

Win-GRAF EMP9K Runtime

User Manual

(Version 1.1)

for EMP-9xx8-xx Series





WARRANTY

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

WARNING

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

COPYRIGHT

Copyright © 2024 by ICP DAS. All rights are reserved.

TRADEMARK

Names are used for identification only and may be registered trademarks of their respective companies.

CONTACT US

If you have any questions, please feel free to contact us via email at:

service@icpdas.com

service.icpdas@gmail.com

Revision

Revision	Date	Description	Author
1.0	27.06.2022	Initial version	M. K.
1.1	29.03.2024	Describes the function of Win-GRAF Runtime Utility Version 1.0.0.3	M. K.

Contents

1	INTRODUCTION.....	5
2	RUN WIN-GRAF RUNTIME.....	6
3	RUNTIME CONFIGURATION.....	6
3.1	WIN-GRAF RUNTIME UTILITY.....	6
3.1.1	<i>Communication Port Setting.....</i>	<i>7</i>
3.1.2	<i>Task Number Setting.....</i>	<i>9</i>
3.2	CONFIGURATION PROCEDURE.....	9
4	DOWNLOAD APPLICATION.....	14
5	MULTITASKING.....	19
5.1	TASK SETTING.....	19
6	EWf FUNCTION.....	23
7	ACTIVATE “LOCK SCREEN”.....	25
8	PLATFORM SPECIFIC LIBRARY.....	28
8.1	BUZZER.....	28
8.1.1	<i>PAC_BuzzerBeep.....</i>	<i>28</i>
8.1.2	<i>PAC_BuzzerStop.....</i>	<i>29</i>
8.2	ROTARY SWITCH.....	29
8.2.1	<i>PAC_GetRotaryId.....</i>	<i>29</i>
8.3	LED CONTROL.....	30
8.3.1	<i>PAC_SetLed.....</i>	<i>30</i>
8.4	SERIAL NUMBER.....	31
8.4.1	<i>PAC_GetSerialNo.....</i>	<i>31</i>

1 Introduction

This manual is an extension to the '*Win-GRAF Workbench User Manual*' and describes the configuration procedure necessary to develop applications for the EMP-9xx8-xx motion control series. In this manual the name '*EMP9K*' will be used for the EMP-9xx8-xx platform. The Win-GRAF EMP9K Runtime version is a SoftPLC runtime specifically developed for the EMP9K motion control series provided by ICPDAS. The PLC application for the runtime has to be developed using the Win-GRAF Workbench, which is not included in the runtime package and has to be ordered separately. The workbench supports all five IEC-61131-3 programming languages as well as multitasking programming with priority settings.

The motion controller uses EtherCAT as its main communication protocol and the motion control commands are implemented according to the PLCopen defined standard. Single axis and coordinated multi-axis PLCopen commands are part of the Win-GRAF library. The integrated EtherCAT master supports all standard EtherCAT slaves from simple IO modules, encoder, stepper controller to servo drives. The user manual "PLCopen_User_Manual" provides further information on the EtherCAT network configuration and the behavior of the PLCopen commands.

All standard Modbus protocols (TCP, RTU, ASCII) are supported. The Modbus functions are introduced in the "Modbus" manual.

The "OPC UA Server_User_Manual" contains step-by-step configuration examples for the integrated OPC UA server.

HMI software provided by ICPDAS such as eLogger and Indusoft can be installed directly on the EMP9K Windows OS IoT system. The 'HMI_User_Manual' describes the data exchange mechanism and the required communication configuration to establish data exchange between Win-GRAF runtime and HMI. In addition the manual describes how to allow external program (e.g. LabVIEW, C#, C++) to exchange data with the PLC application.

All manuals can be directly accessed via the '*Help\Tutorial*' menu of the Win-GRAF workbench. ICPDAS provides a free workbench trial version for download on its website.

2 Run Win-GRAF Runtime

The Win-GRAF runtime and its configuration utility forms part of one execution file '*WGrafRT_EMP9K.exe*' which by default is installed in the directory:
'T:\Win-GRAF\Rt xx.x.x\'

As the '*Win-GRAF Runtime Utility*' is part of the runtime it therefore can not be exited without terminating the runtime.

By default the EMP9K will automatically start the '*Win-GRAF EMP9K Runtime Utility*' and runtime after power on.

Only one runtime process is allowed to execute on the EMP9K at a time.

3 Runtime Configuration

The Win-GRAF EMP9K Runtime supports multitasking programming. Up to 32 tasks are supported, but for optimal performance, no more than four tasks should be used for an application. By default the runtime is set to support four tasks. The number of supported tasks can be increased or decrease using the '*Win-GRAF EMP9K Runtime Utility*'.

3.1 Win-GRAF Runtime Utility

The main purpose of the utility is to configure, start and stop the PLC application. In addition, it monitors the status of each task, such as: E.g. the task execution time, priority settings, number of execution cycles, etc. (Figure 1). In the event of an error, the error type is displayed in the "Status" tab. The utility allows direct deletion of compiled PLC application execution files.

Task #	Name	Thread Status	Task Status	Set Cyc Time [ms]	Exe Time [ms]	Max Exe Time [ms]	Exe Cycles	Overflow	Task Priority
Task 1:	MainTask	RUNNING	RUNNING	10.000	0.018	0.334	5581	0	15
Task 2:	Task2	RUNNING	RUNNING	1000.000	0.015	0.314	88	0	1
Task 3:	Task3	RUNNING	RUNNING	1.000	0.005	0.581	13910986	0	15
Task 4:	Task4	RUNNING	RUNNING	1.000	0.003	0.321	13910640	0	15

APPLICATION: **RUNNING** RUNTIME LICENSE: VALID

Figure 1: Win-GRAF Runtime Utility

The utility is also responsible for checking the license status. It will clearly indicate whether the runtime license is invalid or the PLC application has been compiled by an workbench trial version:

- PLC application compiled with the workbench trial version will only run for about 20 minutes
- A runtime without a valid license will automatically terminate its execution after 20 minutes.

3.1.1 Communication Port Setting

In order to establish a TCP/IP communication the workbench needs to know the IP address and the socket port number of the target runtime. Two PLC applications project types are supported by the EMP9K platform:

- Single-task project
- Multi-tasking project

For single-task projects, the default communication port for downloading and debugging the application is set to 1100. The default port number can be changed via the '*Win-GRAF EMP9K Runtime Utility*'.

In multi-tasking projects, each PLC task communicate via its own port number. By default, the port number of the '*Main task*' is set to 1100. The port number of each subsequent task is incremented by one in task number order: port 1101 for Task2, port 1102 for Task3, etc.

If necessary, the port number of the '*Main task*' and thus the other tasks can be changed via the '*Win-GRAF EMP9K Runtime Utility*' by setting the '*Main Task Port*'

value(Figure 2). Confirm and save the setting with 'Set' and 'File\Save'.

