]	
						<u> </u>	
		· · · · · · · · · · · · · · · · · · ·	 b				
		4					
							•
							•
	Broj						
	sobe						
	sobe)					•
			FOUND	0			
			EQUALS				
		<u></u> .			· · · · · · · · ·		
		RHC_roomNum	а	out		5	
		<u> .</u> /.					
		· · · · · · · · · · · · · · · · · · ·	<u> </u>				
		roomNum					
							-
•							
Design Definition							

2 Example - IPLCPro Development version				- 10 - 22 -2
File Edit Project Tools Compile Online Yew	Beb			
·				
12 Projects	Editor		@ Properties	-0
E - E Skample E - E Functional Blocks	🔥 MARI, FB 🔄 🍨 Code, Blod. 🖉 🍓 Blod., Name			
Alder, Im Alder, Im	<pre>10 +****** = void f_Code_Black(Code_Black *Code_Black) (1 = void f_Code_Black *Code_Black *Code_Black) (1 = void f_Code_Black *Code_Black *Code</pre>	* 		
E Monitolog Tex	C Circlon (Silfiniti)			
	C comple	K So Watch Window		-TT 20 TF
	4		a 💠	
	Lotic	Variable	Value	n/a/?

5.5 **Project View**

Project View is a Dockable that represents hierarchy of the project and allows editing the elements of the project.

Every element in the Project View can be right-clicked for additional actions such as adding a new FB or deleting an existing one.

Every project contains the following sections:

- Functional Blocks
- Symbols
- Constants
- Variables
- Parameters
- Users and Groups
- BACnet Objects
- Data Types
- Universal inputs
- Configuration

Sections are expanded with left-clicking the "+" icon and are folded by left-clicking the "-" button.

Elements of sections are opened with double-clicking the desired element. All elements are opened in the Edit View as a new tab.



Console View displays console information. Console information is all the information that is related to compiling and distribution of the project.

Console	- D - R.7	×
14		
In file included from source_project/startup_sam3s.c:46:0: ./include_project/compiler.h:153:0: warning: "always_inline" redefined # definealways_inlineattribute((always_inline)) inline		-
In file included from c:\program files (x86)\euroicc\jplcpro 5.0.1 - 20160613-1501\bin\arm-gnu-toolchain_4_9_3\arm-none-eabi\include\s from ./include_project/compiler.h:65, from source_project/startup_sam3s.c:46:		
c:\program files (x86)\euroicc\jplcpro 5.0.1 - 20160613-1501\bin\arm-gnu-toolchain_4_9_3\arm-none-eabi\include\sys\cdefs.h:359:0: no location of the previous definition #definealways_inlineattribute((always_inline)) ^	te: this is the	
source_project/startup_sam3s.c: In function 'Init_BACnet': source_project/startup_sam3s.c:122:2: warning: implicit declaration of function 'initiate_BACnet_obj_init_struct_pt' [-Wimplicit-function-de initiate_BACnet_obj_init_struct_pt(); ^	eclaration]	
C:/Program\Files\\(x86\)/EuroICC/JPLCPro\ 5.0.1\ -\ 20160613-1501/bin/arm-gnu-toolchain_4_9_3/bin/arm-none-eabi-gcc ./source_project/startup_sam3s.o./source_project/main_fb.o./source_project/PLCmem.o./UPG_BACnet/BACnet_functions.o ./UPG_BACnet/BACnet_objects.o./source_project/upg_init.o./source_project/init_func_blocks.o./source_project/constants.o./source_p ./source_project/variables.o -mthumb -mcpu=cortex-m3 -nostartfiles -Xlinker -OUPG_SAM3S.elf -Xlinker -M -Xlinker -Map=UPG_SAM3S.m qc-sections -Xlinker -Tflash_sam3s4_upg.ld -Im		0
C:/Program\Files\\(x86\)/EuroICC/JPLCPro\5.0.1\-\20160613-1501/bin/arm-gnu-toolchain_4_9_3/bin/arm-none-eabi-objcopy UPG_SAU3S.hex	M35.elf -0 ihex	
C:/Program\Files\\(x86\)/EuroICC/JPLCPro\ 5.0.1\ -\ 20160613-1501/bin/arm-gnu-toolchain_4_9_3/bin/arm-none-eabi-objcopy UPG_SAU3S.bin UPG_SAU3S.bin Build finished.	M35.elf -0 binar	Y
Coping result to dist folder Coping completed.		
Creating header Header created.		
Source code produced in folder: C:\Users\ilke\Documents\JPLCProProjects\Nikola\jPLCPro_Zahra\FNCupg\build Code produced in folder: C:\Users\ilke\Documents\JPLCProProjects\Nikola\jPLCPro_Zahra\FNCupg\dist End compiling		
1		•

To clear the console click on "Clear" icon.

5.7 Properties View

Properties View displays information about the currently selected element in the Editor View. To select an element simply left-click on desired element. As there can only be two kinds of elements, FB and IO elements, there are only two layouts of displaying information.

Additionally Properties View allows setting the priority of FB execution and display of Execution Control.

5.7.1 FB Properties Layout

Displayed FB properties are:

- Name and Instance (in form of Name_InstanceNumber)
- Priority
- Execution Control

<image>

Users can change Priority and Execution Control by changing the corresponding value or checking the box.

5.7.2 IO Elements Layout

IO Elements are Input Elements and Output Elements. They are mostly the same, as the only difference is that Output Elements have the Priority field. Other properties are as follows:

- Name
- Type
- Filter By Type
- Filter By Data Type
- Priority (Outputs only)

<complex-block>

Name can be set from Properties View. It can be written manually or selected from a drop-down list. IO Elements names follow the rules that would be explained latter.

Filters control what IO Elements would be available in the name drop-down list.

6 Memory Organization

Memory that is at user programs disposal is the same on every CPLC controller. The memory locations can be of various types and memory is divided into several **ZONES**. System Types

6 Memory Organization

Memory that is at user programs disposal is the same on every CPLC controller. The memory locations can be of various types and memory is divided into several zones.

6.1 System Types

System types are types that can be used for inputs, outputs, constants, symbols, variables or building blocks for complex data types:

- BIT
- BYTE
- WORD
- DWORD
- LWORD
- FLOAT
- STRING
- VOID

All types, except STRING and VOID are Numerical types. All numerical types, except FLOAT are Integer types (whole numbers). BIT type is sometimes referred as Boolean type.

6.1.1 BIT Type

BIT type represents one bit. Bit value can be either zero or one (0 or 1).

6.1.2 BYTE Type

BYTE type consists of 8 bits. It is a signed integer, which means it can have either positive, negative or zero integer value. Its range is from -128 to 127.

6.1.3 WORD Type

WORD type consists of 2 bytes or 16 bits. It is a signed integer. Its range is from -32768 to 32767.

