

# ABB MEASUREMENT & ANALYTICS **IEC 61131 Developer's Guide** ABB Totalflow G5 devices

ң PCCU32 - [Entry]				i 🔤 X
Operate View Window Help				- 8
10 🛅 🔟 🖸 💽 🖬 🌆		🖗 🛴 🛄 🖅 🚿		
- TOTALFLOW - Communications - Totalflow/TCP	IsaGraf Resou	urce List Status		
- Totalflow/USB		Description	Value	-
Totalflow/COM0:	91.254.1	Isagraf Version	5.41.22	
I/O System				
Display		Active IEC Resource		_
EC Basic	91.255.254	Resource Number	1	
- System Variables	91.254.54	Resource Name		_
- Symbol Table	91.254.53	Last Message	NA	
IsaGraf				
		Resource State Management		
	91.255.215	Current State	UNKNOWN	
	91.255.50	IEC AutoStart	Off	=
	91.255.217	Start/Stop Resource	No	
	91.255.218	Clear Active Resource	No	
	91.255.64	Operation Mode	Production	
	91.255.251	Annunciate	1	
	91.255.56	Annunciator	Trace	
		Symbol File		
	91.254.0	Symbol File		
	91.255.253	Rescan Symbol File	No	
				-
	Re-read	Monitor	Print Screen Save Send Close Help X Help	۰

Supporting IEC 61131 applications for custom automation and control programming

Measurement made easy



# **Table of Contents**

Ac	ditio	nal inforr	nation6						
Су	vber s	ecurity							
Sa	fety		7						
	Safet	y symbol	conventions7						
	Potential safety hazards								
1	Desc	ription							
	1.1	IEC 6113	31 General specifications8						
	1.2	ISaGRAF	integrated development environment (IDE)8						
	1.3	Activatin	g ISaGRAF IDE9						
		1.3.1	Activate ISaGRAF Workbench9						
2	Deve	lop and i	run an IEC 61131 application11						
	2.1	Workflow	v overview						
	2.2	New Proj	ject from template: ISaGRAF_TPL12						
	2.3	Import T	OTALFLOW5_V2.tdb file13						
	2.4	Set Devi	ce Target = TOTALFLOW5_V214						
	2.5	Set Reso	purce Embed Symbol Table = True15						
	2.6	Create a	Program Organizational Unit (POU) in the project17						
		2.6.1	Add a POU17						
		2.6.2	Add POU variables17						
		2.6.3	Add POU ST code						
		2.6.4	Add global variables						
		2.6.5	Add proprietary ABB Totalflow variables19						
	2.7	Build and	d package the IEC application19						
		2.7.1	Build the project19						
		2.7.2	Create the package20						
		2.7.3	Instantiate the IEC interface from PCCU3221						
	2.8	Install th	e PKG and copy the INI file21						
		2.8.1	Copy the INI file to PCCU/INIFiles folder22						
		2.8.2	Install the IEC application in the target device22						
	2.9	Activate	and run the IEC application in the G5 device23						
		2.9.1	Verify that the IEC application is running26						
3	Inter	face wit	h the IEC application 27						
	3.1	Wire I/O	registers between the IEC application and the Totalflow G5 device27						
		3.1.1	Add an I/O device instance27						
		3.1.2	Wire an IEC variable to an I/O channel28						
		3.1.3	Assign a Totalflow register address to the I/O channel29						
		3.1.4	Verify the I/O wiring in PCCU						
	3.2	Assign a	specific Totalflow register address to IEC variables						
	3.3	Creating	the INI file for an IEC application32						
	3.4	Customizing the INI file							

		3.4.1	Example of INI file text	34
4	Direc	t downlo	ad the resource from the ISaGRAF IDE	35
5	Annu	inciators		39
	5.1	Annuncia	tion to view the state on RMC display	39
6	Debu	g mode.		41
7	Prog	ram exar	nple	42
	7.1	Main pro	gram	42
	7.2	EVENT_L	OG function (Code, Input, Value)	43
	7.3	Variables	3	43
8	Read	and writ	te efficiency	45
	8.1	Backgrou	ınd	45
	8.2	ISaGRAF	I/O Device	46
	8.3	Get*() a	nd Set*() Function Blocks	46
	8.4	Totalflow	apps read IsaVM registers	47
	8.5	Totalflow	apps write IsaVM registers	48
	8.6	Comparis	son	48
	8.7	Run-time	e diagnostics	49
Ту	pogra	phical co	onventions	50
Co	ntact	us		51

# List of figures

Figure 1: ISaGRAF screen displaying pre-existing project	9
Figure 2: Automation Collaborative Platform	. 10
Figure 3: License Info screen	.10
Figure 4: Workflow overview	.11
Figure 5: Main ISaGRAF IDE screen	.12
Figure 6: New project dialog box	.12
Figure 7: Solution Explorer window	
Figure 8: Import the ABB Totalflow device definition file	.14
Figure 9: Set the device properties	.14
Figure 10: View the resource properties	.15
Figure 11: Verify the resource properties	.15
Figure 12: Set Resource Compiler Options Check Array Index = True	.16
Figure 13: Set Resource Hardware target = TOTALFLOW5_V2	.16
Figure 14: Set Resource Settings Cycle Time = 1000	.17
Figure 15: Add Structured Text	.17
Figure 16: Add POU variables	. 18
Figure 17: Add structured text (ST) code	. 18
Figure 18: Add resource global variables	. 19
Figure 19: Build the IEC project	. 19
Figure 20: Packaging the IEC project	. 20
Figure 21: G5 IEC Packager options	.21
Figure 22: Connection Setup for the device loader	.22
Figure 23: G5 Device loader	
Figure 24: IEC application package selected in the G5 device loader	.23
Figure 25: ISaGRAF tab	.24
Figure 26: ISaGRAF Resource List tab	.24
Figure 27: Activate Resource drop-down menu	.25
Figure 28: Activated Resource	.25
Figure 29: Activated ISaGRAF tab	.26
Figure 30: Verify the IEC application is running (example)	.26
Figure 31: Device selector (available G5 I/O devices)	
Figure 32: I/O Device workspace	. 28
Figure 33: Variable Selector dialog box	. 28
Figure 34: I/O parameters for the I/O Device	. 29
Figure 35: Example of the NumRegisters variable	
Figure 36: Register address assigned to Battery Voltage	
Figure 37: Indirect addressing	.31
Figure 38: Indirect address value	.31
Figure 39: Assigning a register to an IEC application variable	. 32
Figure 40: Create an INI file	
Figure 41: G5 IEC Packager (INI generator)	. 33
Figure 42: INI file text example	. 34
Figure 43: PCCU view of the example INI file	. 34
Figure 44: Developer mode	
Figure 45: View the properties > resource number	
Figure 46: Enter IP address	
Figure 47: Download	
Figure 48: Download Resource alert	
Figure 49: Result of a successful resource download	
Figure 50: Annunciate feature	
Figure 51: Debug tab	
Figure 52: Application/License Management tab	

Figure 53: Communication between ISaGRAF and Totalflow apps	46
Figure 54: I/O Device	46
Figure 55: GetRealTest	47
Figure 56: Block selector	47
Figure 57: Reading ISaGRAF app from Totalflow operations app	48
Figure 58: Writing ISaGRAF from Totalflow operations app	48
Figure 59: Message timing	49

# List of tables

Table 1: Related Documentation	6
Table 2: IEC packager options	20
Table 3: ISaGRAF I/O codes	29
Table 4: IEC application register numbers	
Table 5: Pre-defined characters:	
Table 6: Defined variables	43
Table 7: Defined Words	
Table 8: CPU cost for each method	49

# Additional information

<b>Table 1: Related Documentatio</b>
--------------------------------------

Documents	Document number
IEC Installer Wrapper Guide for ISaGRAF <sup>®</sup> IDE	2105856
IEC INI programmer's guide	2105858

# **Cyber security**

The Digital Oilfield application integrates Totalflow products which are designed to be connected, and communicate information and data, via a network interface. All Totalflow products should be connected to a secure network. It is the customer's sole responsibility to provide, and continuously ensure, a secure connection between the product(s) and the customer network as well as a secured and controlled physical access to the hardware equipment, or any other network (as the case may be). The customer shall establish and maintain appropriate measures (such as, but not limited to, the installation of firewalls, the application of authentication measures, encryption of data, installation of antivirus programs, etc.) to protect the products, the network, its system and its interfaces against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Inc. and its affiliates are not liable for damages and/or theft of data or information.

Although ABB provides functionality testing on the products and updates it releases, the customer should institute its own testing program for any product updates or other major system updates (to include, but not limited to, code changes, configuration file changes, third party software updates or patches, hardware change-out, etc.) to ensure that the security measures the customer has implemented have not been compromised and that the system functions in the customer's environment as expected.

# Safety

Observe warning signs on packaging and on the devices, prior to software configuration.

# Safety symbol conventions

"NOTICE" hazards are associated with equipment or property damage, it must be understood that under certain operating conditions, operating damaged equipment can result in degraded system / process performance leading to serious or life-threatening injuries. Therefore, compliance with all "NOTICE" hazards is required at all times.



**NOTICE – Equipment damage or loss of data.** This symbol indicates a potential for equipment damage, loss of data or other unintended outcome. Failure to observe this information may result in damage to or destruction of the product and / or other system components.



**IMPORTANT NOTE:** This symbol indicates operator tips, particularly useful information, or important information about the product or its further uses.

# **Potential safety hazards**

IEC 61131 applications execute and interface with other applications in ABB Totalflow products. IEC application developers must be advanced users and be familiar with the operation of the product where they plan to install the applications.

Avoid disruption of commissioned devices. Carefully conduct and plan installation, activation and verification of the IEC application.

Review and follow all health and safety recommendations for the device described in the user manuals or startup guides when installing and running an IEC application during a device's field installation.

# 1 Description

This guide describes the development of IEC 6113 applications, installation, and activation in the ABB Totalflow G5 devices.

# 1.1 IEC 61131 General specifications

The IEC 61131 standard specification, developed by the International Electrotechnical Commission (IEC), provides a generic programming environment for the PLC industry. In conjunction with ISaGRAF<sup>®</sup>, ABB Totalflow has adapted their IEC compiler to work with the ABB Totalflow product line. This alliance has greatly enhanced the functionality of the products by providing the tools with which to build custom applications.

Different user applications may require different programming languages. One application may lend itself to a graphical language (Function Block Diagram) while another program may be best addressed using ladder logic (Ladder Diagram). The IEC 61131 ISaGRAF IDE supports the following programming techniques for G5 devices:

- ST: Structured Text
- FBD: Function Block Diagram
- LD: Ladder Diagram
- SFC: Sequential Function Chart

Using one of the above programming techniques, or using them in combination, should provide enough flexibility to address any special application problems that the field environment might present.

ABB Totalflow G5 devices, flash version 2.0.0 or greater, can run up to 10 IEC 61131 applications. Each IEC 61131 application requires an application credit.

— One IEC 61131 Interface Turned On = One ISaGRAF Resource

— One IEC 61131 application credit supports one instance of the IEC 61131 interface

These application credits can be applied to the flow computer at the factory or by purchasing a credit key (secure flash drive) with the appropriate credits installed.

# **1.2** ISaGRAF integrated development environment (IDE)



**IMPORTANT NOTE:** The information included in this section does not discuss the ISaGRAF environment beyond the interface to ABB Totalflow products. Consult the ISaGRAF documentation for general-purpose information regarding IDE and project organization. For specific information about the ISaGRAF interface with Totalflow, use the details in this guide.

Figure 1 shows the ISaGRAF screen after opening a pre-existing project. In this view, several windows are visible. In the Solution Explorer window, the project file and its contents are organized in a hierarchical tree. A window with variable definitions and information displays in the middle of the screen. The Navigation window provides access to additional utilities.

e Edit View Project Build Debug	a Tools Window Help			
the second s				- · · · · · · · · · · · · · · · · · · ·
]•⊞• <b>,,,,,,,,,,,,,,,,</b> ,,,,,,,,,,,,,,,,,,,				💀 - 📲 🔛 📕
B I	⊻ *   A   ≣ ≣ ≣   ⊟ •		<b>-</b>	
olution Explorer 🛛 👻 🕂 🤻 🥵	esource1-VAR ×		<ul> <li>Navigation Wind</li> </ul>	dow 🔻 🕂 🗙
	Name	Data Type Dimensi	on Alias Global	View View & Function Blocks evices
Solution 'ChemInj' (1 project)		· A* · A* ·	🖈 🔹 🖈 📕 Deployment	View
Solution Views	PumpType	BOOL -	Config1	▶ ☆
Deployment.isadpl	PlungrChemPumpOnCycle	INT -		
Config1	ManualStatePrev	BOOL -	Target I/O De	& Function Blocks
<ul> <li>Resource1</li> </ul>	TestStatePrev	BOOL -	Target Feature	
▲ I Programs	Spare38	BOOL +		103
<ul> <li>InjectCTL</li> <li>Local Variables</li> </ul>	Spare22	BOOL -		
Global Variables	TankAlm	INT -	Resource1	*
ISaVIEW1	TankLvlAlm	REAL *	I/O Device	
🔺 📴 Lib	TankLvlShutIn	REAL -	Global Variat	<u>oles</u>
▷ 锢 Functions 镭 Function Blocks	RemoteManualPumpOff	INT -		
	Cur_VCMode	INT -		
🖷 DataTypes	Cases 29	DE AI -		
itput				<b>→</b> ‡ ×
how output from: AnkhSVN - Subversion	- ↓ Ø			
				~
				*
Pending Changes 📃 Output				•
🕻 Cross References Browser  💏 Error List				

# Figure 1: ISaGRAF screen displaying pre-existing project

# 1.3 Activating ISaGRAF IDE

The ISaGRAF Workbench software (2104764) is required to build IEC applications for Totalflow controllers. This free software is available from order entry on a USB flash drive.