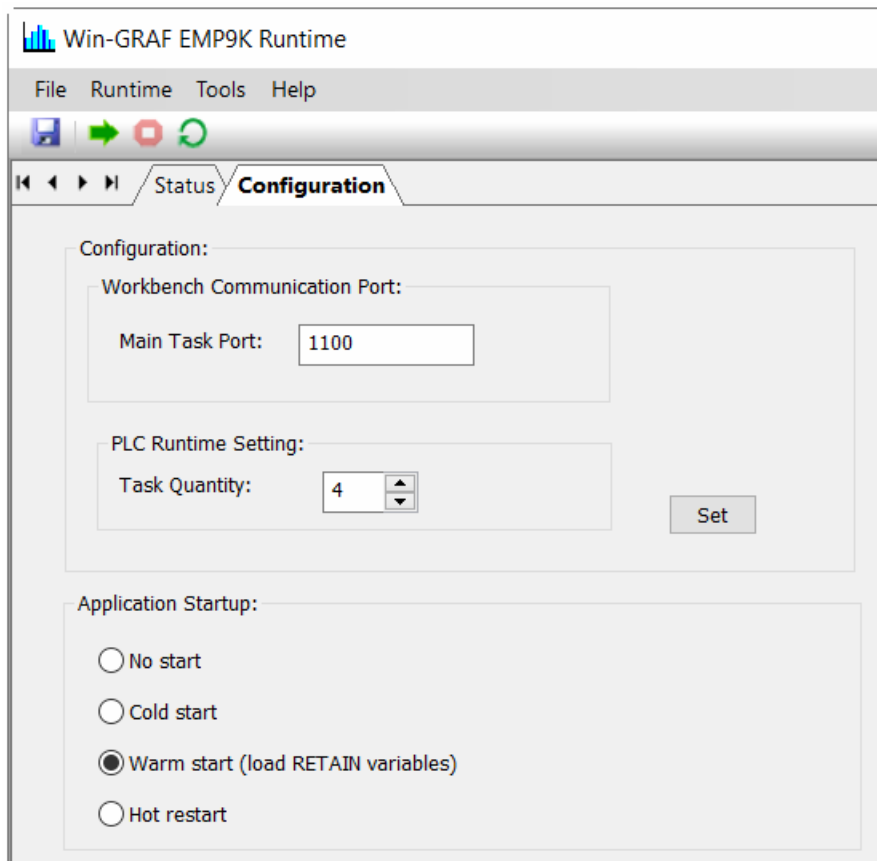


Figure 2: Utility configuration interface

Make sure the workbench Ethernet TCP/IP port setting (Figure 3) is set to the 'Main Task Port', otherwise the connection to the EMP9K runtime cannot be established.

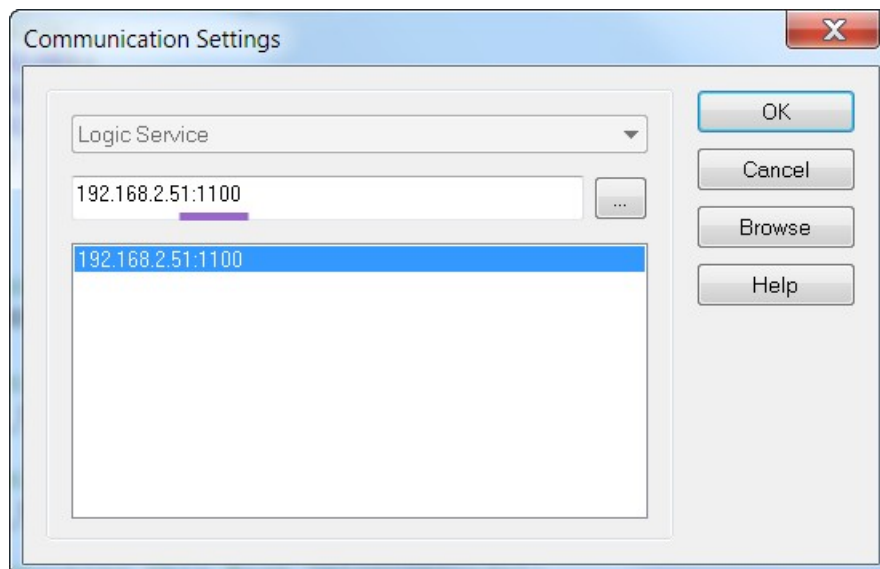


Figure 3: Workbench communication setting

3.1.2 Task Number Setting

The EMP9K runtime supports a maximum of 32 tasks. By default, the task number is limited to four tasks, which means that a PLC application can use fewer than four tasks, but cannot exceed this number. The user can change the supported task count to a higher or lower limit by directly editing a value in the '*Task Quantity*' box (Figure 2) of the Runtime Utility. Confirm and save the setting with '*Set*' and '*File\Save*'.


3.2 Configuration Procedure

In order for the compiled PLC application to be downloaded and executed from the workbench, both the communication settings and the configuration of the number of tasks in the workbench and the runtime must be identical. Next, the basic configuration process is shown:

Step 1: Start the '*Win-GRAF EMP9K Runtime*':

1. Double click the '*Win-GRAF EMP9K Runtime*' icon on the desktop to start the Win-GRAF Runtime utility.



Step 2: Stop the PLC application execution if it is running: Click the stop command 

Step 3: Set the communication parameters between workbench and runtime:

1. **Runtime configuration:** Set the TCP/IP port number used by the workbench to download and debug the PLC application. The runtime server listen on all Ethernet ports of the platform. It is suggested to use the default port number of 1100 for the main task. Only change this port number if it conflicts with the port number of another program running on the EMP9K.

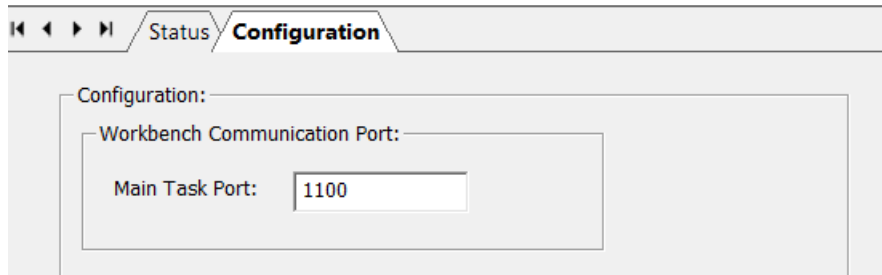


Figure 4: Runtime TCP/IP port number for workbench communication

2. Workbench configuration: Set the IP address and socket port number of the target runtime.

- Select '*Tools/Communication Settings...*' from the menu
- Edit IP address and port number of the target runtime. Both parameters have to be separated by a colon. Only Ethernet TCP/IP communication is being supported.
- Click "OK"

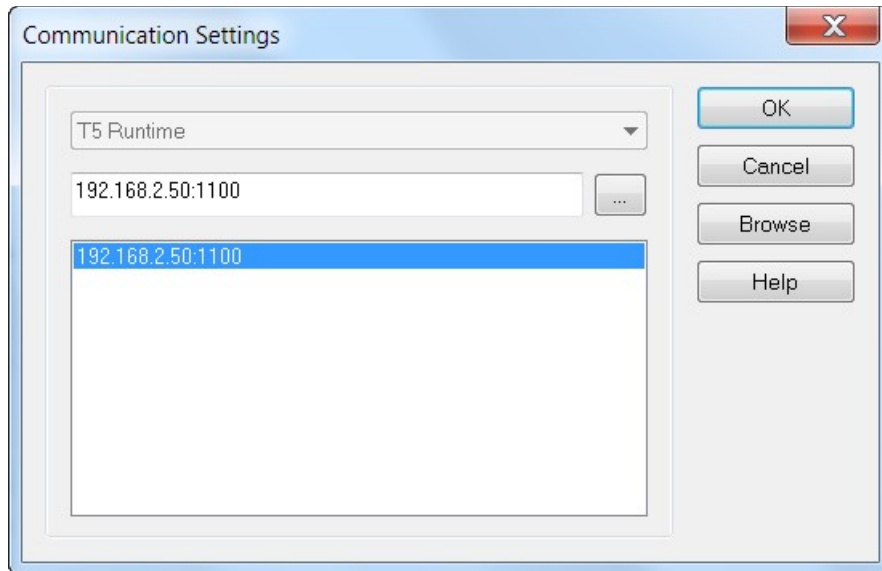


Figure 5: Workbench communications setting

The current communication setting is being displayed at the bottom of the screen. The setting can be directly modified by double clicking the IP address in the status bar.