6.1.4 DWORD Type

DWORD type consists of 4 bytes or 32 bits. It is a signed integer. Its range is from -2147438648 to 2147438647.

6.1.5 LWORD Type

LWORD type consists of 8 bytes or 64 bits. It is a signed integer. Its range is from -4294967296 to 4294967295.

6.1.6 FLOAT Type

FLOAT type represents real values. Its range is from -3.4E+38 to 3.4E+38

6.1.7 STRING Type

STRING type is an array of 32 bytes (or characters).

6 Memory Organization

Memory that is at user programs disposal is the same on every CPLC controller. The memory locations can be of various types and memory is divided into several **ZONES.** System Types

6.1.8 VOID Type

VOID type consists of 4 bytes or 32 bits. It is used for pointers.

6.2 Memory Zones

A memory zone is a part of memory that can be accessed through FB diagrams and FB code. It is represented as an array of a certain type. Each memory zone has a family and a type.

They are accessed by combining the shortcut name for the memory zone family and the shortcut name for the memory zone type.

Accessing the memory zones will be covered after memory zone families and memory zone types.

6.2.1 Memory Zone families

- System device system registers shortcut name s
- Input used for device inputs
- Output used for device outputs
- Memory used for user memory

6.2.2 Memory Zone types:

- BIT shortcut name X
- BYTE shortcut name B
- WORD shortcut name W
- DWORD shortcut name D
- LWORD shortcut name L

BIT Memory Zone is separate from others. BYTE, WORD, DWORD and LWORD Memory Zones overlap.

WORD type contains two BYTE types. For example, WORD type at index 0 contains BYTE types at indexes 0 and 1. WORD type at index 2 contains BYTE types at indexes 4 and 5.

Similarly, DWORD type contains two WORD types and four BYTE types.

LWORD contains two DWORD types, four WORD types and eight BYTE types.

6.2.3 Accessing a Memory Zone

Accessing a memory zone is done by writing ${\tt FT}$ in functional blocks and ${\tt FT[I]}$ in C code where:

- F is a memory zone family shortcut name
- T is a memory zone type shortcut name
- I is an index

For example, to access a bit output at index 25 write: QX[25]

6.2.4 Memory Zones List

The complete memory zone list is presented in the following table:

Name	Function	Туре	Length	Note
SX	System bits	BIT	64	
SB	System byte registers	BYTE	192	Overlapped with SW, SD and SL

- shortcut name I
 shortcut name Q
- shortcut name M

J. 201				
SW	System word registers	WORD	96	Overlapped with SB, SD and SL
SD	System double word registers	DWORD	48	Overlapped with SB, SW and SL
SL	System double word registers	LWORD	24	Overlapped with SB, SW and SD
IX	Binary inputs	BIT	64	
IB	Byte inputs	BYTE	64	Overlapped with IW, ID and IL
IW	Word inputs	WORD	32	Overlapped with IB, ID and IL
ID	Double word inputs	DWORD	16	Overlapped with IB, IW and IL
IL	Long word inputs	LWORD	8	Overlapped with IB, IW and ID
QX	Binary outputs	BIT	64	
QB	Byte outputs	BYTE	64	Overlapped with QW , QD and QL
QW	Word outputs	WORD	32	Overlapped with QB, QD and QL
QD	Double word outputs	DWORD	16	Overlapped with QB, QW and QL
QL	Long word outputs	LWORD	8	Overlapped with QB, QB and QD
MX	Memory bits	BIT	256	
MB	Memory bytes	BYTE	2048	Overlapped with MW, MD and ML
MW	Memory words	WORD	1024	Overlapped with MB, MD and ML
MD	Memory double words	DWORD	512	Overlapped with MB, MW and ML
ML	Memory long words	LWORD	256	Overlapped with MB, MW and MD
L				

6.3 Additional Memory

Beside mentioned memory zones there is an additional zone that is primarily used for variables from C code. This zone is limited so care must be taken that any long structures or arrays are defined as variables in some user zone (in most cases M zone). One additional characteristics of this zone is that it cannot be accessed through any communication channel.
7 FB Programming

All of the programming is done by using Functional Blocks.

The default (non-removable) FB is MAIN_FB. This FB is used as a starting point of the program.

There are two types of FB, FB Diagrams (FB) and FB Code (FB-C). Each will be explained in more detail later.

Every FB has a "Definition" which defines inputs, outputs and similar elements.

In the "Project View" are the bodies of FBs, which define how and what will FB do.

When a FB is used in another FB, an instance of FB is created. An instance of FB will follow the programming of FB, but different instances can have different values on same elements (for example on the same parameter).

7.1 Creating FBs

Creating FBs is done from the "File Menu", from the "Project View" or by using a shortcut.

To create a FB from "File Menu" open file menu with left-click, left-click on "New" and then leftclick on "New Module".



To create a FB from "Project View" right-click on "Functional Blocks" and left-click on "New Module".



The shortcut for creating a new FB is "CTRL+N".

When creating a FB a FB creation window will appear.

Type in the desired name for a new FB in the appropriate field.

New Module		X
Create a new module		
Enter a module name:		
NewModule	FB	•
		8
	OK Cancel	

Select the desired FB type from a drop-down menu.

New Module		×
Create a new module		8
Enter a module name:		
NewModule		FB 💌
		FB FB-C
	ОК	Cancel

Left-click the ${\tt OK}$ button to create new FB or left-click the ${\tt Cancel}$ button to exit the FB creation window.

7.2 Deleting FBs

Deleting FBs is done from the Project View dockable.

Navigate to the desired FB in the Project View. Right-click on the FB name and left-click on the Delete button in the drop-down menu. Or press the Del button on keyboard.

	ماييلم	
···· / Ibraries Ibraries Symbols Ibraries Constants Ibraries Constants Ibraries Ibraries Ibraries Symbols Ibraries Ibrari	æ	Open Open all Open all Functional Blocks Save Reload
Universal in		Rename
🛃 Monitoring Tree	×	Delete
		Close
		Close all
		Close Project

7.3 FB Definition

FB Definition contains:

- Inputs
- Outputs
- Parameters
- Locals

To display FB Definition that FB must be open in Editor View. To display FB definition left-click on the Definition tab in the bottom left corner of the FB tab.

MAIN_FI	3 🔀	Ne	wMo	dule	2	5																									
		 																							_	_	_	_		 _	_
										• •										• •											
	• •																• •			• •	•						• •				
	• •									• •							• •			• •	•						• •				
	• •				• •			• •	•	• •			• •				• •			• •			• •				• •			• •	
	• •									• •							• •										• •				
	• •																• •										• •	, ·			
		 		÷		÷	÷				÷	÷		÷								Ċ							Ċ		

Editor			5.5 19
🔒 MAIN_FB	×	ᡖ NewModule 🔀	
Inputs Outputs	#	Inputs	Туре
Parameters	0		
Locals	1		
	2		
	3		
	4		
	5 6		
	6		
	7		
	8		
	9		
	10		
	11		
	12		
	13		
	14		
	15		
	16		
	17		
	18		Type
	19		
Design Defi			

Design Definition

Maximum number of any element type is twenty.