To order, see <u>Contact us</u> on the last page of this guide. Request 2104764 SOFTWARE, ISAGRAF WORKBENCH IEC DEVELOPMENT.

The USB flash drive includes:

- ISaGRAF Workbench 6.6
- Totalflow Packager for building IEC packages that can be loaded on RMC
- RMC and XRC target files
- Documentation for building IEC applications on RMC and XRC

For best results, verify the latest OS and flash software is loaded on the RMC or XRC.

#### 1.3.1 Activate ISaGRAF Workbench

Registration Keys are required to activate the ISaGRAF work bench. Order Entry at ABB Totalflow will provide the registration keys. Use the following steps to find the information needed.

- 1. Open the License Info window: **ISaGRAF Automation Collaborative Platform.**
- 2. Click Help.
- 3. Click Licensing CAM 5. The Workbench registration window opens.
- 4. Call ABB Totalflow order entry (see <u>Contact us</u>).
- 5. Read the user codes from the Workbench registration window.

# Figure 2: Automation Collaborative Platform

Automation Collaborative Platform			
FILE EDIT VIEW TOOLS WINDOW	HELP	2	
G-0 📅 - 🗀 - 🔛 💾 🗶 🗗	0	View Help	Ctrl+F1
Q Q Q Q	o Bo	Add and Remove Help Content	Ctrl+Alt+F1
Solution Explorer	6	Licensing CAM 5	
		Order Help on DVD	
00 H   -	3	About	

# Figure 3: License Info screen

License Info	
Register to ABB Totalflow License: Engineering License Expiration: This license does not expire.	Registration         User Code 1:       321701723         User Code 2:       87828450         User Code 3:       ABC51779380903         Registration Key 1:
www.isagraf.com	Validate Cancel

# 2 Develop and run an IEC 61131 application

The procedures in this section describe how to create, install, and run an IEC 61131 application in an ABB Totalflow device. The workflow overview provides the end-to-end high-level view of the major tasks required to complete development and run the application.

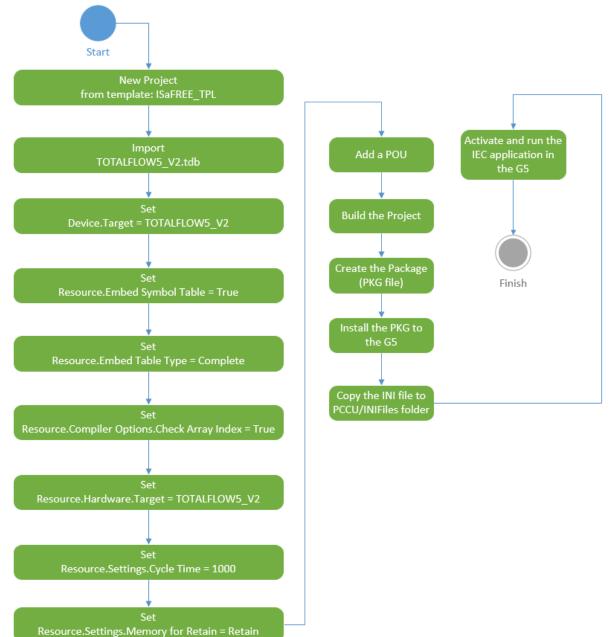


**IMPORTANT NOTE:** This manual refers to the embedded Totalflow application, which supports IEC applications, as simply the "IEC interface". IEC 61131 application is shortened to the "IEC application".

# 2.1 Workflow overview

<u>Figure 4</u> provides the workflow required to create, install, activate, and run an IEC application in an ABB Totalflow device. Review each procedure and follow the detailed instructions in each corresponding section in the order presented.

# Figure 4: Workflow overview



# 2.2 New Project from template: ISaGRAF\_TPL

To create a new project using ISaGRAF:

- 1. Open the ISaGRAF IDE.
- 2. Click File> New> Project (Figure 5).

# Figure 5: Main ISaGRAF IDE screen

	Automation Collaborative Platfor EDIT VIEW TOOLS WII	rm (Administrator) NDOW HELP			
FILE	EDIT VIEW TOOLS WI	NDOW HELP			
	New	•	រ៉ៃ🖥 Project	Ctrl+Shift+N	🔽 🔽 💭 🗮 🖋 📾 🔀 🚽 🏙 🆄 🧌 🐜 🦓 😸 🕼 🍪 🆓 🥵 🖓 🤣 🖉 🌮 Password 🛛 Clear Password 🖉 Cycle Timing (ms):
	Open	•	4 19 ° >		📕 - Connection Timeout (ms): 4000 🔹 🖌 🖕 😩 📲 🚽
	Close				
	Close Solution				
Ш	Save Selected Items	Ctrl+S			
н <sup>ра</sup>	Save All	Ctrl+Shift+S			
	Generate Documentation	Ctrl+P			
	Recent Projects and Solutions	•			
X	Exit	Alt+F4			

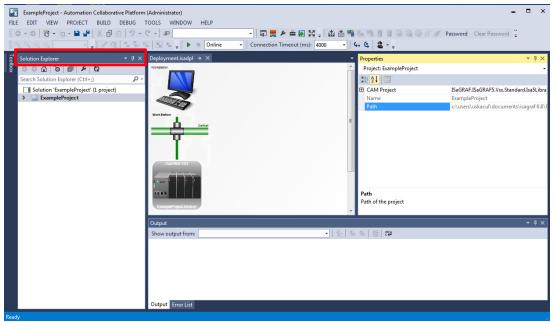
- 3. At the New Project dialog box (Figure 6), select the ISaFREE\_TPL template from the list.
- 4. Type the name of the project in the Name field.
- 5. Click **OK** to use the default location folder to save the project file, or click **Browse** for another location folder.

## Figure 6: New project dialog box

New Project			And and the second second	? ×
▶ Recent	S	ort by: Default	- # E	Search Installed Templates (Ctrl+E)
<ul> <li>Installed</li> <li>CAM Projects</li> </ul>		ISaFREE_TPL	CAM Projects	Type: CAM Projects Template for ISaGRAF Free Windows
<ul> <li>ISaGRAF 3</li> <li>ISaGRAF 5</li> </ul>				target v5.23 or later
Import Simulator Windows				
ISaGRAF installed te	emplates			
Name:	ExampleProject			
Location:	c:\users\tf dev\doc	uments\isagraf 6.6\Project	5 🔫	Browse
Solution name:	ExampleProject			Create directory for solution
				OK Cancel

Verify that the project just created displays in the Solution Explorer window (Figure 7).

# Figure 7: Solution Explorer window



# 2.3 Import TOTALFLOW5\_V2.tdb file

ABB Totalflow provides a definition file (TOTALFLOW5\_V2.tdb) that enables the ISaGRAF environment to specifically target the ABB Totalflow G5 device.



**IMPORTANT NOTE:** The TOTALFLOW5\_V2.tdb file provides functions to read and write to the data types available in G5 devices. The definition file also defines the embedded OS, the processor type, memory mapping and the overall target device environment. The TOTALFLOW5\_V2.tdb file definition is required for compatibility with Totalflow version 2.0.0 and on G5 devices. TOTALFLOW5\_V1.tdb will no longer be supported.

To import the device definition:

- 1. From the ISaGRAF Solution Explorer window, right-click the project that was just created.
- 2. From the pop-up menu, select **Import**> **Import Target Definitions** (Figure 8).
- 3. At the Open dialog box, select **TOTALFLOW5\_V2.tdb**.
- 4. Click **Open**.

			Automation Collaborative Platf					- 🗆 ×
FILE	EDIT VIE		PROJECT BUILD DEBUG		NDOW HELP	_		
0	- 0   13	• [	」 - 🗎 📲 👗 🗗 約 🤊					🖁 🐜 🐴 🗒 🕼 🌚 🖉 🖉 Password 🏺
	२ व व	Q	9		🗧 🕨 🔳 Online 🔹 Conr	nection Tir	meout (ms): 4000 -	6. G 🗟 - 🚽
Toolbox	olution Explo	rer	<b>→</b> ‡		t.isadpl → ×	-	Properties	- ↓ ×
box	00	ō	a > 0	Workstation		*	Project: ExampleProject	7
Se	arch Solutio	on Exp	plorer (Ctrl+;)	D -			8∎ 2↓ 🖾	
			mpleProject' (1 project)				CAM Project	ISaGRAF.ISaGRAF5.Vsx.Standard.Isa5Libra
	📄 Exar		Binding	100			Name	ExampleProject
		*	Build		3		Path	c:\users\uskacul\documents\isagraf 6.6\l
		lice of	Clean					
		-	Download		DerNet	E		
		103	Online Change					
			Scope to This					
			New Solution Explorer View		STGT			
			Add	•				
		Г	Import	,	Import Exchange File			
		Т	Export	•	Import Target Definitions	r.		
		4	Set as StartUp Project		ct.Device1	_	Name	
			Debug	•			Name of the project	
		ж	Cut	Ctrl+X		٣		
		a	Paste	Ctrl+V				<b>-</b> म ×
		×	Delete	Del	it from:		- 9	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
		X	Rename					
			Unload Project					
			Password					
			Target Password					
		s	Properties	Alt+Enter				
	-	_						
				Output Er	ror List			
Ready								

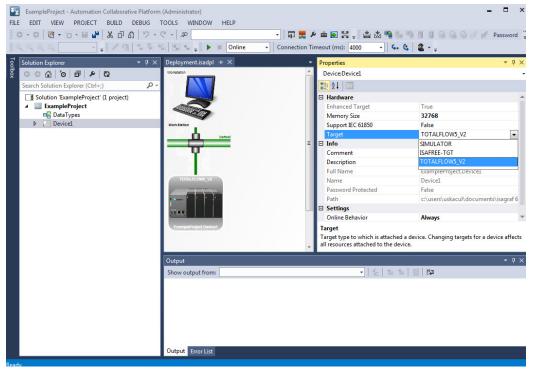
## Figure 8: Import the ABB Totalflow device definition file

# 2.4 Set Device Target = TOTALFLOW5\_V2

To set target priorities:

- 1. From the ISaGRAF IDE main top menu bar, select **View > Properties Window**.
- 2. After the TOTALFLOW5\_V2.tdb is installed, expand the project in the Solution Explorer window (i.e., **ExampleProject**), and select **Device1** to view properties (Figure 9).

#### Figure 9: Set the device properties



- 3. In the Target property field, change the value to **TOTALFLOW5\_V2**.
- 4. Click Yes.

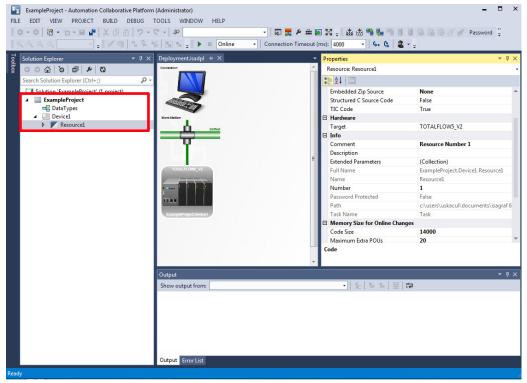
# 2.5 Set Resource Embed Symbol Table = True

The resource properties are set up at the Properties windows. Expand the different sections to locate specific properties as instructed in this procedure. Use the scroll bar if needed to navigate through the window to locate properties.

To set the Resource properties:

- 1. In the Solution Explorer window, expand **Device1**.
- 2. Click **Resource1** to view in the Properties Window (Figure 10).

#### Figure 10: View the resource properties



- 3. In the Properties window, select **Code** > **Compiler Options** (Figure 11):
  - Ensure that Embed Symbol Table property is set to **True**.
  - Ensure that Embedded Table Type property is set to **Complete**.

#### Figure 11: Verify the resource properties

Pr	operties	
F	Resource: Resource1	
•	2↓ 🖻	
Ξ	Code	
	Code For Simulation	True
Ξ	Compiler Options	
	Check Array Index	True
	Dump Configuration Files	False
	Dump Network	False
	Dump POU Files	False
	Enable Code Optimization	False
	Function Internal State Enable	True
	Generate Map File	False
	Indirect Bit Access Validation	True
	Reduce Boolean Expression Evalu	False
	Target Supports Optimized TIC C	False
	Embed Symbol Table	True
	Embedded Table Type	Complete
	Embedded Zip Source	None
	Structured C Source Code	False
	TIC Code	True

4. In the Properties window, expand **Compiler Options**, and set the Check Array Index property to **True**. See the figure below.