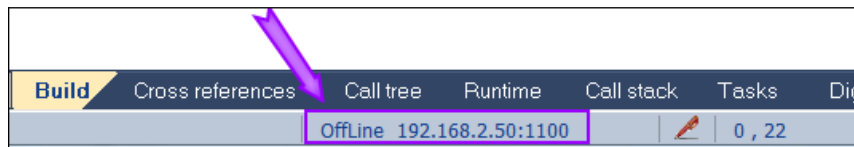


Figure 6: Workbench status bar

Step 4: Set the maximum number of tasks supported by the runtime.

1. **Runtime configuration:** The default runtime setting supports up to four tasks and meets the requirement for most applications. If necessary enter a new task value for the task quantity. Click 'Set' and save the setting.

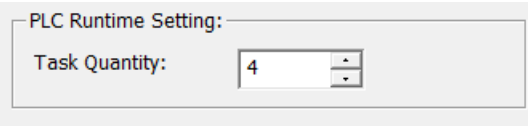


Figure 7: Supported runtime task quantity setting (Runtime Utility)

2. **Workbench setting:**

- (i) Single-Tasking project

- i. On the 'Start Page' click 'Create Singletask Project' to open the single-task project wizard.
 - ii. Select 'Project (WG)'
 - iii. Enter project destination folder and name.
 - iv. Click 'Next'
 - v. Select 'EMP-9000-MPAC' controller or click on the 'Auto-Detect' button to determine the controller type.
 - vi. Select programming language
 - vii. Confirm the setting with 'Finish'. A new single project is created.

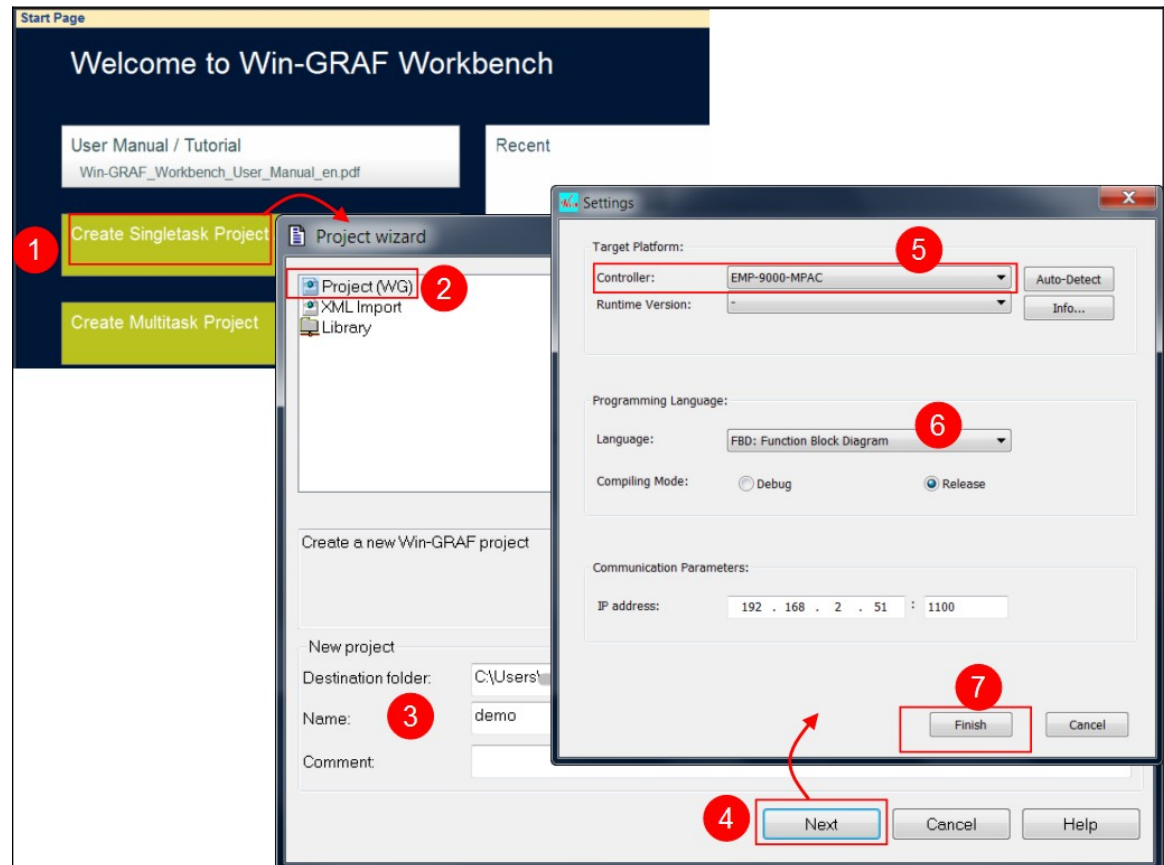


Figure 8: Create a new single-task Workbench project

(ii) Multi-Task project:

- i. On the 'Start Page' click 'Create Multitask Project' to open the multi-task project wizard.
- ii. Enter project destination folder.
- iii. Enter project name.
- iv. Select the number of task to use. Make sure the task count is not greater than the runtime setting
- v. Select target platform: EMP-9000-MPAC
- vi. Click 'OK' to confirm the setting.