To add an element type in the first free row.

To delete an element erase the information from the appropriate row.

To change an element type in the changes in the appropriate row.

7.3.1 Inputs

FB Inputs are the entry point for data in a FB.

They appear on the left side of the graphical interpretation of the FB.

Every input has:

- Name name of the input
- Type type of the input

7.3.2 Outputs

FB Outputs are the exit point for data in a FB

They appear on the right side of the graphical interpretation of the FB.

Every output has:

- Name name of the output
- Type type of the output

7.3.3 Parameters

FB Parameters are parameters of a FB. For each instance of a FB different values of parameters can be assigned.

- Name name of parameter
- Type type of parameter
- Init Value initial value
- Max maximum value
- Min minimum value
- Inc increment

7.3.4 Locals

FB Locals are local variables of a FB. Each instance of a FB can have different values of locals.

Every local has:

- Name name of local
- Type type of local
- Init Value initial value

7.4 FB Diagram

FB Diagrams are a graphical way to program a FB.

The three main parts of every FB diagram are:

- FB instances
- Inputs/Outputs
- Connections

FB instances and Outputs have an Execution Priority. Execution Priority defines execution order in a FB.

Connections connect FB instances, inputs and outputs.

The algorithm of every FB diagram is as follows:

- All inputs are read
- FB instances and outputs are calculated, in the order of the Execution Priority

Note: Usually FB instances will have a higher priority than the output elements.

7.4.1 Adding elements

To add an FB instance, input or output:

Right-click on the desired location and left-click on the desired element.

- System FBs are in the first group of elements.
- User defined FBs are under the "Custom" sub-menu.
- Inputs, outputs and comments are in the third group of elements.

7.4.1.1 Setting Input/Output Elements

When an input or output is added to a diagram it is empty.

Setting what is an input/output element is done from Properties View, as explained before.

Inputs can be FB inputs, FB outputs, variables, constants, symbols or a member of a memory zone.

Outputs can be FB inputs, FB outputs, variables, symbols or a member of a memory zone.

Element inputs can be FB outputs, and vice versa. This seems counter-intuitive, but it is needed. Sometimes calculations depend not only from FB inputs, but from FB outputs also.

7.4.2 Removing elements

To remove an element from FB

Right-click on the element, and left-click on the "Delete" button



Or press the "Delete" button on the keyboard.

When an element is deleted its connections are also deleted.

7.4.3 Connecting elements

To connect elements

Left-click on the desired FB input/output or input/output



Move cursor to the other side of desired connection. Dashed line is following cursor.



Left-click on the other side of the desired connection, again on the desired FB input/output or input/output

<u> </u>					-	-	· ·	<u>`</u>					
		in	рu	ıt				ŀ	•	•		output	5
											_		_

The elements are now connected.



When output of one block is to be connected to more than one input. Each line must be connected completely. It is not possible to connect one line to the middle of another.

FB instances are the part of FB diagrams where all the calculating is done. As explained before, they have inputs and outputs.

In Editor View the name of the FB instance will be without the instance number. In Properties View this number will be displayed in format FBName_Number.

7.5 FB C Code

FB Code is a method of textual (standard) programming. It is done in the standard C programming language.

Writing the code is done in "Editor View" in the appropriate tab.

All of the FB definition elements are accessible through code, as are all the symbols, variables and constants.

There is package of system libraries that can be used.

There is no Execution Order, as the code itself determines it.

Editor	R.A. B'M
📸 Test	×
1⊡typedef struct {	
2 WORD input;	
3 WORD output;	
4 WORD parameter;	
5 WORD local;	
6 }Test;	
8	
9⊡ static void f_Test(Test *Test) {	
10 pTest->parameter = pTest->input;	
<pre>11 pTest->output = pTest->local * pTest->parameter;</pre>	
12 }	
13	
<pre>1 pTest->parameter = pTest->input;</pre>	A
<pre>2 pTest->output = pTest->local * pTest->parameter;</pre>	
<u></u>	•
C function Definition	

In the upper part of FB-C editor is code generated based on your function. In the lower part, you can enter and change your code.

7.5.1 FB-C Features

There is a number of FB-C features in jPLCPro:

- Text is colored considering the syntax
- Undo last action
- Redo last action

7.5.2 Accessing Memory Zones

Accessing a memory zone is done by writing $\mathbb{Z}[I]$ where:

• z is the name of the memory zone

• I is the index in the memory zone

7.5.3 Accessing FB definition elements

When you define FB elements of FB-C element (inputs, outputs, parameters and locals), jPLCPro creates structure with FB-C element name, preceded with two underscores (__), and a function with FB-C element name, preceded with $f_{,}$ whose only parameter is a pointer to created structure. For each instance of FB-C element, one instance of the created structure is allocated in memory, and on each scan, function is called with pointer to the allocated structure.

You can access FB-C elements in C code through passed structure:

pModuleName->elementName

7.5.4 Accessing symbols, constants and variables

To access a symbol, constant or variable in code, simply write its name.

For example, setting the value of variable var1 of DWORD type to zero is done by writing:

var1 = 0

Note: Constants can NEVER be changed

7.5.5 Accessing complex data types

Accessing a value in a structure is done by writing structureName.elementName where:

- structureName is the variable name
- elementName is the value name

Accessing a member of array is done by writing arrayName[index] where:

- arrayName is the variable name
- index is the index in array

Accessing a member of array that is a member of structure is done by writing structureName. arrayName[index] where:

- structureName is the name of structure
- arrayName is the array field name
- index index the index in array

7.5.6 System Libraries

There are number of system libraries that can be used in FB-C code. All of them are standard libraries.

- stdint.h
- stdbool.h
- string.h
- math.h
- stdlib.h

Since users of FB-C code have to have at least basic knowledge of C Programming, these libraries are not discussed in this document.

7.6 Exporting FBs

To export Function Blocks and all other variables inside your project (User Symbols, User Constants, Variables or Data Types) go to File>Export Function Blocks... or press Ctrl+Shift+E. You will be presented with a dialog showing the selection of things to export. Once you select the content you want your library to contain simply press export and and select the location for the library file. After that the file will be created.