### Figure 12: Set Resource Compiler Options Check Array Index = True

Properties	
Resource: Resource1	
Code For Simulation	True
Compiler Options	
Check Array Index	True
Dump Configuration Files	False
Dump Network	False
Dump POU Files	False
Enable Code Optimization	False
Function Internal State Enable	True
Generate Map File	False
Indirect Bit Access Validation	True
Reduce Boolean Expression Evaluation	False
Target Supports Optimized TIC Code	False
Embed Symbol Table	True
Embedded Table Type	Complete
Embedded Zip Source	None
Structured C Source Code	False
TIC Code	True
Hardware	
🗆 Info	
<b>Check Array Index</b> Indication of whether to verify the validity	of array indexes

5. In the Properties window, expand **Hardware** and ensure the Target property is set to TOTALFLOW5\_V2. See figure below.

# Figure 13: Set Resource Hardware target = TOTALFLOW5\_V2

Properties	Ψ.
Resource: Resource1	
₽∎ <b>2</b> ↓ □	
⊞ Code	
🗆 Hardware	
Target	TOTALFLOW5_V2
🗆 Info	
Comment	Resource Number 1
Description	
Extended Parameters	(Collection)
Full Name	ExampleProject.Device1.Resource1
Name	Resource1
Number	1
Password Protected	False
Path	c:\users\uskacul\documents\isagra
Task Name	Task
Memory Size for Online Changes	
Code Size	14000
Maximum Extra POUs	20
SFC States Mem size	4096

- In the Properties window, expand Settings. See Figure 14.
   Change the Cycle Time property value to 1000.
   Ensure the Memory for Retain property is set to RETAIN.

## Figure 14: Set Resource Settings Cycle Time = 1000

Properties	
Resource: Resource1	
⊡ Code	
⊞ Hardware	
⊞ Info	
Memory Size for Online Changes	
Memory Usage Info	
□ Settings	
Cycle Time	1000
Cycle Time Units	ms
Detect Errors	True
Execution Mode	Real Time
Memory For Retain	RETAIN
Nb Stored Errors	16
Online Behavior	Always
Trigger Cycles	True

# 2.6 Create a Program Organizational Unit (POU) in the project

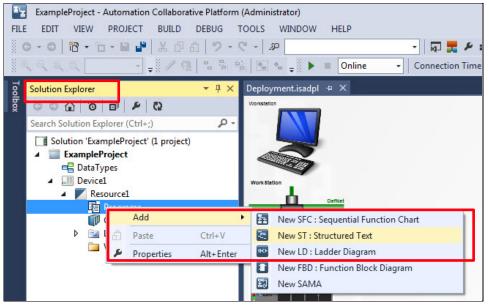
This section includes procedures to create a simple program (POU). The logic for an IEC application is contained in a POU. A POU belongs to a Resource and may use the various languages available within ISaGRAF. The example used in the following steps uses Structured Text (ST) language.

# 2.6.1 Add a POU

To add a POU to the Resource:

 In the Solution Explorer window, Right-click Resource1 and select Add > New ST: Structured Text. See the figure below.

# Figure 15: Add Structured Text



# 2.6.2 Add POU variables

To add local variables to the POU:

- 1. Expand the program or POU previously added. In the Solution Explorer window, expand Prog1.
- Under Prog1, Double-click Local Variables. A table to define variables displays. See Figure 16.
   Add a Local variable. For example: Name = cycleCounter, Data Type = UINT, Initial Value = 0, Direction = Var, Attribute = Read/Write, and Comment = Increments each time Prog1 runs.

# Figure 16: Add POU variables

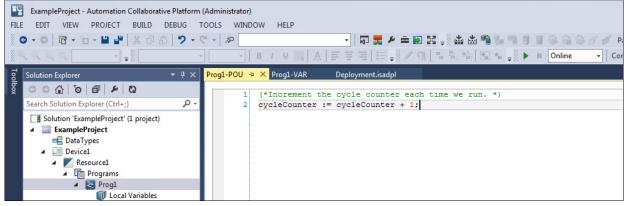
ExampleProject - Automation Collaborative	Platform (Adm	ninistrator)								
FILE EDIT VIEW PROJECT BUILD DE	EBUG TOOL	S WINDOW HELP								
0-0 🔞 - 🖬 - 💾 🖉 🕺 🗗 🙃	12.6.	.\$P		- 🗖 🗮 4	🖉 🛋 🛍 🖌	- 🚢 🛎 🖻	6 9 U U I	899900	Password	Clear Password 🚆
A Q Q Q / A		🖫 🖕 🕨 🔳 On	line +	Connection T	imeout (ms):	4000 -	4. G 🚨 - 🚽			
Solution Explorer	▼ Ŧ × Pro	g1-VAR 🗢 × Deploy	ment.isadpl							
0 4 16 10 10 0		Name	Data Type	Dimension	String Size	Initial Value	Direction	Attribute	Retained	
Search Solution Explorer (Ctrl+;)	ρ-	- A*	- A*	~ dt*	* d*	- at	- A*	- A*	- A*	
Solution 'ExampleProject' (1 project)		cycleCounter	UINT -				Var 👻	Read/Write 🔹		Increments each time Prog1 runs
🖷 DataTypes	*		STRING TIME	*			÷	*		
Device1			UDINT							
<ul> <li>Resource1</li> <li>End Programs</li> </ul>			UINT							
A 🔄 Prog1			USINT	τ.						
Local Variables										

# 2.6.3 Add POU ST code

To add code to the POU:

- 1. Under Programs, double-click **Prog1**. See Figure 17.
- 2. Add the ST code. In this example, this code increments the cycleCounter variable.

# Figure 17: Add structured text (ST) code



## 2.6.4 Add global variables

To add global variables to the Resource:

- 1. In the Solution Explorer window, expand **Device1** and **Resource1**.
- 2. Under Resource1, double-click **Global Variables**. See Figure 18.
- 3. Add the Global variable as shown. For example:
  - Name = PARTNUMBER
  - Data Type = STRING
  - String Size = 64
  - Initial Value = 2509999-001
  - Direction = Var
  - Attribute = Read/Write
  - Comment = the INI file name to use with this IEC application

## Figure 18: Add resource global variables

E EDIT VIEW PROJECT BUILD D	DEBUG T	OOLS WINDOW HEL	Р							
6 G X 🗳 🖬 - 🗗 - 🗃 🖉 X G A	19.0	- <i>s</i> p		- 🗔 🗮	۵ 🖬 🖿 🕅	- 🚢 🛎 🐴 🖏		4400	Password (	Ilear Password Cycle Timing (ms):
( a, a, a,		- B / U					🖕 🕨 🗏 Onl	line 👻 🕹 Co	onnection Tir	neout (ms): 4000 - 🛛 🕹 🦕
Solution Explorer	• å ×	Resource1-VAR 🗢 🗙 P	rog1-POU	Prog1-VAR	Deploy	ment.isadpl				
0 0 10 0 0		Name	Data Type	Dimension	String Size	Initial Value	Direction	Attribute	Retained	Comment
Search Solution Explorer (Ctrl+;)	ρ-	* d	· • •	· A	· A	· 01	· A	* of	· d	
Solution 'ExampleProject' (1 project)		SYSVA_CYCLEC	n dint 👻				VarGlobal 🔹 👻	Read 🔹		Cycle counter
DataTypes		SYSVA_CYCLED	A TIME 🐳				VarGlobal 🔹	Read 🔹		Timestamp of the beginning of the cycle in milliseconds (m
Device1		SYSVA_KVBPER	F BOOL 🔍 👻				VarGlobal 🔹 👻	Read 🔹		Kernel variable binding producing error (production error)
<ul> <li>Resource1</li> <li>Frograms</li> </ul>		SYSVA_KVBCER	F BOOL 🔷 👻				VarGlobal 🔹 👻	Read/Write 🔹		Kernel variable binding consuming error (consumption error
🔺 💽 Prog1		SYSVA_RESNAM	/ STRING 🚽				VarGlobal 🔹 👻	Read 🔹		Resource name (max length=255)
Local Variables Global Variables	_	SYSVA_SCANCI	V DINT 🗸				VarGlobal 🔹	Read 🔹		Input scan counter
🕨 🕍 Lib		SYSVA_TCYCYC	1 TIME 🔷 👻				VarGlobal 🔹	Read/Write 🔹		Programmed cycle time
🔚 Variable Groups		SYSVA_TCYCUF	F TIME 🚽				VarGlobal 🔹	Read 👻		Current cycle time
		SYSVA_TCYMA	X TIME 🔶				VarGlobal 🔹	Read 🔹		Maximum cycle time since last start
		SYSVA_TCYOVE	F DINT 🔹				VarGlobal 🔹	Read 🔹		Number of cycle overflows
		SYSVA_RESMO	e sint 🔹 👻				VarGlobal 🔹	Read 🔹		Resource execution mode
		SYSVA_CCEXEC	BOOL -				VarGlobal 🚽	Read/Write 👻		Execute one cycle when application is in cycle to cycle mod
		J PARTNUMBER	STRING 🔍		64	'2509999-001'	Var 👻	Read/Write 👻		The INI file name to use with this IEC application

4. In Solution Explorer, right-click **Resource1**. Select **Rename**, enter the name of the project, and Press **Enter**.

# 2.6.5 Add proprietary ABB Totalflow variables

There are two special variables that ABB Totalflow uses to help the developer:

- PARTNUMBER: Used by PCCU32 to determine the INI file name for the selected IEC program.
- NumRegisters: Set the size of the indirect address array. For details, see section <u>3</u>: <u>Interface with the IEC application</u>.



**IMPORTANT NOTE:** ABB Totalflow variables are case sensitive.

# 2.7 Build and package the IEC application

After the application has been completed, use the following procedures to run the application on the Totalflow G5 device.

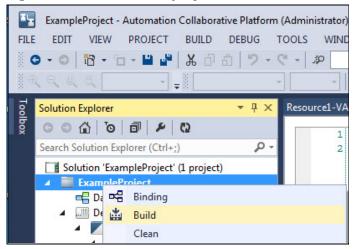
- Build the project
- Package the IEC application

# 2.7.1 Build the project

To build the project code:

- 1. Right-click on the project.
- 2. Click **Build** (Figure 17). The IEC code is created.

#### Figure 19: Build the IEC project



# 2.7.2 Create the package

The packager can create a package file or PKG (a file with extension .pkg) for installation in a Totalflow G5 device.

#### Table 2: IEC packager options

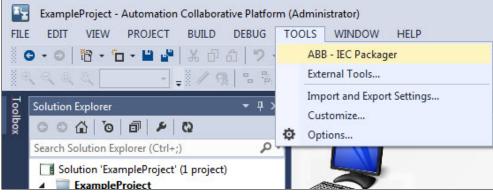
IEC Resource	Select the Resource for packaging from the IEC Resource dropdown menu if the IEC project contains multiple Resources.						
IEC Package Name	The name of the package file to create. This is the name that displays in PCCU under Available IEC Resources after installation of the IEC application. The default is the name of the resource within the ISaGRAF IDE.						
Description Tag	The description displays in the Totalflow device loader when installing the IEC application (PKG file). The default is the comment property of the resource within the ISaGRAF IDE.						
INI Name	The name of the INI file to generate. The default is the value of the PARTNUMBER string variable within the ISaGRAF IDE.           Important note: The PARTNUMBER is the name of the INI that PCCU will load when this IEC application is running in the G5 devices. The PARTNUMBER is automatically retrieved from the Global Variables.						
Create PKG File	Option to create a package file for installing to the G5 device						

Create INI File	Option to create an INI file for interfacing to the IEC application within PCCU						
Show PKG/INI file in file explorer	When checked, file explorer will open to show the new PKG and INI file.						

To package the IEC application:

- 1. Select the project in Solution Explorer.
- 2. Select **Tools** on the top main menu (Figure 20).
- 3. Select **ABB IEC Packager**.

# Figure 20: Packaging the IEC project



4. In the G5 IEC Packager window, select **Options**. See the figure below.

#### Figure 21: G5 IEC Packager options

IEC Project		IEC INI F	ìle
IEC Project Name	ExampleProject	INI Name	2509999-001
IEC Device	Device1		Expand Registers 10
IEC Resource	Resource1 🔹		Expand Registers 10     Add System Variables
IEC Recource POU's	PROG1		Add Auto Var Tabs
Pkg File			
	EB-1		Create PKG File
IEC Package Name	ExampPrj		Create INI File
Description Tag	G5 ExampleProject Resource Number 1		Show PKG/INI file in file expl

5. Click **OK** to create the package and INI file.

File explorer opens to show the location of the new files. Take note of the location of the files. This is required to complete the installation.

# 2.7.3 Instantiate the IEC interface from PCCU32

The IEC Interface is the ABB Totalflow application required to support and run IEC applications. The IEC Interface is the link between the ISaGRAF application and the embedded Totalflow software in the G5 device.

To instantiate the IEC interface:

- 1. Connect to the target device using PCCU.
- 2. In Entry mode, locate and select the meter ID at the top of the PCCU32 tree view.
- 3. Click the Application/License Management tab.

Under **Device Credits**, verify that there is at least one unused basic IEC credit available per instance. Any available credits display under the Surplus/Deficit column (highlighted in green).

4. If there are no basic IEC credits available, insert a credit key and add the required credits. Click **Help** for more information.



**IMPORTANT NOTE:** The IEC Interface application requires a basic IEC credit. To add a basic IEC credit, refer to the Application and License Management help topics in PCCU.