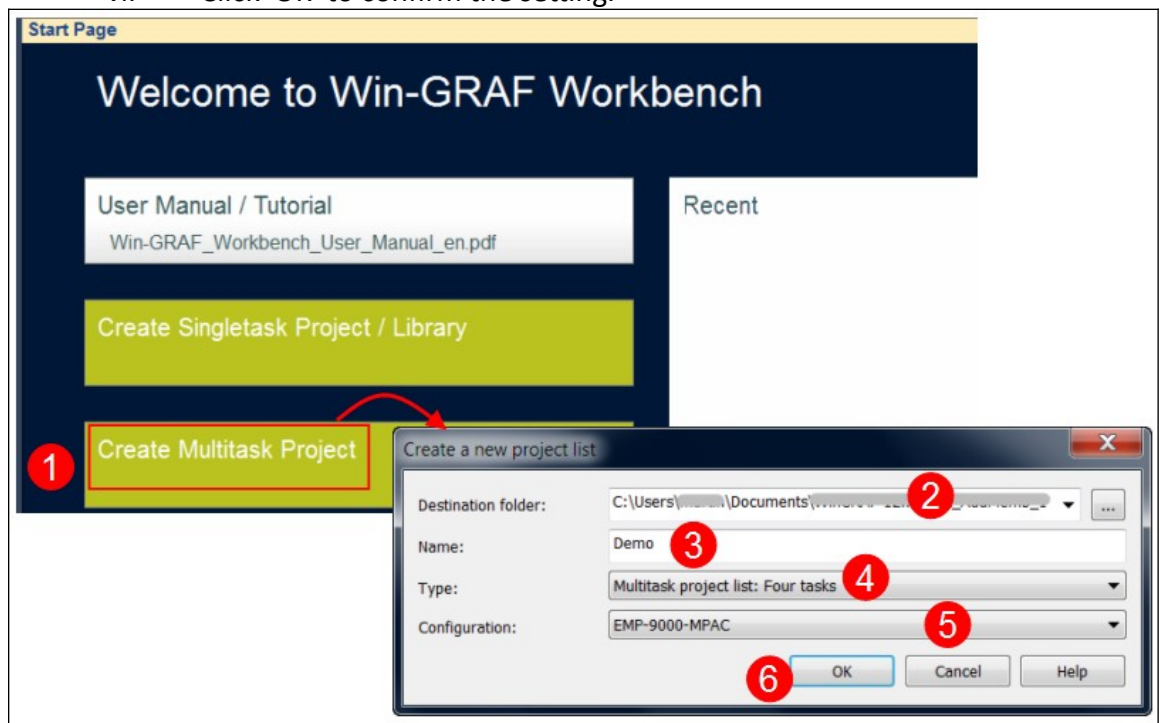


Figure 9: Create a new multi-task Workbench project

Step 5: Determine how the PLC logic application should launch. Several options are available:

1. **No Start:** The logic control application does not start with the *Utility*.
2. **Cold Start:** The application starts with the utility. All application variables are initialized at startup according to the '*Init Value*' setting.

Name	Type	Di...	Attrib.	Syb.	Init value
dwVar	UDINT			<input type="checkbox"/>	UDINT#0
Global variables					
RETAIN variables					
reVar	UDINT			<input type="checkbox"/>	UDINT#0

3. **Warm Start:** All application variables except the '*RETAIN variables*' are initialized at startup according to the '*Init Value*' setting. The '*RETAIN variables*' use the last values before the application stopped.
4. **Hot restart:** No variables are initialized at startup. All variables use the last values before the application stopped. These mode is mainly used for debugging process and it is not supported by all function blocks (e.g. PLCOpen FB).

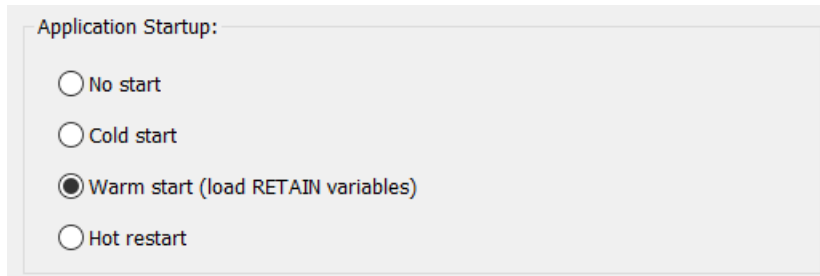


Figure 10: Launch the PLC application along with the utility option

- Step 6:** Save the setting: Click 'Set' button and select '*File\Save*' from the menu. The next time the utility is started, the PLC application will automatically start running.

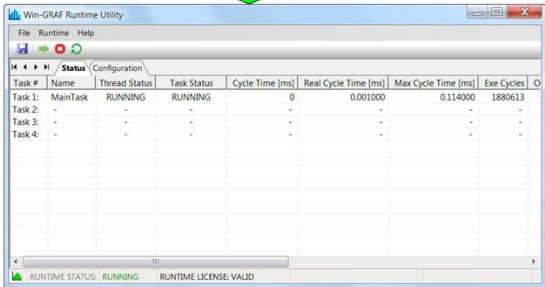
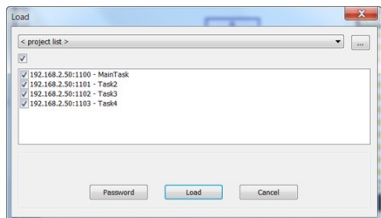
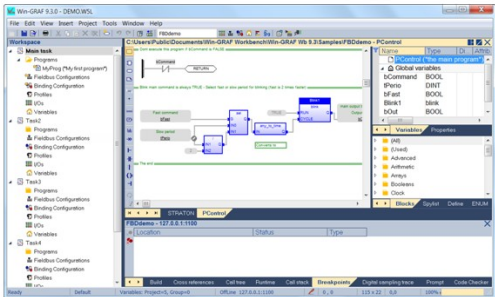
4 Download Application

After the PLC application is successfully compiled, the application has to be downloaded to the runtime for execution. In Win-GRAF, the data exchange between the workbench and the runtime takes place via TCP/IP communication.

1. The PLC application is written and compiled in the Win-GRAF workbench

2. Download compiled application

3. The Win-GRAF runtime executes the application



Task #	Name	Thread Status	Task Status	Cycle Time [ms]	Real Cycle Time [ms]	Max Cycle Time [ms]	Exe Cycles
Task 1:	MainTask	RUNNING	RUNNING	0	0.001000	0.114000	1880613
Task 2:	-	-	-	-	-	-	-
Task 3:	-	-	-	-	-	-	-
Task 4:	-	-	-	-	-	-	-

Figure 11: Edit and download PLC program

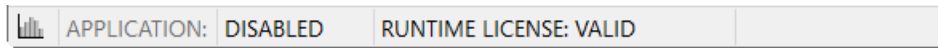
Procedure for downloading the compiled application:

Step 1: Start the Win-GRAF runtime utility if it is not running:

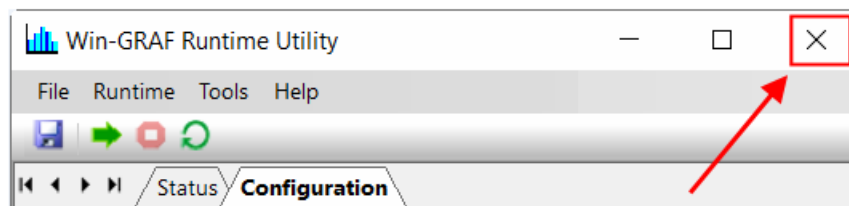
1. Double click the Win-GRAF runtime icon on the desktop to start the Win-GRAF Runtime utility.



The status bar of the utility indicates whether the runtime has been successfully started. In addition the runtime license status will be shown. If the license is invalid the runtime automatically stops running after 20 minutes.