Export Library Function Blocks			
☑ JustStuff.jfb		⊡ FB_1.jfb	
✓ FB_2.jfb		FB_3.jfb	
User Symbols		🗹 hello21	
✓ hello_0		iv first	
		⊘ third	
✓ fourth		✓ fifth	
⊠ sixth		seventh	
ight ⊴		⊡ sevenu. ⊘ ninth	
∠ tenth		eleventh	
		_	
User Constants			
√ hello2		const_1	
🗹 const_2		🗹 const_3	
☑ const_4		☑ const_5	
☑ const_6		const_7	
Variables			
☑ var_1		☑ var_2	
☑ var_3		☑ var_4	
Data Types			
Structures	Arrays		Enums
✓ struct_1	⊘arr_1		enum_1
struct_2	☑ arr_2		⊡ enum_2
✓ struct_3	☑ arr_3		enum_3

8 BACnet Objects

By clicking on node BACnet Object, you open the table for setting the list of BACnet objects.



First you choose the type of object you want to define:

Analog values Analog inputs Binary values Binary inputs Multistate values

8 BACnet Objects

By clicking on node BACnet Object, you open the table for setting the list of BACnet objects.



First you choose the type of object you want to define:

- Analog values
- Analog inputs
- Binary values
- Binary inputs
- Multistate values

8 BACnet Objects

By clicking on node BACnet Object, you open the table for setting the list of BACnet objects.



First you choose the type of object you want to define:

Analog values Analog inputs Binary values Binary inputs Multistate values

log values In	nstance #	Object name	Address	Write function	Read function	Units	Comment
alog inputs		value_1	MWO	write_simple	read_simple	NO_UNITS	comment
ary values	10000	value_1	MW1	write_simple	read_simple	NO_UNITS	
ary inputs state values		value_3	MW2	write_simple	read_simple	NO_UNITS	

8 BACnet Objects

By clicking on node BACnet Object, you open the table for setting the list of BACnet objects.



First you choose the type of object you want to define:

Analog values Analog inputs Binary values Binary inputs Multistate values

9 Compiling

Compiling the program is done through the Compile Menu or a shortcut.

Compilation process is syntax analysis of all of FB (both diagrams and code) and translating the program into target code.

After successful compilation a data file named after the project is created. This file is used to write the program to the PLC. Also, all of the ".c", ".h" and ".o" files are created. These files can be explored through a standard text editor.

Successful compilation example:

8 BACnet Objects

By clicking on node BACnet Object, you open the table for setting the list of BACnet objects.



First you choose the type of object you want to define:

Analog values Analog inputs Binary values Binary inputs Multistate values



To open the build or distr (distribution) folder left-click on the folder path in Console View.

Source code produced in folder: <mark>C:\Users\ilke\Documents\JPLCProProjects\Test\build</mark> Code produced in folder: <mark>C:\Users\ilke\Documents\JPLCProProjects\Test\dist</mark> End compiling

8 BACnet Objects

By clicking on node BACnet Object, you open the table for setting the list of BACnet objects.



First you choose the type of object you want to define:

Analog values Analog inputs Binary values Binary inputs Multistate values

All of the information about the compilation process and files locations can be found in Console View.

If an error occurs during the compilation user is informed. More information about the error can be found in the Console View.

TExample - IPECPro Development version	the art to come the set of the		10-18 - 2 3-
Life Edit Project Looks Compile Online y			
的现在的 化化子 三部	• 🔯 🕸 🗰 🕲 💬		
14 Projects	di Editor		Properties 0
Electronia Bode Body (29 Bode) Bode Bode Bode Bode Bode Bode Bode Bode B	Borneyre (Book, Stock) (Code, Stock		
	C function Definition		
	C Conste (8 C 10 C 20	Ne Watch Window	* 12 B &
	L.		B Q
	<pre>Vectorstandingspect experiences and 0.PACK_TERECTON_Dest_attractive(tracket(t)):0 ALSDA_TERECT_RDD</pre>		ndue (Strine)

An error is displayed in Console View, painted red, in the following format:

8 BACnet Objects

By clicking on node BACnet Object, you open the table for setting the list of BACnet objects.



First you choose the type of object you want to define:

Analog values Analog inputs Binary values Binary inputs Multistate values

module:file_location:error_row:error_position:error_info

where:

- module is the module where error occurred
- file_location is the file where error occurred
- error_row is the row where error occurred
- error_position is the position of the error in a row
- error_info is the information generated by compiler about what caused the error

10.1 Downloading the program

To download the program to EPLC the device needs to be connected to jPLCPro. The program must compile without error or download will fail.

There are two ways to download the program, via the Write To PLC menu item or from the tool bar.

To compile and download a program left-click the Compile and Run icon in the toolbar. After compiling Write to PLC dialog will appear, download already in progress. Wait for download to complete and left-click the OK button.

10.1.1 Write To PLC dialog

Write to PLC dialog is accessed through the Write To PLC menu item in Online Menu

/rite To PLC Dialog	×
Project file : JPLCProProject.tar.gz PLC Mode : Running Run Idle Restart	Send
	OK

It features:

- Project file name
- Status bar
- Run button
- Idle button
- Send button
- Ok button
- Download status bar

Project File name is the name of the project file.

Status bar displays information about the current state of the program (running, idle or sending). It is green when the program is running, yellow when the program is stopped and blue while the program is downloaded to the PLC.

Run button is used to start the program on PLC.

Idle button is used to stop the program on PLC.

Send button is used to download the program to PLC.

OK button is used to close the Write To PLC dialog.

Download status bar displays information about the download of the program.

To be able to start or download the program the PLC status must be Idle.

10.1.2 Monitoring

Monitoring is a way to observe information about the FBs and various values of the PLC in realtime.

Monitoring needs for jPLCPro and EPLC to be connected and the program on EPLC to be running.

Monitoring is started through the Monitor On menu item in Online Menu, or via Monitor On icon from toolbar, or via shortcut CTRL+M

Monitoring is stopped through the Monitor Off menu item that replaces the Monitor On menu item in Online Menu during monitoring or via Monitor Off icon from toolbar or via shortcut CTRL+M.

When monitoring is started all of the FB tabs (except the MAIN_FB tab) are temporarily closed. This happens because monitoring is done on instances and editing is done on the bodies of FB.

During monitoring, FB instances can be entered with double-clicking the appropriate FB. When a FB instance is entered during monitoring a new tab is opened. The name of this tab is Parent/Name_Index where:

- Parent is the parent FB
- Name is the name of the FB instance
- Index is the index of the FB instance

FB-C cannot be entered as there is nothing to display in them.

10.1.3 FB Monitoring

FBs look the same as in edit mode with addition of the current values of inputs and outputs. Values of inputs and outputs are displayed below their names.

10.1.4 Input/Output Monitoring

Inputs/Outputs look the same as in edit mode, with addition of their current values. These values are displayed next to the name of input/output.