- 5. After the credit is available, click **Add**.
- 6. Click the **Application to add** drop-down list.
- 7. Select the **IEC Interface**. A default application number is automatically configured.
- 8. Click **OK**. In the application table, verify that the IEC Interface application is displayed.



**IMPORTANT NOTE:** If there are no credits available, the Basic IEC application may still show in the list, but it will not be possible to enable the application.

9. Click Send.

Verify that the IEC Interface application displays in the table and in the PCCU tree view.

# 2.8 Install the PKG and copy the INI file

After building the code and packaging the IEC application files, install the application in the target device and copy the INI file to the PCCU directory. The procedures in this section describe how to install the IEC application package and its INI file. The IEC application must be installed in the device to be activated, and the INI file must be in the PCCU directory to be able to display the custom screens designed for the application.



**IMPORTANT NOTE:** For more details, see the help topics for IEC 61131 in PCCU.

# 2.8.1 Copy the INI file to PCCU/INIFiles folder



**IMPORTANT NOTE:** During development, it may be advantageous to use the built-in automatic INI until most variables are defined.

To copy the INI file:

- 1. Locate the INI file created. The default location is in the ISaGRAF installation directory shown by the G5 IEC Packager, as described in section <u>2.7.2. Create the package</u>
- 2. Copy the INI file to the IniFiles folder in the PCCU installation directory.

# 2.8.2 Install the IEC application in the target device



**IMPORTANT NOTE:** Use the following procedure for G5 devices and the RMC. For more information about the device loader, see the G5 Device Loader help files.

- 1. Start PCCU.
- 2. On the top tool bar, select the **32-bit Loader** icon.
- 3. If you have backed up the device data and configuration, click **OK**.
- 4. At the Connection Setup window, click **Connect**. See the figure below.

#### Figure 22: Connection Setup for the device loader

Connection Setup	X	
		٦
Connection Type	Serial/USB •	
Port	COM8:Totalflow 32 Bit X-Series	
Security Code	0000	ł
(Required if security	v switch is on.)	
	Connect Exit Help	
	- 0	

5. On the device loader screen (Figure 23), under Package Location, click **Browse** to locate the IEC application package file. It is shown by the G5 IEC Packager (see section 2.7.2 Create the package).

# Figure 23: G5 Device loader

Eu PCCU32	
E Services Help	
Package Location	Connection
	Serial/USB COM10:To
Package	Device
	Device Information - Device Type: RMC - Board Part Number: 2105023-001 - Serial Number: RE2002541-00000059 - OS: Production OS
	Version: 1.1.0-9

- 6. From the browser window, select the file, and click **Open**. The example below uses ExamplePrj. Pkg.
- 7. As shown in <u>Figure 24</u>, verify that the file is listed in Package View and is classified as an App package. Select the **App checkbox.**

Figure 24: IEC application package selected in the G5 device loader

PCCU32				
Services Help				_ 8 :
Package Location		Connection		·
C:\PCCU7\PackageDir\ExamplPrj.pkg	▼ Browse	Serial/USB	COM10:Totalflow	32 Bit X-Seri
Paskage	Device			
Paokage Information └∭App: G5 ExampleProject Resource 1 └Package Size: 10.47 (KB)	-Serial Num -OS: Produc -Version: -Part Num	e: RMC Number: 2105023 ber: RE2002541-( tion OS 1.1.0-9 ber: 2105411-019 Size: 11.03 (MB)	9	
Loader Status Logs				
		Send	d 🔉 🛛 Abort 🛛 🕻 Cla	Help
dy		#Polls: 0 #Errors:	0 Connected to COM10:To	otalflov Login: user

#### 8. Click Send.

Monitor the Log view in the loader screen to verify that the file is sent successfully to the target device. After the package file transfer is complete, the Loader Status Logs shows the message "Device info updated".

9. Click **Close** to exit the device loader. Activate the application as described in the following section.

# 2.9 Activate and run the IEC application in the G5 device

After the application is installed in the device, the application must be activated to run or execute. The procedure included in this section describes how to activate and run the IEC application. For the example application used:

Change the ISaGRAF Active Resource to **Activate & Run** for the ExamplePrj resource.

View the cycleCounter variable in PCCU to verify that the ExamplePrj IEC application is running.

To activate and run the IEC application:

- 1. Connect to the device using PCCU and select **Entry** mode.
- 2. On the tree view, expand **IEC Basic**.
- 3. Under IEC Basic, select ISaGRAF.

#### Figure 25: ISaGRAF tab

PCCU32 - [Entry]				)
Operate View Window	Help			- 6
10 🛅 🛅 💽 💽	ŭa 🥵 🖻	j 📧 🗇 🎁 🖃 🛄 🖉	S. 🔗	
□ TOTALFLOW				_
Communications	IsaGraf Reso	urce List Status		
- Totalflow/TCP		Description	Value	
Totalflow/USB	91.254.1	Isagraf Version	5.41.22	Ê.
Totalflow/COM0: ⊡-I/O System				1
⊞ Display		Active IEC Resource		
E IEC Basic	91.255.254	Resource Number	1	1
System Variables	91.254.54	Resource Name		
- Symbol Table	91.254.53	Last Message	N/A	
IsaGraf				]
		Resource State Management		
	91.255.215		UNKNOWN	=
	91.255.50	IEC AutoStart	Off	
	91.255.217	Start/Stop Resource	No	_
	91.255.218	Clear Active Resource	No	4
	91.255.64	Operation Mode	Production	4
				4
	91.255.251 91.255.56	Annunciate Annunciator	I Trace	-
	91.255.56	Annunciator		4
		Symbol File		
	91,254,0	Symbol File		4
	91,255,253	Rescan Symbol File	No	_
				1 ÷
	Re-read	Monitor	Print Screen Save Send Close Help X.Help	а.

4. Select the **Resource List** tab to display the installed application.

## Figure 26: ISaGRAF Resource List tab

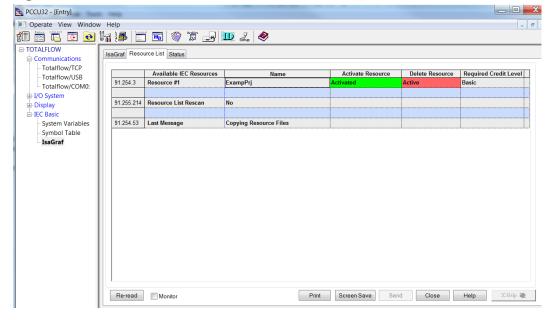
PCCU32 - [Entry]					_ <b>O</b> X
Operate View Window Help					- 8 ×
10 😇 😇 💽 🔯 🞼	😇 🖼 🗇 🎁 🖃	🛄 🛃 🧇			
TOTALFLOW Communications Totalflow/TCP	esource List Status				
- Totalflow/USB	Available IEC Resources	Name	Activate Resource	Delete Resource	Required Credit Level
Totalflow/COM0: 91.254.	3 Resource #1	ExampPrj	Not Active	No	Basic
I/O System					
	214 Resource List Rescan	No			
🖨 IEC Basic					
System Variables 91.254.	53 Last Message	N/A			
Symbol Table <b>IsaGraf</b>					
IsaGraf					
					Þ
Re-rea	d Monitor	Print	Screen Save Send	Close	Help X Help 🗮
Ready		#Polls:	29 #Errors: 0 Co	onnected to 169.254.0.	.11 Login: user

- 5. On the Resource List tab, select the **Activate Resource** drop-down menu (see <u>Figure 27</u> and <u>Figure 28</u>).
- 6. Click Select & Run.
- 7. Click Send.

the PCCU32 - [Entry]	_		-			
	v Help	Second Second	Sec. Sec.			
			m 3 🔌			
Operate View Window     Operate View View View     Operate View View View View     Operate View View View View View View     Operate View View View View View View View Vie	in 18   E	Available IEC Resources Resource #1 Resource List Rescan Last Message	No NA	Activate Resource Not Active Not Active Select & Run	No	Required Credit Level Basic
	Re-read	Monitor		Print Screen Save S	end Close	Help XHelp 🀲

#### Figure 27: Activate Resource drop-down menu





After activation, return to the **ISaGRAF** tab. Under the Description Column, Current State (app 255.215), verify that the resource has started. To automatically start the IEC resource when the TOTALFLOW controller boots, set the IEC Auto Start = **ON** (see figure below). For more information, click the context-sensitive **Help**.

#### Figure 29: Activated ISaGRAF tab

PCCU32 - [Entry]									
Operate View Window	v Help		_ <i>6</i> ×						
10 🖻 🖻 💽 💽 🙀 18   🗂 🖼   🎯 🛱 🖃 🛄									
-TOTALFLOWCommunications	IsaGraf Resource List Status								
- Totalflow/TCP		Description							
- Totalflow/USB	91,254,1	Isagraf Version	Value 5.41.22						
Totalflow/COM0:	51.254.1								
I/O System		Active IEC Resource							
	91.255.254	Resource Number	1						
System Variables	91,254,54	Resource Name	ExampPrj						
Symbol Table	91.254.53	Last Message	Copying Resource Files						
IsaGraf									
		Resource State Management							
	91.255.215	Current State	TARTED						
	91.255.50	IEC AutoStart	Off						
	91.255.217	Start/Stop Resource	No						
	91.255.218	Clear Active Resource	No						
	91.255.64	Operation Mode	Production						
	91.255.251	Annunciate	R						
	91.255.56	Annunciator	Trace						
		Symbol File							
	91.254.0	Symbol File	/tfData/IEC-91/ExampPrj/IDS00101						
	91.255.253	Rescan Symbol File	No						
	Re-read	Monitor	Print Screen Save Send Close Help XHelp						
Ready			#Polls: 56 #Errors: 0 Connected to 169.254.0.11 Login: user						



**IMPORTANT NOTE:** For additional instructions to manage the IEC resource, click **Help** from the **ISaGRAF tab** screen. Or search for the **IEC 61131 Applications** topic within PCCU help.

# 2.9.1 Verify that the IEC application is running

To verify that the IEC is running:

1. Select **IEC Basic** in the tree view. The defined tabs reflecting the IEC application custom INI should display (<u>Figure 30</u>).



**IMPORTANT NOTE:** The INI file is dependent on the specific definitions of the IEC application. Verifying that an IEC is running may require viewing results, providing input, etc.

2. For the example used, verify that the cycleCounter variable is incrementing as expected. **Figure 30: Verify the IEC application is running (example)** 

Devic	ce1 Prog	1-VAR ⊀	X	ExampProj-V	'AR D	eployment.isad	pl						
	Name	Data T	/pe	Dimension	String Size	Initial Value	Direction	Attribute	Retained	Comment	Alias	Wiring	Address
	· 01		at -	· Æ	· A*	· A	· #	· A	· de	~ <i>d</i> t	· 01*	· 01*	· A
	cycleCounter	UINT	•				Var -	Read/Write -		Increments each time Prog1 runs			
•	Battery	REAL	*				VarInput 🗸 🗸	Read 🔷 👻		RMC Battery level		%IR0.0	9001
*			*				+	-					

# 3 Interface with the IEC application

The procedures included in this section describe how to configure the IEC application to interface with other Totalflow applications.

# **3.1** Wire I/O registers between the IEC application and the Totalflow G5 device

This procedure describes how to wire an IEC variable to receive its value from a Totalflow register that is external to the IEC Interface application. For example, an IEC variable can reflect the value from the battery voltage at register address 7.3.99. For a complete explanation of I/O wiring, see the ISaGRAF help files.

The following are simplified examples of how to:

- 1. Add an I/O device instance.
- 2. Wire an IEC variable to an I/O channel.
- 3. Assign the I/O channel to a Totalflow register address (App.Array.Index).
- 4. View the results in PCCU.

# 3.1.1 Add an I/O device instance

An I/O device is a group of I/O channels that are specific to a hardware platform (the G5 device in this case). The I/O channels have a data type and a direction (input to IEC, or output from IEC). The I/O device definitions are sourced from the imported TDB (TOTALFLOW5\_V2.tdb). As such, the G5 TDB provides I/O devices for each supported data type (Figure 31).

# Figure 31: Device selector (available G5 I/O devices)

Package	Driver	Device	Data Type	Direction	Comment
Totalflow	Register	DINT_Outputs	DINT	Output	Write 32 bit Signe
Totalflow	Register	DINT_Inputs	DINT	Input	Read 32 bit Signe
Totalflow	Register	LINT_Inputs	LINT	Input	Read 64 bit Signe
Totalflow	Register	LINT_Outputs	LINT	Output	Write 64 bit Signe
Totalflow	Register	LREAL_Outputs	LREAL	Output	Write 64 bit Floati
Totalflow	Register	LREAL_Inputs	LREAL	Input	Read 64 bit Floati
Totalflow	Register	REAL_Outputs	REAL	Output	Write 32 bit Floati
Totalflow	Register	REAL_Inputs	REAL	Input	Read 32 bit Floati
Totalflow	Register	STRING_Outputs	STRING	Output	Write 64 Byte Stri
Totalflow	Register	STRING_Inputs	STRING	Input	Read 64 Byte Stri
Totalflow	Register	UDINT_Outputs	UDINT	Output	Write 32 bit Unsig
Totalflow	Register	UDINT_Inputs	UDINT	Input	Read 32 bit Unsig
Totalflow	Register	ULINT_Inputs	ULINT	Input	Read 64 bit Unsig
Totalflow	Register	ULINT_Outputs	ULINT	Output	Write 64 bit Unsig
Index: 0	Number (	of 5			

To add an I/O device:

1. From Solution Explorer, right-click on the resource, then click **I/O Device** to view the I/O Device workspace (Figure 32).