Attention:



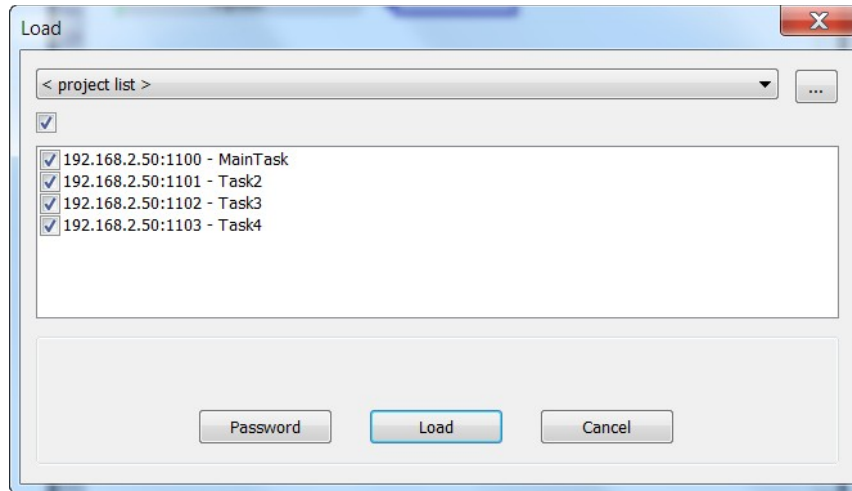
Clicking the window close button does not exit the utility, but only closes the user interface (UI) and adds the corresponding icon to the toolbar in the right section of the taskbar. Even though the utility's interface is not visible, all of its functions and associated runtime are still active and running normally. To display the interface again, simply click the corresponding icon in the toolbar.



To exit the utility including the PLC application, click the '*File\Exit Runtime and Utility*' menu.

Step 2: Download the built plc application to the runtime:

- Click the download button in the toolbar or select '*Project\Download All Projects...*'. The figure below shows an example of an application program with four tasks.
- Click '*Load*' to start the download process



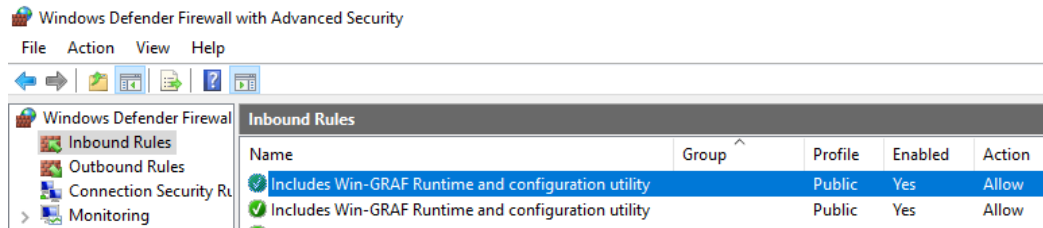
Wait until the download has finished.

ATTENTION:

The runtime stops executing the currently running logic application when a new application file is downloaded. Enabling "Online change" mode ensures that the application runs continuously even when downloading minor application changes.

Note:

- By default all tasks files are selected for download. For large PLC programs: To shorten the download time, disable the task files that have not been changed since the last download.
- After the download process is complete, the application will not restart automatically. This has to be done by the user. The user can decide between a cold or warm restart. With a cold start, the PLC program starts again with the initial values; with a warm start, the program uses retentive data. Not all function blocks support warm start (e.g. PLCOpen function blocks)
- Possible causes if the workbench fails to download the application program:
 - Incorrect workbench IP address setting. The IP address entered in the workbench does not match with the address of the EMP9K target platform.
 - Incorrect network and subnet setting.
 - Firewall of the EMP9K blocks the port used by the workbench. Make sure the port has been enabled for public use.



Step 3: Create a online connection between workbench and the runtime

- Click the 'On Line' button on the toolbar or



- Enter "Ctrl+F5".

Step 4: Start the PLC application:

Normally the 'Start mode' window automatically pops up right after a online connection. Select your required start mode to start the logic control application.

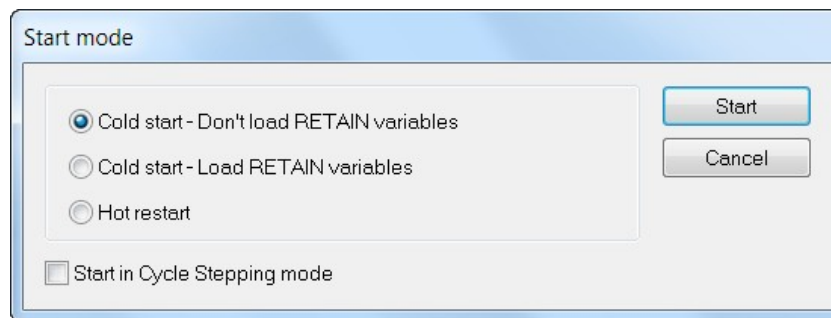
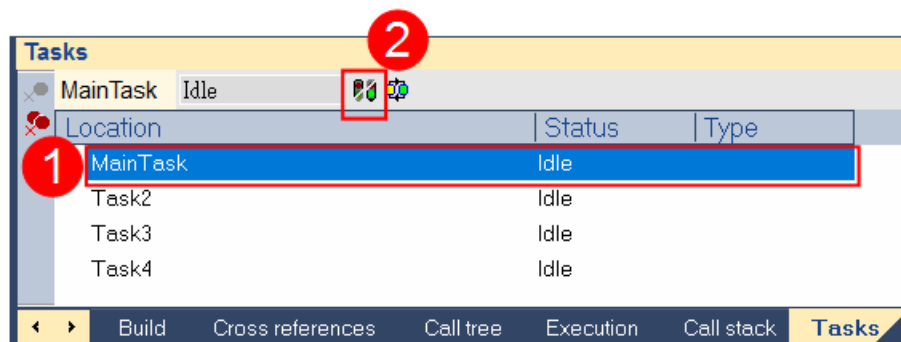


Figure 12: Start mode dialog

For multi-tasking project:

If the 'Start mode' window does not pop up during the previous step, then you can open this window as follows:

- Go to the 'Tasks' tab output window at the bottom. All tasks are shown in idle mode.



- First select the 'MainTask' and click the traffic light button. Now the 'Start

mode' window pops up, which allows you to select the start mode of the PLC application.

The main task starts executing the other tasks when the SYSSTARTTASK() function is called from the main task. Each task can be started manually by selecting it in the 'Tasks' tab output window and activating the traffic light button.

After clicking 'Start' of the 'Start mode' window (Figure 12) the 'Main Task' and the tasks being called by SYSSTARTTASK() will enter the run status.

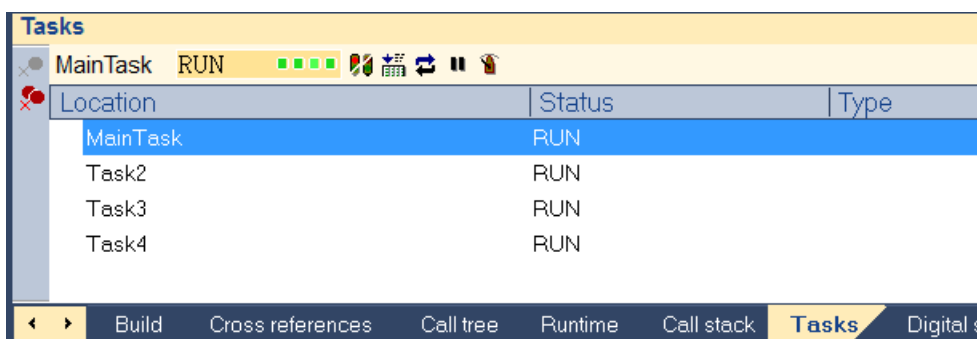


Figure 13: Tasks status tab

The Win-GRAF runtime utility (Figure 14) shows the status of the runtime and its tasks. Further information of each task are shown e.g. the cycle time, number of execution cycle, errors, priority setting, etc..