10.1.5 Forcing Values

Forcing (values) is a feature that can be used only during monitoring. It allows a user to change the current value of element (input or output) i.e. to force a value.

Forcing is done via Force... or Force on menu items in Online menu. Force on is used for forcing values of Bit type, and Force... is used for forcing all the others values.

To use Force on an input/output of type bit must be selected. Force on forces variable value

to 1. To force value of 0 to variable of bit type, select Force off. To turn forcing off, select Force off again.

Force... displays a dialog where user enters what and how should be forced in the format of N=V where:

- N is the name of variable, symbol or memory zone
- v is the value

To force left-click the OK button. To exit the dialog left-click the Cancel button.

Force		×
?	Force value: IW=1000	
	OK Cancel	

Force dialog can be invoked by pressing F11 shortcut.

10.1.6 Deforcing Values

De-forcing values is stopping the forcing of values. It can only be done on forced values.

To de-force any value left-click the "Deforce..." menu item in the "Online" menu.

A deforce dialog will appear. Its layout and button functions are the same as in the "Force..." dialog.

To deforce a value write its name.

Deforce dialog can be invoked by pressing F12 shortcut.

jPLCPro User Manual **11 FB Overview**

As mentioned before, there are a number of already defined FBs. Some of them have predetermined number of inputs, and some allow user to choose the number of inputs.

For those FBs with variable number of inputs a "Number of Inputs" dialog will appear

		×
(i)	Number of inputs	_
	OK Cancel	

Type in the desired number of inputs from the keyboard and left-click the OK button. Left-click the Cancel exit the dialog.

When number of inputs is selected, the element will have that number appended to its name.

11.1 Logical

Logical is a group of FBs that are operating with bits or bits in bytes or words etc.



11.1.1 AND

AND FB is a FB that performs the and operation on inputs.

	AND2	0
input1		out
input2		•
).

input2	input1	out
numeric	numeric	BIT
0	0	0
0	1	0
1	0	0
1	1	1

	AND3	1
input1		out
input2		
input3		

input3	input2	input1	out
numeric	numeric	numeric	BIT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	1	1

11.1.2 OR

OR FB is a FB that performs the or operation on inputs.

	OR2	2
input1		out
input2		

input2	input1	out
numeric	numeric	BIT
0	0	0
0	1	1
1	0	1
1	1	1

OR3	3
input1	out
input2	-
input3	
	j.

input3	input2	input1	out
numeric	numeric	numeric	BIT
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	1	1

11.1.3 XOR

XOR FB is a FB that performs the ${\tt xor}$ (exclusive or) operation.

	XOR	4
а		out
b		

a	b	out
BIT	BIT	BIT
0	0	0
0	1	1
1	0	1
1	1	0

jPLCPro User Manual **11.1.4 LESS**

LESS FB compares a and b. Output is 1 if a < b and is 0 if $a \ge b$.



a, b	out
numeric	BIT
a >= b	0
a < b	1

11.1.5 LEQ

LEQ FB compares a and b. Output is 1 if $a \le b$ and is 0 if a > b.



a, b	out
numeric	BIT
a > b	0
a <= b	1

11.1.6 GREATER

GREATER FB compares a and b. Output is 1 if a > b and is 0 if a <= b.



a, b	out
numeric	BIT
a <= b	0
a > b	1

11.1.7 GEQ

GEQ FB compares a and b. Output is 1 if $a \ge b$ and is 0 if a < b.



a, b	out
numeric	BIT
a < b	0
a >= b	1

11.1.8 EQUALS

EQUALS FB compares a and b. Output is 1 if a = b and is 0 if a != b.

EQUALS 9

11 FB Overv		ew
a, b	out	
numeric	BIT	
a != b	0	
a = b	1	

11.1.9 NOT_EQUALS

NOT_EQUALS FB compares a and b. Output is 1 if a != b and is 0 if a = b.



a, b	out
numeric	BIT
a = b	0
a != b	1

11.1.10 BITWISE_AND

BITWISE_AND performs bitwise and function on corresponding bits of a and b.



input1, input2	out
integral	integral
11000011,	01000001
01010101	0100001

11.1.11 BITWISE_OR

BITWISE_OR performs bitwise or function on corresponding bits of a and b.

BITWISE_OR2	12	2
input1	out	-
input2		•

input1, input2	out
integral	integral
11000011,	11010111
01010101	IIOIOIII

11.2 Arithmetic

Arithmetic group is a group of arithmetic FBs.

	Logical	•		
	Arthmetic	•		ADD
	Trigonometic	•		SUBTRACT
	Timers	•		MULTIPLE
	Edge detection	•		DIVIDE
	Bistabile	•		ABS
	Statistic	•		SQRT
	Selection	•		Nth_ROOT
	Custom	•		POW
	Input	I	1	log
	Output	0		log10
	Comment			SHIFT_LEFT
n.	Paste	Ctrl+V		SHIFT_RIGHT

11.2.1 ADD

ADD FB is a FB that performs addition.



input1	input2	sum
numeric	numeric	numeric
a	b	a + b

11.2.2 SUBTRACT

SUBTRACT FB is a FB that performs subtraction.



input1	input2	result
numeric	numeric	numeric
a	b	a - b

jPLCPro User Manual 11.2.3 MULTIPLY

MULTIPLY FB is a FB that performs multiplication.



input1	input2	result
numeric	numeric	numeric
a	b	a * b

11.2.4 DIVIDE

DIVIDE FB is a FB that performs division.

DIVIDE	3	3)
input1	result	
input2		•
		•

input1	input2	result	
numeric	numeric	numeric	
a	b	a / b	

11.2.5 ABS

ABS FB is a FB that finds absolute value of input.



input	out
numeric	numeric
x	x

11.2.6 SQRT

SQRT FB is a FB that finds square root of input.

	<u> </u>
SQRT	5
	out
	SQRT

input	out
numeric	FLOAT
x	\sqrt{x}

11.2.7 NTH_SQRT

NTH_SQRT FB is a FB that finds n^{th} root of input x.



		11 FB Overvi	ew
x	У	out	
numeric	numeric	FLOAT	
x	У	$\sqrt[y]{x}$	

11.2.8 POW

POW FB is a FB that finds the n^{th} power of input.



x	n	out
numeric	numeric	FLOAT
x	n	x ⁿ

11.2.9 LOG

LOG FB is a FB that finds natural logarithm of input value.



input	out
numeric	FLOAT
x	ln(x)

11.2.10 LOG10

LOG10 is a FB that finds logarithm with base 10 of input value.



input	out
numeric	FLOAT
x	log(x)

11.2.11 SHIFT_LEFT

SHIFT_LEFT is a FB that shifts bits of value to the left.