## Figure 32: I/O Device workspace

Reso	urce1 I/O Device * 🌵 🗙 Deployment.isadpl		Prog1-POU	Properties
	O:: Totalflow: Register: REAL_Inputs			
×		In	Name	
		0	%IR0.0	
<b>*</b>		1	%IR0.1	
*		2	%IR0.2	
<b>a</b>		3	%IR0.3	
		4	%IR0.4	
B				
3				
1				
<u> </u>				
			Gain = 1.0/1.0 Offset = 0.0 Conversion=None	

- 2. In the upper left of the I/O Device workspace, on the I/O Wiring toolbar, click the **Add Device** icon.
- 3. In the Device Selector dialog (Figure 32), select **REAL\_Inputs** device.
- 4. Enter **5** in the Number of field (this is the number of channels to expose for wiring).
- 5. Click **OK**.
- 6. The I/O Device workspace shows the I/O Device instance and the 5 channels in the wiring grid.

# **3.1.2** Wire an IEC variable to an I/O channel

An I/O channel wires an IEC variable to an I/O device (specific data type, direction, register address). To wire an IEC variable to the first I/O channel:

1. Double-click **First Channel** in the Wiring Grid: %IR0.0. The Variable Selector dialog displays (<u>Figure 33</u>).

#### Figure 33: Variable Selector dialog box

💼 Va	riable Selector											X
Nan Bat	ne ttery		Type RE/			•		obal Scope xampProj		T	Local Scope Prog1	-
Glob	oal Variables - Exan	npProj Local	Variables - Pro	og1								
	Name	Data Type	Dimension	String Size	Initial Value	Direc	tion	Attribu	te	Retained		C_
	· 04*	· 0#*	· 0#*	* 0 <b>*</b> *	* 0**		* <b>A</b> *		de*	* 0 <sup>#*</sup>		=
	cycleCounter	UINT				Var	-	Read/Write	-		Increments each time Prog1 runs	
	Battery	REAL				VarInput	+	Read	+		RMC Battery level	
*							<b>.</b>		-			
											OK Cano	el

2. Enter a new variable to wire. Click **OK**. The wired variable displays as %IR0.0=Battery@Prog1 in the wiring grid.



**IMPORTANT NOTE:** The wired variable name is coded to show %<direction><data type><I/O device index>.<channel index>. See the Table below.

# Table 3: ISaGRAF I/O codes

Direction	Code	Туре	Code
INPUT	I	DINT / UDINT	D
Ουτρυτ	Q	REAL	R
		String	S
		LREAL	L
		LINT / ULINT	L

# 3.1.3 Assign a Totalflow register address to the I/O channel

An I/O channel wires an IEC variable to a Totalflow register address (App.Array.Index). The Totalflow register address is exposed in PCCU, allowing the user to enter an App.Array.Index. The live value at that App.Array.Index register address is then available in the wired IEC variable, thus interfacing and connecting external G5 device application variables to IEC application variables.

To assign a Totalflow register address to the I/O Channel:

- 1. Expand the I/O Device Instance (0:Totalflow:Register:REAL\_Inputs).
- 2. Expand Parameters and double-click the **Ch1=0.0.0** item. The I/O Parameters REAL\_Inputs dialog box displays. See the figure below.

#### Figure 34: I/O parameters for the I/O Device

Prog	l-VAF	R Exam	pProj I/O Device * 🕒 🗙	Properties	
		Commen			In Name 0 %IR0.0=Battery@Prog1
*		I/O Parame	eters REAL_Inputs		
		Name	PhysicalValue	Comment	Format 🔺
		Ch1	255.253.1	App.Array.Index	STRING =
		Ch2	0.0.0	App.Array.Index	STRING
3		Ch3	0.0.0	App.Array.Index	STRING
	- 1	Ch4	0.0.0	App.Array.Index	STRING
		Ch5	0.0.0	App.Array.Index	STRING
	- 1	Ch6	0.0.0	App.Array.Index	STRING
	- 1	Ch7	0.0.0	App.Array.Index	STRING
	- 1	Ch8	0.0.0	App.Array.Index	STRING Dne
	- 1	Ch9	0.0.0	App.Array.Index	STRING
		Ch10	0.0.0	App.Array.Index	STRING
		Default			OK Cancel

3. In the Ch1 Physical Value column, type **255.253.1**.



**IMPORTANT NOTE:** An App value of 255 resolves to the IEC Interface app slot (91 by default). An Array value of 253 indicates the data type is Register, see <u>Table 4: IEC</u> <u>application register numbers</u>. An index value of 1 indicates to use the first register entry in the array. Therefore, with the IEC Interface Application at slot 91, the address resolves to 91.253.1 at runtime by the IEC interface app.

4. Click **OK** and note that the I/O Device 0:Totalflow:Register: REAL\_Inputs, I/O Channel 1 under Parameters indicates the newly created register address.



**IMPORTANT NOTE:** The IEC developer must use the special IEC variable named NumRegisters to expose assigned register addresses in PCCU. The Initial Value of the NumRegisters variable (see <u>Figure 35</u>) will determine the range of addresses to create (255.253.1 to 255.253.<NumRegisters>).

The value given to NumRegisters must also be used in the G5 IEC Packager when generating an INI. The value should be entered into the Expand Registers field.

#### Figure 35: Example of the NumRegisters variable

Devi	ice1 Prog1-VA	AR <mark>Ex</mark>	ampProj-VAR	🕈 🗙 Deplo	yment.isadpl				
	Name	Data Type	Dimension	String Size Initial Value		Direction	Attribute	Retained	Comment
	~ <i>0</i> **	~ <i>A</i> *	· #*	· A	~ <i>d</i> **	· A*	~ dt*	· A*	
	SYSVA_RESMOD	SINT -				VarGlobal 👻	Read 👻		Resource execution mode
	SYSVA_CCEXEC	BOOL -				VarGlobal 🕞	Read/Write 🕞		Execute one cycle when application is in cycle to cycle mode
	_IO_IR0_2	REAL -				VarDirectly 🔹	Read 🔹		
	_IO_IR0_3	REAL -				VarDirectly 👻	Read 🔹		
	_IO_IR0_4	REAL -				VarDirectly 🗸	Read 🔹		
	_IO_IR0_1	REAL -				VarDirectly 🔹	Read 🔹		
Þ	NUMRegisters	UDINT 🔍 🗸			5	Var 🗸	Read 🗸 🗸		Number of indirect addresses
*		÷				÷	÷		

# 3.1.4 Verify the I/O wiring in PCCU

Follow this verification procedure once the IEC application has been built, packaged, and installed in the device as described in section <u>2</u>, <u>Develop and run an IEC 61131 application</u>.

To verify that the newly created I/O wiring is functioning as expected:

- 1. Activate and run the IEC application as described in section <u>2.9</u>, <u>Activate and run the IEC</u> <u>application in the G5 device</u>.
- 2. On the tree view, expand the **I/O System** application.
- 3. Select Auxiliary I/O.
- 4. Take note of the Totalflow register address to wire to the IEC application. For the example here, the register for the value of the Battery Voltage is 7.3.99 (See the figure below).

#### Figure 36: Register address assigned to Battery Voltage

晧, PCCU32 - [Entry]							_ 🗆 🗙
Operate View Window Help							_ & ×
10 🛅 🔟 💽 💽 🦙 17 18 🖉 18	i 🚽	🛄 🥜 🧶					
□→G5 IEC	nalog [	igital					
- Totalflow/TCP							
Totalflow/USB	7.2.00	Description	13.509		Value		
Totalflow/COM0:	7.3.99	Battery Voltage	13.509				
I/O System	7.3.5	System Voltage Charger/External Voltage	0.201				
	1.3.0	Charger/External voltage	0.201				
└─ TFIO-A Modules ⊕ Display							
H-G5 IEC Inteface							
	Re-rea	ad Monitor	Print	Screen Save	Send	lose Help	X Help 🗮

- 5. On the tree view, select **IEC Interface**.
- 6. Select the **Registers** tab.

# Figure 37: Indirect addressing

强 PCCU32 - [Entry]			<u>ا</u> ز ا	X
Operate View Window Help				- 8
10 🔤 🗔 💽 💽 🗽 🎼 🚞 🚳	18 🚽 🛄	🗴 🛃 🧇		
⊟ G5 IEC	Llint16 Llint32	Float String Registers		
		riour ounig		
- Totalflow/TCP		Name	Value	
- Totalflow/USB - Totalflow/COM0:	91.253.1	Register 1	7.3.99	
	91.253.2	Register 2	0.0.0	
Auxiliary I/O	91.253.3	Register 3	0.0.0	
TFIO-A Modules	91.253.4	Register 4	0.0.0	
	91.253.5	Register 5	0.0.0	
G5 IEC Inteface	91.253.6	Register 6	0.0.0	
• Operations	91.253.7	Register 7	0.0.0	
	91.253.8	Register 8	0.0.0	
	91.253.9	Register 9	0.0.0 91.253.8	
	91.253.10	Register 10	0.0.0	
	91.253.11	Register 11	0.0.0	
	91.253.12	Register 12	0.0.0	
	91.253.13	Register 13	0.0.0	
	91.253.14	Register 14	0.0.0	
	91.253.15	Register 15	0.0.0	
	91.253.16	Register 16	0.0.0	
	91.253.17	Register 17	0.0.0	
	91.253.18	Register 18	0.0.0	
	91 253 19	Ponietor 19	0 0 0	*
	Re-read	Monitor	Print         Screen Save         Send         Close         Help         XHelp         XHelp	۰.

7. Type the register address **7.3.99** into the register 91.253.1, Register[1].



**IMPORTANT NOTE:** The register 91.253.1 was the register given when assigning a Totalflow register address to the wired I/O channel in section <u>3.1.3</u>, <u>Assign a Totalflow register address to the I/O channel</u>.

- 8. Click Send.
- 9. Verify that the Battery Voltage value shown in the IEC interface (Figure 38) matches the value in register 7.3.99 (Figure 36).

# Figure 38: Indirect address value

탄국 PCCU32 - [Entry]						_ 🗆 🗙
Operate View Window Help						- 8 ×
10 🛅 🗔 💽 💽 🐂 🌆 🛅 🖼 🚳 1		🛄 🤪 🧇				
G5 IEC	Uint16 Ui	nt32 Float String Registers				
Totalflow/TCP Totalflow/USB		IEC Variable	Value		Description	
Totalflow/COM0:		PROG1 Local Variables			•	
I/O System	91.9.1	Battery	13.50879	RMC Battery level		
Auxiliary I/O						
TFIO-A Modules  Display						
⊕-G5 IEC Inteface						
Operations						
	1					_
	Re-rea	ad Monitor	Print	creen Save Send	Close Help	X Help 💐
Ready		#Po	lls: 32 #E	rrors: 0 Connected to	169.254.0.11 Logi	n: user

# 3.2 Assign a specific Totalflow register address to IEC variables

The IEC variables are exposed to the user through the INI file in PCCU. Each IEC variable is automatically assigned a Totalflow register address. For the example used, the IEC variable named Battery appeared at address 91.9.1 under the Float tab in PCCU. See the figure above.



**IMPORTANT NOTE:** Use fixed address, instead of automatic address.

Fixed address assigns specific addresses to the IEC variables and will remain constant. It is important when external systems are programmed to access the IEC variables.

Instead of using a 255.253.x indirect address, an absolute external address could be used. For example, enter 7.3.99 in the Ch1 Physical Value to source the value directly. Use this method when it is not necessary to enter an App.Array.Index. The address property of an IEC variable can be used to control the Totalflow register address that is assigned. The format for the address is:

— The first digit of the array is the first part of the address.

— The 3 digit hex value of the index (or register number) is the second part of the address. For example, to set the address to 91.9.2, enter 9002. The slot number is omitted, the hex value of array 9 is entered as A, and the hex value of register number 2 is entered as 002.

The table below shows the array number to use depending on the data type of the IEC variable to be addressed.

	•	-9			
Туре	Array	Туре	Array	Туре	Array
Bool	0	Int32	5	Float	9
Int8	1	Uint32	6	Double	10
Uint8	2	Int64	7	String	11
Int16	3	Uint64	8	Time	12
Uint16	4			Date	13
				Register	253

#### Table 4: IEC application register numbers

To set the assigned register address for the Battery IEC variable:

1. In the Prog1 Local variables window (Figure 39), type the address into the Address column of the Battery variable.

## Figure 39: Assigning a register to an IEC application variable

Device	1 Prog	1-VAR +⊨ 🔅	< ExampProj-	VAR E	eployment.isad	ipi					<u> </u>	ĺ
	Name	Data Type	Dimension	String Size	Initial Value	Direction	Attribute	Retained	Comment	Alias	Wiring	Address
	· A	- 0	· · ·	· · ·	· A	· A	~ of*	· A	· d	· d'	· A	~ ø
	cycleCounter	UINT				Var 🐳	Read/Write 🐳		Increments each time Prog1 runs			
•	Battery	REAL				VarInput 🔹	Read 🔹		RMC Battery level		%IR0.0	9001
*						*	*					

# 3.3 Creating the INI file for an IEC application

ABB Totalflow's PCCU32 uses INI files to define user screens within PCCU32. INI files are text files. PCCU32 parses the INI files to create the various screens within PCCU32. If a custom INI file is not created for the IEC application, a default INI file is provided.