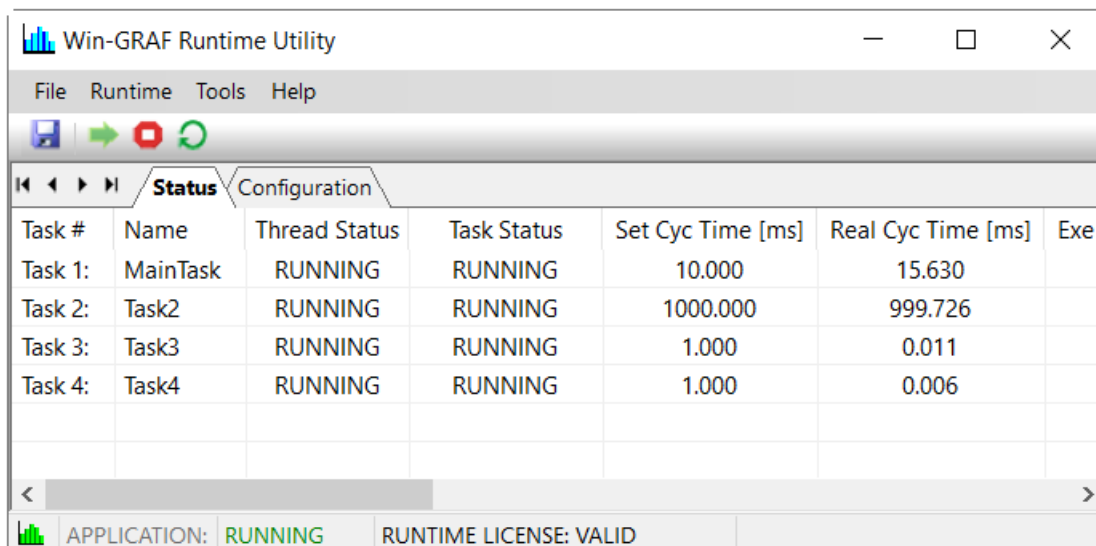


Figure 14: Win-GRAF Runtime Utility

5 Multitasking

5.1 Task Setting

The cycle of each task has to be set directly from the workbench by right-clicking on the task name and selecting 'Task...' from the pop-up menu. Double click the period column next to the task to enter the cycle time. If only one task is being used then the '*Run as fast as possible*' option can be used (Figure 15).

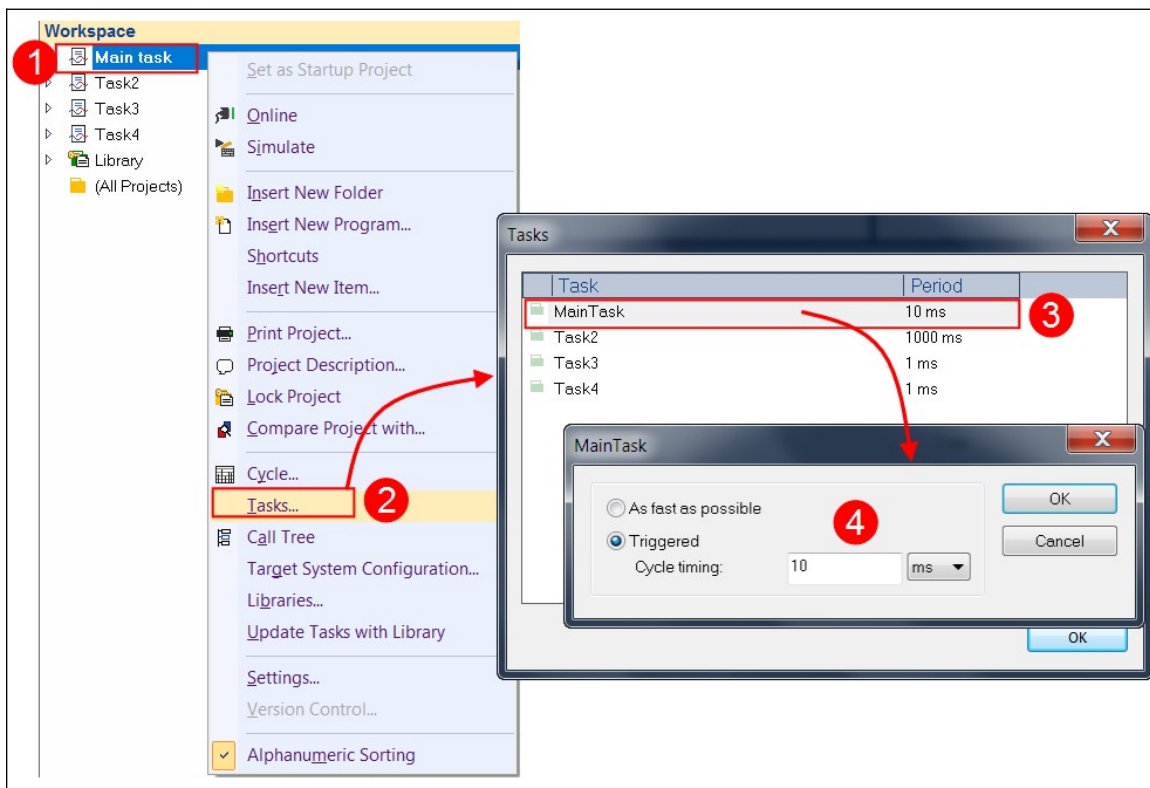


Figure 15: Task cycle time configuration

The '*Main task*' is the first task that is started when the PLC application begins execution. The other tasks have to be started from the '*Main task*' by calling the `SYSSTARTTASK()` function. Additional functions are available for configuring and terminating each task within the PLC program (Figure 16).

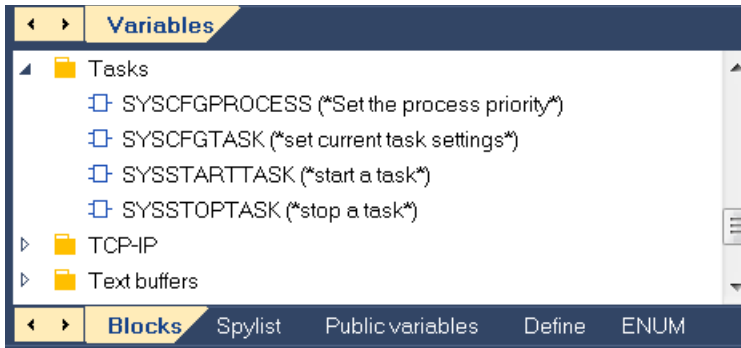


Figure 16: Task dedicated functions

The task library provides the function for configuring and controlling a task:

- To start another task:

```
SYSSTARTTASK( Task(*DINT*), Warm(*BOOL*) )
```

This function should only be called once for starting a task.

- To stop a running task:

```
SYSSTOPTASK( Task(*DINT*) )
```

- Configure a running task:

```
SYSCTGTASK( Prio(*DINT*), Opts(*STRING*) )
```

This function should only be called after a task has been started. It sets the task priority inside a running task.