SHIFT_LEFT	10
value	out
count	

value	count	out
integral	integral	integral
10101010	3	00010101

11.2.12 SHIFT_RIGHT

SHIFT_RIGHT is a FB that shifts bits of value to the right

SHIFT_RIGHT 11 value out count

11 FB Overview			ew
value	count	out	
integral	integral	integral	
10101010	3	01010000	

11.3 Trigonometric

Trigonometric group is a group of FBs that perform trigonometric calculations.

	Logical	•]::	
	Arthmetic	•		
	Trigonometic	•		SIN
	Timers	•		COS
	Edge detection	•		TAN
	Bistabile	•		CTAN
	Statistic	•		ARCSIN
	Selection	•		ARCCOS
	Custom	•		ARCTG
	Input	I		ARCCTG
	Output	0		
	Comment			
ß	Paste	Ctrl+V		

11.3.1 SIN

SIN FB is a FB that finds sinus of a value.



input	out
numeric	FLOAT
x	sin(x)

11.3.2 COS

COS FB is a FB that finds cosines of a value.



input	out
numeric	FLOAT
x	cos(x)

11.3.3 TAN

TAN FB is a FB that finds tangents of a value.

	11 FB Overvi			
input	ut out			
numeric	FLOAT			
х	tan(x)			

11.3.4 CTAN

CTAN FB is a FB that finds cotangents of a value



input	out
numeric	FLOAT
x	ctg(x)

11.3.5 ARCSIN

ARCSIN FB is a FB that finds arcus sinus of a value



input	out
numeric	FLOAT
x	arcsin(x)

11.3.6 ARCCOS

ARCCOS FB is a FB that finds arcus cosines of a values



input	out
numeric	FLOAT
x	arccos(x)

11.3.7 ARCTG

ARCTG FB is a FB that finds arcus tangents of a value



input	out
numeric	FLOAT
x	arctan(x)

11.3.8 ARCCTG

ARCCTG FB is a FB that finds arcus cotangents of a values

	ARCCTG	7
input		out
		j

input	out				
numeric	FLOAT				
x	arcctan(x)				

11.4 Timers

Timers group is a group of FB that deal with time.

Logical		•							
Arthmetic		•							
Trigonometic		•	•	:	•	•	:	•	•
Timers		Þ			то	N			
Edge detection		•			то	F			
Bistabile		•			TP				
Statistic		•							
Selection		•							
Custom		•	•	:	•	•	:	:	•
Input	I								
Output	0								
Comment			•	:	•	•	•	•	•
Paste	Ctrl+V		•	•	•	•	•	•	•

11.4.1 TON

TON FB activates the output after it is triggered, and time period expires.



11.4.2 TOF

TOF FB activates the output upon triggering and deactivates the output after a period of time.



set	time	out
BIT	integral	BIT



11.4.3 TP

TP FB hold the output active for a period of time.



11.5 Edge Detection

Edge Detection group is a group of FBs that are triggered on signal level change.



11.5.1 EDRE

EDRE FB detects signal rising edge.



11.5.2 EDFE

EDFE FB detects signal falling edge.



11.6 Bistable

Logical		۲			•					
Arthmetic		۲								
Trigonometic		Þ								
Timers		Þ	•	:	÷	•	•	•	•	•
Edge detection		Þ								
Bistabile		۲			BE	SD				
Statistic		۲			BE	RD				
Selection				-		_	_			
SEIECUON		۲	•						-	
Custom		•	•	•	•	•	•	•	•	•
	I		-	•	•	•	•	•	•	•
Custom	I					•	•	- - -	• • •	•
Custom				· · ·	· · ·	-	- - -	- - - -	-	-

11.6.1 BESD

BESD FB (Bistable Edge Set Dominant) detects status of its inputs, and sets its output Page 60

jPLCPro User Manual according to table.



set	reset	prev. out	out
BIT	BIT	BIT	BIT
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

11.6.2 BERD

BESD FB (Bistable Edge Reset Dominant) detects status of its inputs, and sets its output according to table.

	BERD	1
set		out
reset		•
		j.

set	reset	prev. out	out
BIT	BIT	BIT	BIT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

11.7 Statistic

Static group is a FBs groups that find various statistical information.

	Logical	•										:	:	•
	Arthmetic	•	ľ	:	:	•	•	•	:	:	•	:	:	
	Trigonometic	•	ŀ											
	Timers	•	ŀ											
	Edge detection	•	· E	:		•	:	:	:	:		:	:	
	Bistabile	•												
	Statistic	•	·		4	١VE	RA	GE						
	Selection	•	•		Ν	١N	IMU	JM						
	Custom	•	·		Ν	1A)	ΩM	UM						
	Input	I			Ν	٩IN	IML	JM.	_AI	NAI	.00	G		
	Output	0			Ν	1A)	(IM	UM	_A	NA	LO	G		
	Comment		ŀ		•									
-	Paste	Ctrl+V												

11.7.1 AVERAGE

AVERAGE FB finds the average value of inputs.



inputs	out
numeric	numeric
<pre>input1, input2,</pre>	$\sum_{m=1-N}$ inputM
 inputN	N

11.7.2 **MINIMUM**

MINIMUM FB finds the minimum value of inputs.



input1, input2	out
numeric	numeric
a, b	min(a, b)

11.7.3 MAXIMUM

MAXIMUM FB finds the maximum value of inputs.

MAXIMU	м2 2	2
input1	result	
input2		•
		•

	11 FB Overvie	эw
input1, input2	out	
numeric	numeric	
a, b	max(a, b)	

11.7.4 MINIMUM_ANALOG

MINIMUM_ANALOG FB finds the minimum value of inputs.

MINIMUM_A	ANALOG2 3
input1	result
input2	

input1, input2	out
numeric	numeric
a, b	min(a, b)

11.7.5 MAXIMUM_ANALOG

MAXIMUM_ANALOG FB finds the maximum value of outputs.



input1, input2	out
numeric	numeric
a, b	max(a, b)

11.8 Selection

Selection group is a FBs group selects one of the inputs and passes it to output.



jPLCPro User Manual 11.8.1 SEL

SEL FB passes the first input if "g" is inactive or the second input if "g" is active.



g integral	result numeric
0	input0
1	input1

11.8.2 MUX

MUX FB passes the input indicated by the "k" input.



k	result
integral	numeric
0	input0
1	input1
2	input2

12.1 Binary Input to Binary Output

In FB programming, there is not much to do with strings there is no use of Hello World application (unless you have a device with Display). So our first example will be to just pass Binary Input (BI) to Binary Output (BO). This example works on all devices, which have at least one BI, and at least one BO. It does not have much sense if yours BO is relay output, but you may want to lit LED (which is also BO), when your BI is On.