To create a custom INI file:

1. Create the variable, PARTNUMBER, within the ISaGRAF development tool (Figure 40).

#### Figure 40: Create an INI file

ISaGRAF - [SineWaves E Eile Edit Debug Tools Opt	tions <u>W</u> indow <u>H</u> elp				) - Dictionar	y - Variał	oles]								-   =   > -   =   >
▲ ◆ ● ■ ■ × ● * ■ ● ■ ● × +		79 E													
	All variables														
🖃 🔤 Variables 🚺	Name	Alias	Type	0	Initial value	Dimensi	Group	Attribute	Scope	Direction	Retain	Wiring	Address	Comment	
G SineWaves (XFC)	One		REAL		1.0		None	Free	Global	Internal	No				
	Run		BOOL		1		None	Free	Global	Internal	No				1
😑 🛄 Any Group	WaveCount1		DINT		100		None	Free	Global	Internal	No				1
All variab	WaveCount2		DINT		200		None	Free	Global	Internal	No				]
	WaveCount3		DINT		300		None	Free	Global	Internal	No				1
Global va	WaveOffset1		REAL		50.0		None	Free	Global	Internal	No				1
. SineWaw	WaveOffset2		REAL		100.0		None	Free	Global	Internal	No				1
- Silenan	WaveOffset3		REAL		75.0		None	Free	Global	Internal	No				1
	WaveOut1		REAL				None	Write	Global	Output	No	%QR0.0			
	WaveOut2		REAL				None	Write	Global	Output	No	%QR0.1			
	WaveOut3		REAL				None	Write	Global	Output	No	%QR0.2			
	WavePeriod1		TIME		T#1s		None	Free	Global	Internal	No				
	WavePeriod2		TIME		T#1s		None	Free	Global	Internal	No				1
	WavePeriod3		TIME		T#1s		None	Free	Global	Internal	No				1
	WaveRange1		REAL		25.0		None	Free	Global	Internal	No				1
	WaveRange2		REAL		50.0		None	Free	Global	Internal	No				1
	WaveRange3		REAL		10.0		None	Free	Global	Internal	No				
	PartNumber		STRING	65	"SineWaves"		None	Free	Global	Internal	No				
							1				100				1
VO-W-0063392.USJIWEL														NUM	

2. In the IEC Interface application in PCCU32, point to the specific, custom INI file to be used. For the purpose of this manual, the custom INI file is named Sine-001.ini (or Sine-001fm.ini Expert view) and should reside in the \PCCU\IniFiles folder.



**IMPORTANT NOTE:** The file prefix cannot be longer than ten (10) characters. PCCU32 can automatically decrement the revision level (-001) to a lower number of the file if the current version does not exist on the current PCCU32 directory.



**IMPORTANT NOTE:** ABB Totalflow supplies a program, G5 IEC Packager, which will read an ISaGRAF project and generate an INI file as a starting point for a developer to customize. (see Figure below). Use Add Auto Var Tabs to make an Expert view file. Make all of the customizations and duplicate the files as an Advanced view INI file.

Figure 41: G5 IEC Packager (INI generator)

IEC Project			IEC INI File	e
IEC Project Name IEC Device	ExampleProject Device1		INI Name	2509999-001
IEC Resource	ExampProj	-		Expand Registers 10     Add System Variables
IEC Recource POU's	PROG1	*		Add Auto Var Tabs
Pkg File				Create PKG File
IEC Package Name	ExampProj			Create INI File
Description Tag	Resource Number 1			Show PKG/INI file in file explorer.

# 3.4 Customizing the INI file

INI files are text files that are read by PCCU32 or WinCCU32 to determine how to display data within an application. A simple example is presented in this section to illustrate the screens that are rendered based on the definitions in the INI file. Refer to the INI Programmer's Manual for details about creating the INI file and for reference information about INI commands.

# 3.4.1 Example of INI file text

The following example shows text used in an INI file. The description for each line is provided below. The line numbers are for reference only and are not used in an actual INI file.

# Figure 42: INI file text example

```
1 [Status]
```

2 dsc:Enable Controller;cmd:13.0.1;typ:e;lst:ok=0,ALARM=1;

- 3 dsc:Alarm;cmd:13.6.1;typ:J;lst:ok=0,ALARM=1;
- 4 dsc:Stick Available;cmd:13.5.11;typ:l;
- 5 dsc:VALUES;typ:Z;
- 6 dsc:SP;cmd:13.9.1;typ:f;

## Legend: INI file text example

Line	Description	Line	Description
1	The name of the first tab	4	Stick Available. The register is displayed as a 4-byte integer.
2	Enable Controller. The register displayed is array 0, index 1. The 13 represents the App type, not the application number. The value is to be displayed as a drop-down box of values: OK or ALARM. The register type is byte integer as indicated by the type.	5	VALUES. Does not display a register value. This causes a header for the next several values to help the user distinguish sections of values.
3	Similar to line 2 but with a data type of 4-byte integer.	6	SP. The register is displayed as a floating point value with default width and precision.

The definitions in this example render the screen shown in the figure below. The Status tab contains the Enable Controller, Alarm, Stick Available, VALUES and SP variables. The value of the SP variable are defined in the INI file.

#### Figure 43: PCCU view of the example INI file

<u>i i</u>	Description	Value
14.0.1	Enable Controller	ok
14.6.1	Alarm	ok
14.5.11	Stick Available	0
	VALUES	
14.9.1	SP	0.0000

# 4 Direct download the resource from the ISaGRAF IDE

Before Download, change the operation mode in PCCU: on the ISaGRAF tab to Developer.

- 1. On the ISaGRAF tab, for the operation mode, Select **developer.**
- 2. Click Send.

Note the Resource Number value in the ISaGRAF screen. This will be used to update the correct application with the downloaded code.

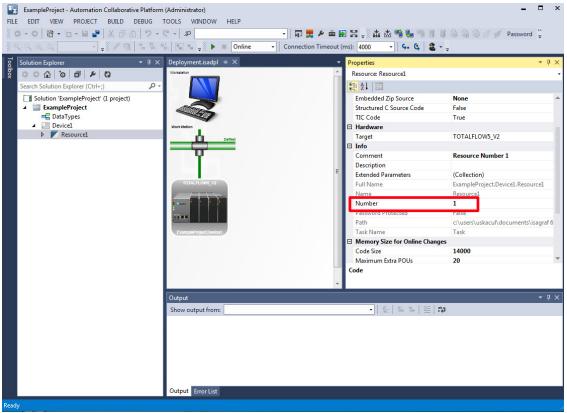
#### Figure 44: Developer mode

ң PCCU32 - [Entry]										
Coperate View Window He	elp			_ 8						
fi 🛅 🖾 💽 💽 🕅	ð 👼		1 🚳 🎼 🚽 🛄 🔍 (							
G5 IEC	k	saGraf Reso	urce List Status							
Totalflow/TCP Totalflow/USB			Description	Value						
Totalflow/COM0:		91.254.1	Isagraf Version	5.41.22						
I I/O System										
⊞-Display			Active IEC Resource							
E-IEC Basic		91.255.254	Resource Number	1						
System Variables		91.254.54	Resource Name	ExampProj						
Symbol Table		91.254.53	Last Message	Copying Resource Files						
IsaGraf										
Operations			Resource State Management							
		91.255.215	Current State	STARTED						
		91.255.50	IEC AutoStart	Off						
		91.255.217	Start/Stop Resource	No						
		91.255.218	Clear Active Resource	No						
		91.255.64	Operation Mode	Production						
				Production						
		91.255.251	Annunciate	Developer						
		91.255.56	Annunciator	Trace						
			Symbol File							
		91.254.0	Symbol File	/tfData/IEC-91/ExampProj/IDS00101						
		91.255.253	Rescan Symbol File	No						
		1		1						
		Re-read	Monitor	Print Screen Save Send Close Help						

The Properties Resource Number (Figure 45) must match the Resource Number on the TOTALFLOW Controller. To see this on the TOTALFLOW Controller go to:

- 1. PCCU Entry Mode. Click IEC Interface.
- 2. Click **ISaGRAF.**
- 3. Click ISaGRAF Tab.
- 4. Click Resource Number (APP.255.254).

# Figure 45: View the properties > resource number



Download to the target device using the ISaGRAF Workbench (IDE):

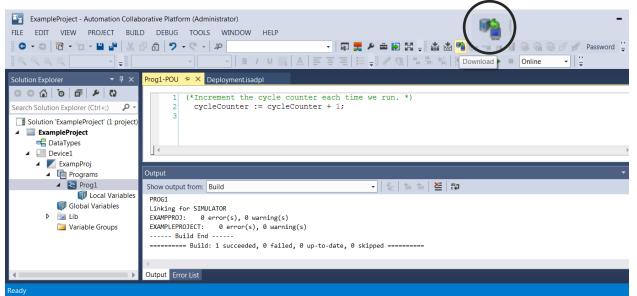
- 1. Select the **View** tab.
- 2. Select Deployment View>Properties.
- 3. Enter the IP Address.

## Figure 46: Enter IP address

	Example	Proje	ct - Automation Collabora	tive Platform (Adminis	strator)					-		×
FILE	EDIT	VIEV	W PROJECT BUILD	DEBUG TOOLS	WINDOW	HELP						
G	• ©	M.	Solution Explorer	Ctrl+Alt+L			•	kaj 🗾 🏓 🏛 🛛	🔊 🔀 🚽 🖄 🐱 🖲	1 W P1 B B		ù ¥.
	0,0	00+	Parameters			Onl	ine 🗸 C	onnection Timesut	(ma): 4000 -	G. G 🏦 - 🚽		
Calut	ing Funda	5	Navigation Window			Ŧ		Deployme	ent view			γ×
	ion Explo		Deployment View			×	Properties Connection:				· · ·	ŕ ^
	○ 🏠	<b></b>	Block Library	Ctrl+Alt+T								•
	ch Solutic		Description Window				□ Info					
	Solution		Document Overview				IPAddress		169.254.0.11			
1	<b>Exam</b>	Ĝ	Error List	Ctrl+ E								
	4 🛄 D	0	Cross Reference Browser	Ctrl+X, Ctrl+N								
	▲ 🗾	<b>E</b> ¢	Output	Ctrl+Alt+O		-						
	-		Start Page									
		â	Toolbox	Ctrl+Alt+X								
			Find Results		•							
	⊳		Toolbars		•							
		5	Full Screen	Shift+Alt+Enter	r 📔							
		đ	All Windows	Shift+Alt+M								
		Θ	Navigate Backward	Ctrl+-								
		Θ	Navigate Forward	Ctrl+Shift+-			IPAddress					
		J,	Properties Window	F4		•						
Outpu	it Error I		Property Pages	Shift+F4								

4. Select the **Download** icon to download the resource.

#### Figure 47: Download



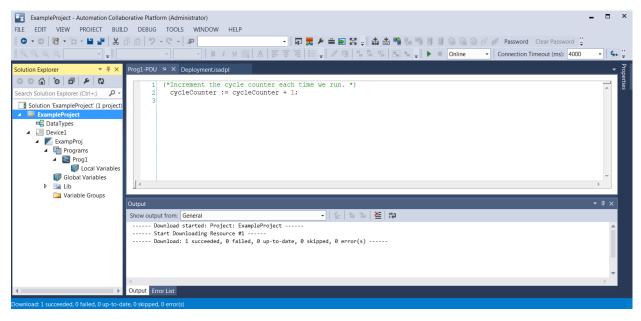
The resource is running, the following alert will open.