SYSSTOPTASK(Prio)		
Thread Priority Level	Prio	Description
THREAD_PRIORITY_IDLE	0	
THREAD_PRIORITY_LOWEST	1	
THREAD_PRIORITY_BELOW_NORMAL	2	
THREAD_PRIORITY_NORMAL	3	
THREAD_PRIORITY_ABOVE_NORMAL	4	
THREAD_PRIORITY_HIGHEST	5	
THREAD_PRIORITY_TIME_CRITICAL	6	Default task priority setting of the Win-GRAF runtime

The `SYSCTGTASK` internally calls the `SetThreadPriority()` Windows API to adjust its priority relative to other threads in the process. To keep the priority setting simple it is suggested to just select between `THREAD_PRIORITY_NORMAL` for the normal task and `THREAD_PRIORITY_TIME_CRITICAL` for the high priority task.

Opts:

This parameter is not supported by the EMP9K Win-GRAF runtime. Therefore just enter an empty string (' ').

- Set the priority classes of the process. This is a Windows-related runtime process priority setting.

`SYSCFGPROCESS(Prio(*DINT*))`

This setting is only in rare cases necessary. By default the process priority is set to 'HIGH_PRIORITY_CLASS'.

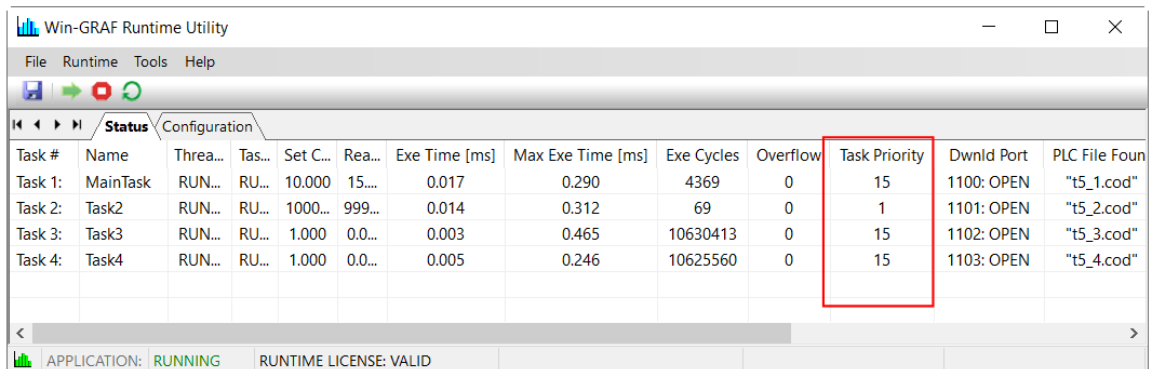
Windows supports the following priority classes:

SYSCFGPROCESS(Prio)		
Priority Classes	Prio	Description
IDLE_PRIORITY_CLASS	0	
BELOW_NORMAL_PRIORITY_CLASS	1	
NORMAL_PRIORITY_CLASS	2	
ABOVE_NORMAL_PRIORITY_CLASS	3	
HIGH_PRIORITY_CLASS	4	Default setting of the Win-GRAF runtime
REALTIME_PRIORITY_CLASS	5	

The white field in Table 1 shows the task priority value for the combination of process class and thread priority setting. The priority setting for each running task is shown in the 'Win-GRAF Runtime Utility' (Figure 17).

SYSSTOPTASK(Prio)		SYSCFGPROCESS(Prio)					
Thread Priority Level (THREAD_PRIORITY_)	Prio	IDLE_PRIORITY_CLASS	BELOW_NORMAL_PRIORITY_CLASS	NORMAL_PRIORITY_CLASS	ABOVE_NORMAL_PRIORITY_CLASS	HIGH_PRIORITY_CLASS	REALTIME_PRIORITY_CLASS
		0	1	2	3	4	5
IDLE	0	1	1	1	1	1	16
LOWEST	1	2	4	6	8	11	22
BELOW_NORMAL	2	3	5	7	9	12	23
NORMAL	3	4	6	8	10	13	24
ABOVE_NORMAL	4	5	7	9	11	14	25
HIGHEST	5	6	8	10	12	15	26
TIME_CRITICAL	6	15	15	15	15	15	31

Table 1: Task priority



Win-GRAF Runtime Utility

File Runtime Tools Help

◀ ▶ 🔍 🔄

◀ ▶ ▶ ▶ / **Status** Configuration

Task #	Name	Threa...	Tas...	Set C...	Rea...	Exe Time [ms]	Max Exe Time [ms]	Exe Cycles	Overflow	Task Priority	Dwnld Port	PLC File Foun
Task 1:	MainTask	RUN...	RU...	10.000	15...	0.017	0.290	4369	0	15	1100: OPEN	"t5_1.cod"
Task 2:	Task2	RUN...	RU...	1000...	999...	0.014	0.312	69	0	1	1101: OPEN	"t5_2.cod"
Task 3:	Task3	RUN...	RU...	1.000	0.0...	0.003	0.465	10630413	0	15	1102: OPEN	"t5_3.cod"
Task 4:	Task4	RUN...	RU...	1.000	0.0...	0.005	0.246	10625560	0	15	1103: OPEN	"t5_4.cod"

< >

APPLICATION: **RUNNING** RUNTIME LICENSE: VALID

Figure 17: Runtime task priority setting

6 EWF Function

The EWF is a feature of the EMP9K to protect the Windows IoT operating system.

When the EWF feature is enabled, the data written to the C: partition will be lost after a system reboot. This protects the operating system and allows the user to directly turn off the power to the EMP9K device without shutting down the operating system. Only the C: drive is protected by EWF and the other partitions and drives are not affected by the EWF setting and do not lose their data after a reboot.

By default, the Win-GRAF runtime is installed in the 'T:\Win-GRAF' directory, which means that no PLC-related settings and retentive data are lost when the power to the EMP9K is turned off.

To permanently save data to the C: partition, update the operating system, or install new software, you must first disable EWF protection using the '**Pac Utility**' (Figure 18) and restart the operating system for the setting to take effect .