- 1. First, we have to create new jPLCPro Project:
 - a. Click File->New->NewProject and in dialog that opens, type MyFirstProject, and then press OK button.
 - Select where you want your Project to be saved.
 - b. jPLCPro creates new Project, and displays empty Main FB in Editor View.
- 2. Next, we want to give meaningful names to our BI and BO:
 - a. In Project View, open MyFirstProject->Symbols->User Symbols
 - Empty User Symbols table is opened in Editor View.
 - b. Click on green + icon, and type MyBI, IXO, 0, My Binary Input in Symbol name, Address, Init Value and Comment fields respectively.
 - This will create User Symbol MyBI, which corresponds to first BI on your Device.
 - c. Click on green + icon, again, and type MyLED, QX0, 0, My LED in Symbol name, Address, Init Value and Comment fields respectively.
 - This will create User Symbol MyLED, which corresponds to first BO on your Device (we assume that it is LED).
 - d. Click on File->Save Project to make MyBI and MyLED visible to the rest of the Project.
- 3. Next, we want to connect MyBI with MyLED
 - a. Go to Editor View, Main FB
 - b. Right click on a grid of Main FB, and select Input (or you can just type I on a keyboard).

• New Input is created.

- c. Click on newly created Input (to make it selected), and in Properties View, from Drop Down Menu select MyBI.
- d. Right click on the grid of Main FB, and select Output (or you can just type O on a keyboard).

• New Output is created.

- e. Click on newly created Output (to make it selected), and in Properties View, from Drop Down Menu select MyLED.
- f. Hover with a mouse over output pin of MyBI, left click on a yellow square that appears, then hover over input pin of MyLED and left click on a yellow square that appears.
 - MyBI and MyLED are connected with a line.
- 4. You may want to add a comment, to explain your intent, to somebody else who is looking into your FB Diagram, or for yourself, if you open Diagram after number of years:
 - a. Right click on the grid of Main FB, and select Comment.
 - Green Comment is displayed in Diagram.

- b. Type text into Comment Text Box of Properties View:
 - Lit LED when BI is ON
- c. If you wish, you can change a color of Comment, by clicking on Color button, and selecting a color from a pallet.
- 5. Now, we are ready to compile our project.
 - a. Select Compile->Compile from Main Menu.
 - Project is saved and compiled. It is now ready to be downloaded to your Device.
- 6. If you have your Device ready and connected, you can download program to Device.
 - a. Select Online->Write to PLC (or press Ctrl+W)
 - Write to PLC dialog opens.
 - b. Stop progam on the Device by click on Idle button.
 - Program status indicates that Device is in Idle state.
 - c. Download program to Device with click on Send button.
 - d. After download has finished, start your program with click on Run button.
 - When you apply voltage to first BI on the Device, LED is lit.

Start	HyfirstProject - JPLCPro 5.0.1		
8	Ele Edit Broject Iools Compile Online	New Reb	
€ G	10 B 😕 16 A 🗸 🗳 😪 17 B	N 🔹 📾 🏨 🔳 🗢 🖸	
HET R	E Projects	rditor 80 2 Properties	-01
HEUT R	E HyfirstProject	💇 Persenters 🕱 🍓 MNDL/F6" 🚿 🦑 System Symbols 🕱 🗳 User Symbols 🕷	
HET R HET R MIT R M M M M M M M	Annuro Annuro		
	Hostiang Tree (1	I Desgn ■ Console ■ B M X	
E4		Wite implicit. function-declaration - 6-0-symbols.o. symbols.o. * C/Program Files (580)/EurotCC/IPECPro 5.0.1 - 20160613-1501/bm/cyqwin/opt/tookhaiss/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc" * -1°C/Program Files (580)/EurotCC/IPECPro 5.0.1 - 20160613-1501/bm/cyqwin/opt/tookhaiss/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc" * -0_X800T/C/Program Files (580)/EurotCC/IPECPro 5.0.1 - 20160613-1501/bm/cyqwin/opt/tookhaiss/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc" * 20160613-1501/bm/cyqwin/opt/tookhaiss/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc * C/Program Files (580)/EurotCC/IPECPro 5.0.1 - 20160613-1501/bm/cyqwin/opt/tookhaiss/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc" * -Wite-implicit. function-declaration - 6-0-variables.o variables.o * * C/Program Files (580)/EurotC/IPECProschains/arm-2012.09/b/m/cyqwin/opt/tookhains/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc" * 20160613-1501/bm/cyqwin/opt/tookhains/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc * * 20160613-1501/bm/cyqwin/opt/tookhains/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc * * 20160613-1501/bm/cyqwin/opt/tookhains/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc * * 20160613-1501/bm/cyqwin/opt/tookhains/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc * * 20160613-1501/bm/cyqwin/opt/tookhais/arm-2012.09/bin/arm-angstrom-linux-gnueabi-gcc	
* • 41 • • •		Despin Image: Console C/Program files (589)/EurolCC/JPCCPro 5.0.1 - 20150613-15301/bm/cypum/opt/Lookchains/arm-2012.09/bin/arm-angatrom-inux-gnuesebi-gcc" -1°C/Program files (589)/EurolCC/JPCCPro 5.0.1 - 20160613-15301/bm/cypum/opt/Lookchains/arm-2012.09/bin/arm-angatrom-inux-gnuesebi-gcc" -0_XX00C/Program files (589)/EurolCC/JPCCPro 5.0.1 - 20160613-15301/bm/cypum/opt/Lookchains/arm-2012.09/bin/arm-angatrom-inux-gnuesebi-gcc" -1°C/Program files (589)/EurolCC/JPCCPro 5.0.1 - 20160613-15301/bm/cypum/opt/Lookchains/arm-2012.09/bin/arm-angatrom-inux-gnuesebi-gcc" -0_XX00C/Program files (586)/EurolCC/JPCCPro 5.0.1 - 20160613-15301/bm/cypum/chuck/blc/burnolC/JPCCPros 20160613-15301/bm/cypum/opt/Lookchains/arm-2012.09/bm/arm-angatrom-inux-gnuesebi-gcc" main_files (586)/EurolCC/JPCCPros -20160513-15301/bm/cypum/opt/Lookchains/arm-2012.09/bm/arm-chuck/burnout/surf Loo - Sulf - Sundon/arm-2012.09/bin/arm-201	

12.1.1 Lessons Learned

In this example, we learned:

- How to create new Project
- How to give meaningful names to Device inputs and outputs
- How to connect FB inputs and outputs
- How to add Comments and change their color
- How to compile Project
- How to download program to Device and start it

jPLCPro User Manual 12.2 OnCounter FB

Sometimes, you may want to count number of cycles while your BI is On. In the next example we will create OnCounter FB, which does exactly that.