#### Figure 48: Download Resource alert

Solution Explore: Solution Expl	ExampleProject - Automation Collaborative Plat FILE EDIT VIEW PROJECT BUILD DEBU			×
Solution Explore: + X Solution Explore: (UH+) + Prog1=POU + X Deployment/sadpl Search Solution Explore: (UH+) + C Solution Expl				
Solution Explorer • # X Search Solution Explorer (Ctrl+> Search Solution Explorer (Ctrl+> Ctrl ExampleProject (I project) CampProj Collocal Variables CampProj Collocal Variables CampProj Collocal Variables Control Co				
Search Solution Explorer (Ctrl+)       I         I Device1       I         Show output from: General       Do you want to stop I?         I Device1       I         I De	े <b>स्                                   </b>	· · · · · · · · · · · · · · · · · · ·	- <u></u> <u></u>	Ţ
Search Solution ExampleProject (1 project) Solution ExampleProject (1 project) Solution ExampleProject (1 project) Solution StampleProject) Solution Variables Solution Variable Groups Utput Show output from: General Solution Collaborative Platform Solution Collaborative Platform Solutio	Solution Explorer 🔻 🕂 🗙 Prog1-PO	DU 🍄 X Deployment.isadpl	-	Prop
Search Solution Explorer (Ctrl+) Solution Explorer (Ctrl+) Solution ExampleProject CataTypes Device1 Solution StampleProject CataTypes CotaVariables Solution Variables Solution StampleProject CotaVariables Solution StampleProject CotaVariables Solution StampleProject Solution StampleProject Start Download started: Project Device1ExampProj is already running. Start Download started: Project Device1ExampProject Device1ExampProj is already running. Start Download started: Project Device1ExampProject Device1ExampPro	ତ ଜ ଜ ବ କ ଦ	1 (*Increment the cycle counter each time we run, *)		ertie
ExampleProject     DetaTypes     Device1     ExampleProj     Local Variables     Global Variables     Global Variable Groups     Ub     Variable Groups     Output     Show output from General     Download started: Pt     Download started: Pt     Start Downloading R     Ves No	Search Solution Explorer (Ctrl+;)		<b>^</b>	105
DataTypes Device1 Programs I Cocal Variables Variable Groups Output Show output from: General Show output from:	Solution 'ExampleProject' (1 project)	3		
<ul> <li>Devici</li> <li>Example j:</li> <li>Programs</li> <li>Coldad Variables</li> <li>Global Variable Groups</li> <li>Variable Groups</li> <li>Output</li> <li>Resource ExampleProject.Device1.ExampProj is already running.</li> <li>Download started: Program at the started at th</li></ul>				
<ul> <li>ExampProj</li> <li>Programs</li> <li>Programs</li></ul>				
Prog1   Coloput   Show output from:   General   Output   Show output from:   General   Vers   No				
Image: Construction of the second				
Global Variables   Lib   Variable Groups     Output   Show output from: General   Ownload started: Pr   Start Downloading Re     Ves     Output     Output from: General				
Variable Groups       Output     Resource ExampleProject.Device1.ExampProj is already running.       Show output from:     General       Ownload started:     Pr       Start Downloading Re     Yes       Output     Output		Advantage Callaboration Medican	-	
Output     Resource ExampleProject.Device1.ExampProj is already running.       Show output from: General     Do you want to stop it?       Download started: Pr     Do you want to stop it?       Start Downloading Re     Yes       Ves     No		Automation Collaborative Platform	Þ.	
Show output from: General Council Show				
Show output from General Download started: Pr Start Downloading Re Ves No 		Do vou want to stop it?	Ψ×	
Start Downloading Re     Yes No		utput from: General		
Output Error List				
		Yes No		
	Output	Error List		
Start Downloading Resource #1 Ln 2 Col 1 Ch 1 INS	Start Downloading Resource #1	Ln 2 Col 1 Ch 1	INS	

5. Click Yes.

#### Figure 49: Result of a successful resource download



# 5 Annunciators

To trace the progress of the running resource, view the **ISaGRAF tab**>**Annunciate** feature (255.25.1).

Figure	50: Annunciate feature
	17

E. PCCU32 - [Entry]				х
Operate View Window Hel	p		-	e ×
fii 😇 🔟 🔽 💽 🕅		] 🚳 🛱 📑 🛄 🗰 🧃	Ø	
⊡-G5 IEC	IsaGraf Reso	urce List Status		
Totalflow/TCP Totalflow/USB		Description	Value	
Totalflow/COM0:	91.254.1	Isagraf Version	5.41.22	
I I/O System				
Display		Active IEC Resource		
EC Basic	91.255.254	Resource Number	1	
System Variables	91.254.54	Resource Name	ExampProj	
Symbol Table	91.254.53	Last Message	Copying Resource Files	
IsaGraf				
		Resource State Management		
	91.255.215	Current State	STARTED	
	91.255.50	IEC AutoStart	Off	=
	91.255.217	Start/Stop Resource	No	
	91.255.218	Clear Active Resource	No	
	91.255.64	Operation Mode	Production	
	91,255,251	Annunciate	R	
	91.255.56	Annunciator	Trace	-
				-
		Symbol File		
	91.254.0	Symbol File	/tfData/IEC-91/ExampProj/IDS00101	
	91.255.253	Rescan Symbol File	No	
				*
	Re-read	Monitor	Print Screen Save Send Close Help XHelp	

## 5.1 Annunciation to view the state on RMC display

The Annunciator option has several modes that display application statues to the RMC board's display.

Manual: Displays the character selected in the Annunciate field

**Trace**: Displays pre-defined characters as shown in <u>Table 5</u>. These characters are associated with different status in the application.

**Auto**: Building on Trace, after the application is running, a "breathing" asterisk is shown on the RMC display (highest CPU usage.)

Indicator	Description
К	Killed
С	Cleared Active Resource
н	Halting Signal on Annunciator
S	Starting Resource
0-6	Progress of Resource Connect/Disconnect
Α	Active Resource
F	Failed to Connect Resource
x	Resource Disconnected

#### Table 5: Pre-defined characters:

Indicator	Description
R	Running IEC Application
F+ (numeric value)	(Numeric Value)' = Failed Connection Working toward Recovery
N	Not Running
I	Idle
r	Requesting PARTNUMBER from IEC application at startup.
м	Resource files are missing and cannot be load
т	The resource process has terminated unexpectedly or is not reachable
Р	State change request is pending
U	App has undecided license allocation and is not running.
D	App is disabled due to license allocations and is not running.
?	Unknown Kernel State

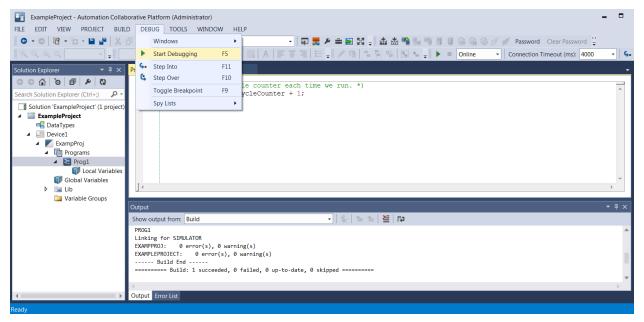


**IMPORTANT NOTE:** Other characters which are not listed in this document may appear due to internal state management.

## 6 Debug mode

- 1. Select the **Debug** tab from the main toolbar.
- 2. Select Start Debugging.

#### Figure 51: Debug tab



After debug is completed, the system is ready for use.

## 7 Program example

An example of an ISaGRAF program is included in this section for reference. The program uses ST (structured text) language and implements a hysteresis on 10 inputs. An input value larger than the HiHiLimit will set the output to a 1 and will reset the output when the input is below the HiLimit. The example includes an event log function that will record the Cold start, Warm Start, and change in the output value.

## 7.1 Main program

(\* setup initial CFG values - first time. They will be read back from config file after a first time.\*)

```
IF (Cold = FALSE) THEN
```

```
Enable := FALSE;
```

```
HiHiLimit := 100.0;
```

HiLimit := 90.0;

```
(* clear event log on first run *)
```

FOR pos := 1 TO ANY\_TO\_DINT(Event\_Log\_Size) DO

```
Event_Code[pos] := State_Unused;
```

```
Event_Input[pos] := 0;
```

Event\_Value[pos] := 0.0;

Event\_Stamp[pos] := 0;

```
END_FOR;
```

ReturnStatus := Event\_Log(State\_Cold, 0, 0.0);

Cold := TRUE;

```
END_IF;
```

```
(* setup variables after a warm start *)
```

```
IF (Warm = False) THEN
```

FOR pos := 1 TO ANY\_TO\_DINT(Input\_Size) DO

```
State[pos] := State_OK;
```

Output[pos] := 1;

```
END_FOR;
```

ReturnStatus := Event\_Log(State\_Warm, 0, 0.0);

```
Warm := TRUE;
```

END\_IF;

```
(* if not enabled - don't execute main body *)
```

```
IF (Enable) THEN
```

```
FOR pos := 1 TO ANY_TO_DINT(Input_Size) DO
```

```
IF (State[pos] = State_OK and Value[pos] > HiHiLimit[pos]) THEN
```

```
State[pos] := State_HiHi;
```

ReturnStatus := Event\_Log(State, pos, Value);

```
Output[pos] := 0;
```

```
ELSIF (State[pos] = State_HiHi and Value[pos] < HiLimit[pos]) THEN
```

```
State[pos] := State_OK;
ReturnStatus := Event_Log(State, pos, Value);
```

Output[pos] := 1;

END\_IF;

END\_FOR;

END\_IF;

## 7.2 EVENT\_LOG function (Code, Input, Value)

(\* Move all data down by one row – start at bottom, filling from above \*)

FOR pos := ANY\_TO\_DINT(Event\_Log\_Size) TO 2 BY -1 DO

Event\_Code[pos] := Event\_Code[pos - 1];

Event\_Input[pos] := Event\_Input[pos - 1];

Event\_Value[pos] := Event\_Value[pos - 1];

Event\_Stamp[pos] := Event\_Stamp[pos - 1];

END\_FOR;

(\* place new log in first row \*)

Event\_Code[1] := Code;

Event\_Input[1] := Input;

Event\_Value[1] := Value;

Event\_Stamp[1] := DateTime;

Event\_Log := TRUE;

## 7.3 Variables

The tables below describe the defined variables and words used by the example program shown above.

Name	Туре	Retain	Scope	Dim	I/O	Init
HiHiLimit	Real	Yes	Global	[110]	rw	
HiLimit	Real	Yes	Global	[110]	rw	
Value	Real	Yes	Global	[110]	rw	
Cold	bool	Yes	Global		rw	
Warm	bool	No	Local		rw	
DateTime	uint32	No	Global		R	
Enable	uint32	Yes	Global		rw	
Event _Log_Size	uint32	No	Global		rw	30
Event_Input	uint32	Yes	Global	[130]	rw	
Event_Code	uint32	Yes	Global	[130]	rw	

#### **Table 6: Defined variables**

Event_Stamp	uint32	Yes	Global	Global [130]		
Input_Size	uint32	No	Global	Global		10
Pos	uint32	No	Main		rw	
Pos	uint32	No	Event_Log		rw	
ReturnStatus	uint32	No	Main		rw	
Output	uint32	No	Global	[110]	W	
State	uint32	Yes	Global	[110]	rw	
DateTime	uint32	No	Global		R	
PartNumber	String	No	Global	Global		2509999
NumberRegisters	String	No	Global		rw	1

#### Table 7: Defined Words

Define words	Value	
State_OK	0	
State_HiHi	1	
State_Cold	2	
State_Warm	3	
State_Unused	4	

## 8 Read and write efficiency

When developing your own custom Totalflow application with ISaGRAF Workbench, there are several methods of passing data between your ISaGRAF application and the other Totalflow applications running on the same controller. Each of these communication methods has different advantages. These methods vary widely in how much CPU load is required to process the data request. The following is a comparison of these communication methods.

## 8.1 Background

Your custom ISaGRAF application runs as a separate process along-side the built-in Totalflow applications. The Totalflow.exe process contains all the built-in Totalflow applications (AGA3, I/O Interface, Trend, Operations, etc.).

All the applications listed on the PCCU Application/License Management tab are contained inside the Totalflow.exe process.

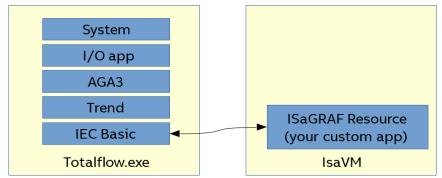
Entry]										
■ <u>Operate</u> <u>View</u> <u>W</u> indow <u>H</u> elp										- 5
1 🖾 🔁 🐼	🛄 2, 🧇									
-TOTALFLOW -Communications Totalflow/TCP Totalflow/USB	Station Setup Application			Transfer to	Device	Device Credits				
	Cre	dit Type Amount	Ту	Transfer to pe nount	o Key V	Credit Typ General(non-rem General(removal IEC Basic CO2(NIST)	ovable) 1 ble) 0 1	Surplus/Deficit 5 0 0 0 0 0 0	Ŷ	
		Load		Tr	ansfer	(spare) IEC Tier 1 IEC Tier 2	- 0	- 2 0	Ŷ	
	App# Name/ID	Туре	Revision	Station	Directory	License Status	Restart	D	elete App	
	0 System	System	2105252-005		Dir = \	Enable				
	1 Totalflow/TCP	Communications	2101348-005		Dir = \Comm-1	Enable				
	2 Totalflow/USB	Communications	2101340-005		Dir = \Comm-2	Enable				
	3 Totalflow/COM0	Communications	2101340-005		Dir = \Comm-3	Enable				
	7 I/O Interface	I/O Interface XSeries	2105253-002		Dir = \TFIO-A	Enable				
	8 Display	Display XSeries	2103137-002		Dir = \Display	Enable				
	11 TOTALFLOW	AGA-3 Measurement	2101306-008		Dir = \AGA3-1	Enable				
	91 IEC Basic	IEC Basic	2105829-003		Dir = \IEC-91	Enable				
	95 Trend System	Trend System	2101309-002		Dir = \Trend	Enable				
I	241 Operations	Operations	2101320-005		Dir = \Operations	Enable				
	<									>
	<u>R</u> e-read		Add Ap	p	Credit/Ap	p Info		<u>S</u> end C	lose	Help
leady					#Polls:	55 #Errors:	0 Conne	cted to 192.168.1.21	Logir	n: user

Figure 52: Application/License Management tab

The IsaVM process runs along-side Totalflow.exe on the same controller. IsaVM is the interpreter that runs your custom ISaGRAF application. The ISaGRAF application must send messages to the Totalflow process to request register values.

The IEC Basic app acts as a communication interface to pass register values between your custom ISaGRAF application and the Totalflow applications.