- 1. We can reuse Project from previous example, by saving it under new name:
 - a. Select File->Save Project As... from Main Menu (or press Ctrl-W), and type MySecondProject in opened dialog.
 - b. You may select the same location as for the first Project.
- 2. For this example, we will need three constants 0, 1 and 1000:
 - a. Select Constants->UserConstants from Project View.
 - b. Create ${\tt zero}\ constant\ of\ type\ {\tt DWORD}\ with\ value\ 0$
 - c. Create one constant of type DWORD with value 1
 - d. Create thousand constant of type DWORD with value 1000
- 3. Now, we can create new FB
 - a. In Project View right click on Functional Blocks, and select New Module...
 - b. In opened dialog type OnCounter, and leave FB as selection in Drop Down Box.
 - Click OK button, and new, empty, FB will be crated.
- 4. Next, we create input and output for our new FB:
 - a. Select Definition tab
 - b. In Inputs list, create one input (with name input) of type BIT.
 - c. In Outputs list, create one output (with name count) of type DWORD.
 - d. Save Project to share changes with the rest of the program.
- 5. Next, we create FB Diagram for our OnCounter FB:
 - a. Select Design tab of OnCounter
 - b. Create four inputs, and in Project View name them input, zero, count and one.
 - c. Create one output and in Project View name it count
 - d. Right click on the grid of OnCounter and add one Arithmetic->ADD block, with 2 inputs
 - e. Right click on the grid of OnCounter and add one Selection->SEL block
 - f. Connect inputs, outputs and blocks as in following picture:

Editor	
💑 MAIN_FB* 🛛 🛛 🥔 User Symbols 🔀 📥 OnCounter 🕱 🗎 Ժ User Constar	nts 🕱
	SEL 1
· · <u>· · · · · · · · · · · · · · · · · </u>	
input	g result count count
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
	- input 0
••••••••••••••••••••••••••••••••••••••	
ADD2 2	- input 1
count input1 sum	
	· · · · · · · · · · · · · · · · · · ·
input2	
• • • • • • • • • • • • • • • • • • •	
	<u>•</u>
Design Definition	

- 6. Now, we can use our newly created FB:
 - a. Go to Main FB in Editor View
 - b. Right click on Main FB grid, and select Custom->OnCounter (our local FBs are located there).
 - Our new FB is displayed, with one input and one output.
- 7. To make some use of our new FB, let's add one more LED output
 - a. Let's suppose that it is our second output QX2, and name it OnLED
- 8. Finish our Main FB, by adding one input with thousand value, and one Logical->Greater block.
 - a. Connect diagram as in following picture

Editor	8, 51 6 36
💑 MAIN_FB* 🚿 🛛 🖑 User Symbols 🚿 🛛 💑 OnCounter 🚿 🗌 🖑 User Constants 🚿 📄	
Lit LED when BI is ON	
MyLED MyBI	
Lit another LED if BI is On for more than 1000 cycles OnCounter	
input count a out OnLed	
· · · · · · · · · · · · · · · · · · ·	
thousand = 1000	•
Design	

- b. One input can be connected to multiple outputs, as MyBI is connected to MyLED and OnCounter->input.
- c. Any output, can have only one line in.
- 9. Select Compile->Compile & Run from Main Menu, download program to your Device and run it.

12.2.1 Lessons Learned

In this example we learned:

- How to save Project under new name
- How to create new FB
- How to define inputs and outputs of FB
- How to use output of FB as input to contained FB
- How to create Project constants
- How to use our new FB
- How to compile and download program to Device in one step

NOTE: You may have noticed that functionality that we implemented in this example could be achieved with standard Timers->TON FB.

jPLCPro User Manual 12.3 DelayedOut FB

If you have experience with C programming, you may notice that it would be very easy to combine OnCounter and GREATER FBs in one simple C function. Let's see how we can do that in jPLCPro.

- 1. Start with saving our previous example as MyThirdProject
- 2. Now, we can create new FB
 - a. In Project View right click on Functional Blocks, and select New Module ...
 - b. In opened dialog type DelayedOut, and change selection Drop Down Box to FB-C.
 - Click OK button, and new, FB will be crated, with empty structure ____DelayedOut and empty function f__DelayedOut.
- 3. Next, we create input and output for our new FB:
 - a. Select Definition tab
 - b. In Inputs list, create one input (with name input) of type BIT.
 - c. In Outputs list, create one output (with name out) of type DWORD.
- 4. Let us create one Parameter:
- 5. In Parameters tab, create parameter count of type DWORD, initial value 1000, max value 100000, min value 0 and increment of 1.
- 6. Save Project to publish new values.
- 7. Go to C function tab of DelayedOut FB.
 - a. You can notice that _____DelayedOut structure is filled with created variables.
 - b. Paste following code in lower part of DelaydOut editor:

```
if (DelayedOut->input) {
    DelayedOut->count++;
    if (DelayedOut->count > DelayedOut->time) {
        DelayedOut->out = 1;
    }
}
else {
    DelayedOut->count = 0;
    DelayedOut->out = 0;
}
```

- c. Save project
- 8. Replace OnCounter and GREATER FBs with newly created FB:
 - a. Drag a square around OnCounter and GREATER FBs, and thousand input, and press Del key on keyboard
 - Selected elements will be deleted
 - b. Place new Custom->DelayedOut FB instead of deleted elements
 - c. Connect MyBI with DelayedOut->input and DelayedOut->out with OnLed.
 - d. Select DelayedOut FB, and in Properties View change its priority to 1
 - \bullet Priority of <code>DelayedOut</code> will change to 1, and priority of <code>OnLed</code> will change to 2
- 9. Create another DelayedOut FB, change its priority to 2 and add Logical->XOR element, as infollowing picture.
 - a. Save Project

jPLCPro User Manual 12 FB Programming Examples 💑 MAIN_FB 🚿 🛛 🧈 User Symbols 🕺 🛛 📥 OnCounter 🚿 🗍 🖑 User Constants 🚿 🗍 🍓 DelayedOut 🔀 🛛 😻 Parameters* Lit LED when BI is ON Q MyBI MyLED Lit another LED if BI is On for more than 1000 cycles 1 3 DelayedOut XOR input out out 4 OnLed b 2 DelayedOut

input out

10. Change parameters for DelayedOut_2 FB

- a. In Project View double click Parameters
 - Parameters window will open
- b. Expand Default Group and MAIN_FB
- c. If DelayedOut_1 and DelayedOut_2 Parameter groups are not displayed, right click on Default Group and select Reset.
 - DelayedOut_1 and DelayedOut_2 parameter groups will appear.
- d. Select DelayedOut_2->time parameter's Init value and change it to 5000
- 11. Compile and download project to Device

12.3.1 Lessons Learned

In this example we learned:

- How to create new FB-C
- How to select group of FBs
- How to delete FBs
- How to crate FB parameters, and change it for different instances

▼ ►