#### Figure 53: Communication between ISaGRAF and Totalflow apps



There are several communication methods available to pass data between your custom ISaGRAF application and the Totalflow.exe applications:

- I/O Device Inputs and Outputs
- Get\*() and Set\*() function blocks
- Read and write ISaGRAF registers from Totalflow.exe applications

For instructions about how to configure each of these read/write methods, see section <u>3, Interface</u> with the IEC application.

Following is a brief description of each of these methods.

### 8.2 ISaGRAF I/O Device

The ISaGRAF I/O Device is the most efficient method of reading register values from the Totalflow application into your custom ISaGRAF application.

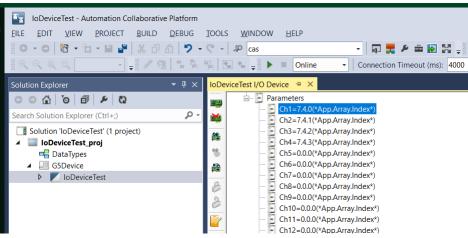
The ISaGRAF I/O Device can read 64 Totalflow registers in a single message request.

Advantages:

— The most efficient way to read and write registers from Totalflow applications. Disadvantages:

— No programmatic control. Read and write is automatic and cannot be disabled.

#### Figure 54: I/O Device



## 8.3 Get\*() and Set\*() Function Blocks

Get\*() and Set\*() function blocks are called from within your ISaGRAF code. These function blocks provide some capability not available with I/O Device, but require much more CPU overhead.

Advantages:

- The App/Array/Index inputs can be variables.
- The function can be called conditionally only when required.

Disadvantages:

High CPU cost

#### Figure 55: GetRealTest

getRealTest - Automation Collaborative Platform         FILE       EDIT       VIEW       PROJECT       BUILD       DEBUG         O       O       IS       I       III       IIII       DEFUG         O       O       IS       I       IIIIIII       IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	FQRMAT IOOLS WINDOW HELP 
Solution Explorer  Search Solution Explorer (Ctrl+) Search Solution 'getRealTest (1 project)  GoBevice  GoBevice  GoBevice  Programs  Programs  Global Variables  Global Variable Groups	Prog1-POU + ×

#### Figure 56: Block selector

Search				Show Paramete
Name	Туре	Category 🔺	Commen	t
UNLOCK_CPU	SFL	System Operations	Unlocks the CPU	
LOCK_CPU	SFI	System Operations	Locks the CPU	
TP	SFE	Time Operations	Pulse timing	
TON	SFE	Time Operations	On-delay timing	
TOF	SFE	Time Operations	Off-delay timing	
CURRENT_ISA_DA	SFL	Time Operations	Gets the current date	
SUB_DATE_DATE	SFI	Time Operations	DATE subtraction	
GetDINT	CFI	Totalflow	Read a 32 bit Signed Integer Register	
GetLREAL	CFI	Totalflow	Read a 64 bit Floating Point Register	
GetREAL	CFI	Totalflow	Read a 32 bit Floating Point Register	
SetULINT	CFI	Totalflow	Read a 64 bit Unsigned Integer Register	
SetUDINT	CFI	Totalflow	Read a 32 bit Unsigned Integer Register	
SetSTRING	CFI	Totalflow	Write a 64 Byte String Register	
SetREAL	CFI	Totalflow	Write a 32 bit Floating Point Register	
SetLREAL	CFI	Totalflow	Write a 64 bit Floating Point Register	
SetLINT	CFI	Totalflow	Write a 64 bit Signed Integer Register	
SetDINT	CFI	Totalflow	Write a 32 bit Signed Integer Register	[
GetLINT	CFI	Totalflow	Read a 64 bit Signed Integer Register	=
GetUDINT	CFI	Totalflow	Read a 32 bit Unsigned Integer Register	-
GetSTRING	CFI	Totalflow	Read a 64 Byte String Register	
				•

## 8.4 Totalflow apps read IsaVM registers

The following example shows an Operations app reading registers from the ISaGRAF application (app slot 91).

This method has a very high CPU cost. Its use should be minimized if possible. When data must be read from the ISaGRAF application (for example, status to be displayed on SCADA or HMI), it is recommended to use an intermediate Holding Registers application. Configure the HMI to read the status from a Holding Register, and use ISaGRAF I/O Device to write the status to the Holding Register.

Advantages:

Configuration changes can be made easily without re-compiling the ISaGRAF application.

Disadvantages:

- Very high CPU cost

Entry]									- 0	×
Derate View Window Help										- 8 ×
1 🖻 🖪 💽 🚳	🔟 🔬									
-TOTALFLOW -Communications	Periodic Flo	at								
Totalflow/TCP										_
Totalflow/USB		Description	Туре	Interval	Operation	R1	R2	Output		~
Totalflow/COM0:	241.1.0	Operation 0	Interval	00:00:01	R1 -> Out	91.9.1	0.0.0	241.200.0		
I/O System	241.1.1	Operation 1	Interval	00:00:01	R1 -> Out	91.9.2	0.0.0	241.200.1		
Display	241.1.2	Operation 2	Interval	00:00:01	R1 -> Out	91.9.3	0.0.0	241.200.2		
Holding Registers	241.1.3	Operation 3	Interval	00:00:01	R1 -> Out	91.9.4	0.0.0	241.200.3		
IEC Basic	241.1.4	Operation 4	Interval	00:00:01	R1 -> Out	91.9.5	0.0.0	241.200.4		
Trend System	241.1.5	Operation 5	Interval	00:00:01	R1 -> Out	91.9.6	0.0.0	241.200.5		
-Plunger	241.1.6	Operation 6	Interval	00:00:01	R1 -> Out	91.9.7	0.0.0	241.200.6		
Operations	241.1.7	Operation 7	Interval	00:00:01	R1 -> Out	91.9.8	0.0.0	241.200.7		
Capacity	241.1.8	Operation 8	Interval	00:00:01	R1 -> Out	91.9.9	0.0.0	241.200.8		
Operations-1	241.1.9	Operation 9	Interval	00:00:01	R1 -> Out	91.9.10	0.0.0	241.200.9		
	241.1.10	Operation 10	Interval	00:00:01	R1 -> Out	91.9.11	0.0.0	241.200.10		

#### Figure 57: Reading ISaGRAF app from Totalflow operations app

#### 8.5 Totalflow apps write IsaVM registers

The following example shows an Operations app writing registers to the ISaGRAF application (app slot 91). Software architecture limits these Totalflow-to-ISaGRAF writes to 5 registers-per-second.

This method should be avoided if possible. If data must be written to the ISaGRAF application (for example, commands from SCADA or HMI), it is recommended to use an intermediate Holding Registers application. Configure the HMI to write its command to a Holding Register, then use I/O Device to read the Holding Register into the ISaGRAF application.

Advantages:

— Configuration changes can be made easily without re-compiling the ISaGRAF application. Disadvantages:

Limited to 5 registers-per-second

### Figure 58: Writing ISaGRAF from Totalflow operations app

Operate View Window He	Halp	- 8 ×
		= 0 A
Communications	Periodic Float	
Totalflow/TCP		
Totalflow/USB	Description         Type         Interval         Operation         R1         R2         Output           242.1.0         Operation 0         Interval         00:00:01         R1 -> Out         242.200.0         0.0.0         91.9.1	
Totalflow/COM0:	242.1.1         Operation 1         Interval         00:00:01         R1 > Out         242.200.1         0.0.0         91.9.2	
i I/O System ⊡ Display	242.12 Operation 2 Interval 00:00:01 R1 > Out 242.200.2 0.0.0 91.9.3	
Holding Registers	242.1.3 Operation 3 Interval 00:00:01 R1 -> Out 242.200.3 0.0.0 91.9.4	
IEC Basic	242.1.4 Operation 4 Interval 00:00:01 R1 > Out 242.200.4 0.0.0 91.9.5	
Trend System	242.1.5 Operation 5 Interval 00:00:00 None 0.0.0 0.0.0 0.0.0	
Plunger	242.1.6 Operation 6 Interval 00:00:00 None 0.0.0 0.0.0 0.0.0	
Operations	242.1.7 Operation 7 Interval 00:00:00 None 0.0.0 0.0.0 0.0.0	
Operations-1     Capacity	242.1.8 Operation 8 Interval 00:00:00 None 0.0.0 0.0.0 0.0.0	
- capacity	242.1.9 Operation 9 Interval 00:00:00 None 0.0.0 0.0.0 0.0.0	
Ready	Reviewal Monitor Show editable fields Print	Screen Save Send Close Help X Help ★ s 289 #Errors: 0 Connected to 192.163.1.21 Login: user

#### 8.6 Comparison

These communication methods vary widely in how much CPU load is required to process the data request. The following table shows the CPU cost-per-register for each method.

#### Table 8: CPU cost for each method

Method	CPU % / register	Notes
I/O device	0.004 %	64 registers per I/O device
Get*() and Set*() function blocks	0.1 %	
Totalflow app read from IsaVM	0.2 %	
Totalflow app write to IsaVM	N/A	5 register limit

To find the CPU load caused by reading registers into your ISaGRAF application:

CPU load = (# of registers to be read) x (CPU % / register)

Example 1. If you read 128 registers using GetReal() function blocks:

- CPU load = 128 x 0.1 %
- CPU load = 12.8 %

Example 2. If you read 128 registers using the I/O Device:

- CPU load = 128 x 0.004 %
- CPU load = 0.51 %

### 8.7 Run-time diagnostics

RMC flash 2105457-031 and newer includes message and register counters. To view the counters, launch PCCU and go to **Entry mode** > **IEC Basic** > **ISaGRAF** > **Timings** tab.

The Msgs/sec column shows the number of messages transferred between the IsaVM (ISaGRAF interpreter) and Totalflow.exe processes. More messages means a higher CPU load. To reduce CPU load, use the ISaGRAF I/O Device to send 64 registers per message.

#### PCCU32 - [Entry] Image: Operate View Window Help 10 🛅 🔁 💽 🧠 🛄 🐊 🧇 - TOTALFLOW IsaGraf Resource List Status Timings Communications I/O System Msg Timing (msec) Reg Timing (msec) Msgs/sec Regs/sec Comments Flow Measurement Operation Display 91.255.49 Time Events Turn on and off timing data collection Both Holding Registers 0.10.1 CPU % 2.438246 E IEC Basic System Variables IEC requests for Totalflow Registers 91.34.45 Gets from IEC Symbol Table 91.34.46 Sets from IEC IEC Setting for Totalflow Registers IsaGraf 0 91.34.50 Get IEC Variabl Totalflow Getting IEC Variable Trend System 91.34.51 Set IEC Variable Totalflow Setting IEC Variable -Plunger • Operations Derations-1 Re-read Monitor Show editable fields Print Screen Save Send Close Help XHelp 🖉 #Polls: 111 #Errors: 0 Connected to 192.168.1.21 Login: user Ready

#### Figure 59: Message timing



# Typographical conventions

Element	Convention	Example
Cross-reference to a figure or table in the document	Hyperlink to the figure or table in blue, with underline.	See <u>Figure 2</u> .
Cross-reference to a specific section in the document	Hyperlinks to sections referenced throughout the document appear in blue, with underline.	See <u>Section 2.</u>
Cross-reference to another document or website	Hyperlink to the website appears in blue, with underline	See the RMC user manual at <u>abb.com.</u>
Greater-than character (>)	Indicates that the following item is an additional menu selection	From the menu, locate and select Calibrate > Diff. Press. Sensor > Calibration Units > Edit.
Name of selection buttons, menus, or navigation tree items in instructions that the user will click	Bold text, and the capitalization agrees with the name as displayed on the user interface	Click the <b>Monitor</b> tab and select the <b>Add Advanced Setup</b> tab.
Programs, including utility and accessory programs	Title capitalization	Microsoft Word
URL	All lowercase for a fully specified URL	www.abb.com/totalflow
User input	Bold and lowercase, unless case sensitive. If the user-input string contains placeholder text, the text is in small caps.	Type <b>config</b> . SMALL CAPS FOR PLACE HOLDERS

## **Contact us**

ABB Inc. Measurement & Analytics Quotes: totalflow.inquiry@us.abb.com Orders: totalflow.order@us.abb.com Training: totalflow.training@us.abb.com Support: upstream.support@us.abb.com +1 800 442 3097 (opt. 2) www.abb.com/upstream

Main Office - Bartlesville 7051 Industrial Blvd Bartlesville, OK 74006 Ph: +1 918 338 4888

Texas Office - Houston 3700 W. Sam Houston Parkway S., Suite 600 Houston, TX 77042 Ph: +1 713 587 8000 Kansas Office - Liberal 2705 Centennial Blvd Liberal, KS 67901 Ph: +1 620 626 4350

Texas Office – Odessa 8007 East Business 20 Odessa, TX 79765 Ph: +1 432 272 1173 Texas Office – Pleasanton 150 Eagle Ford Road Pleasanton, TX 78064 Ph: +1 830 569 8062

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents - in whole or in parts – is forbidden without prior written consent of ABB.

Windows<sup>®</sup> is a registered trademark of Microsoft.

ISaGRAF<sup>®</sup> is a registered trademark of ICS Triplex ISaGRAF Inc.

Copyright<sup>©</sup> 2020 ABB all rights reserved