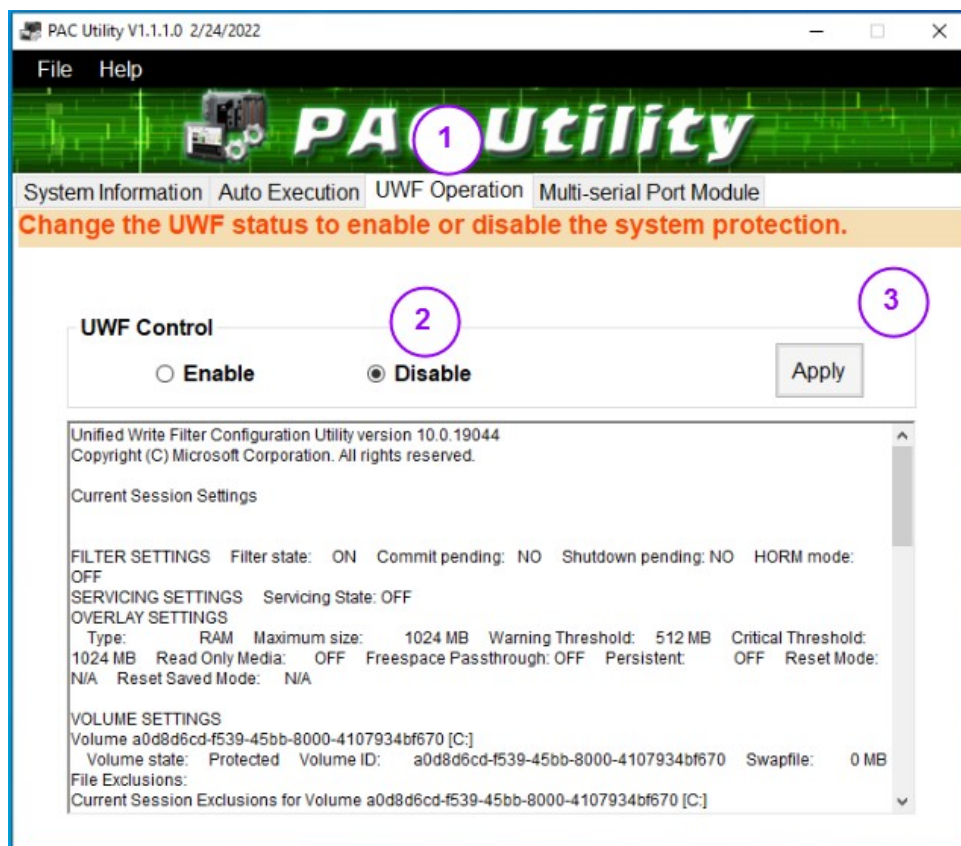


Figure 18: Disable UWF control

Important

Once UWF control has been disabled, the EMP9K IoT operating system must be shut down like a standard Windows operating system before turning off. Turning off the power directly will damage the operating system.

It is highly recommended to enable UWF control after making C: partition changes to protect the operating system. Immediately after enabling UWF control, a reboot is required for it to take effect.

7 Activate “Lock Screen”

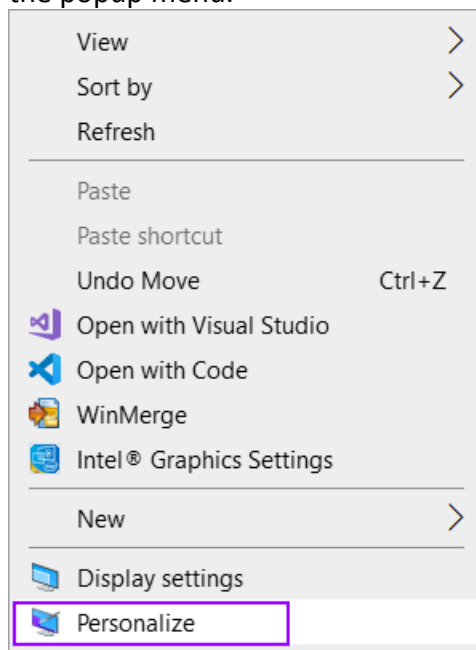
To prevent unauthorized persons from accessing and manipulating the Win-GRAF runtime while the PLC application is running, it is strongly recommended that you protect your computer:

1. Set the password for the operating system.
2. Activate the screen lock automatically.

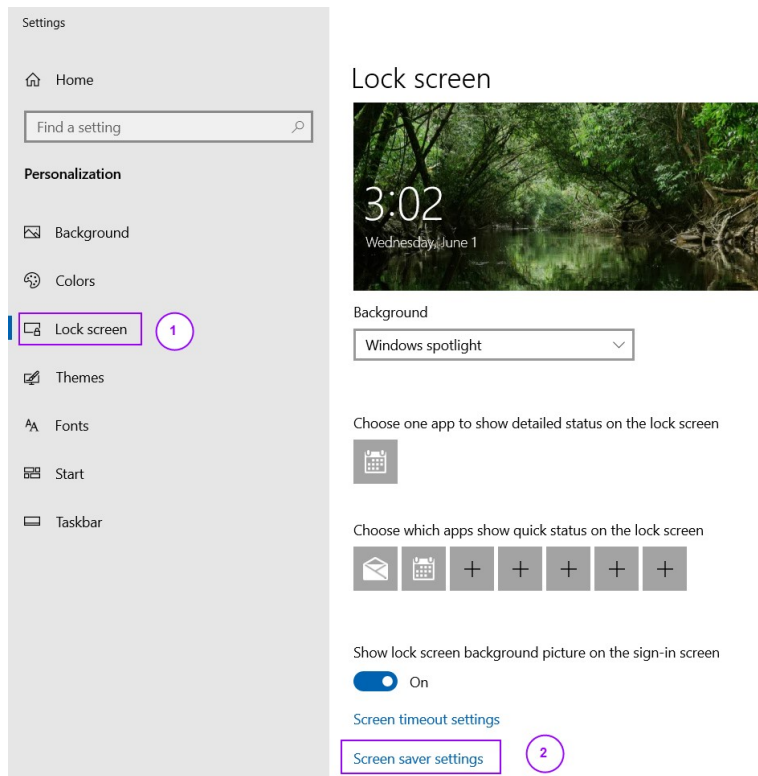
Enabling automatic screen lock prevents people from tampering with runtime settings. The screen can only be unlocked by entering the operating system password (e.g. User Account password)

Steps for activating the screen lock functions:

Step 1: Right-click on an empty place on your desktop and select '*Personalize*' from the popup menu.

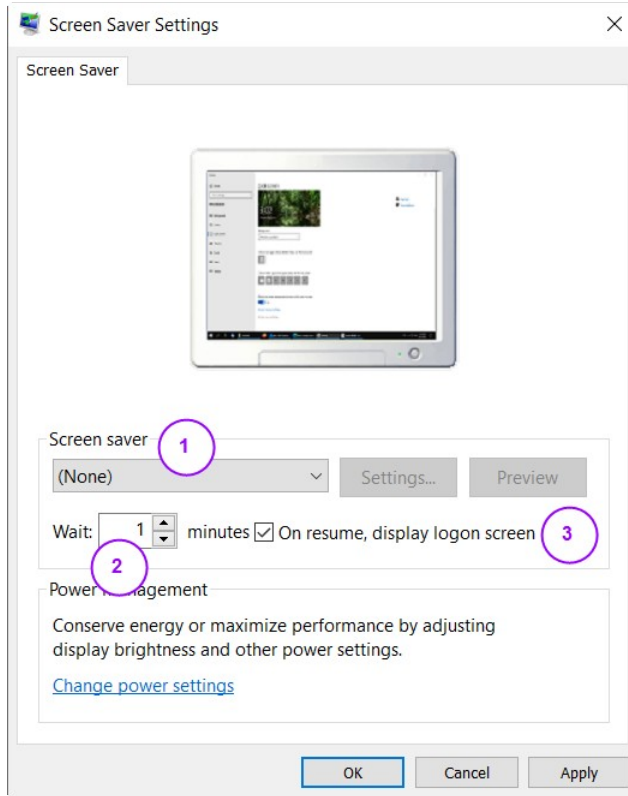


Step 2: From the side menu select '*Lock Screen*' and click '*Screen Saver Settings*'




Step 3: Set Screen saver:

- i. Choose the type of screen saver.
- ii. '*Duration*': The '*Wait*' option sets the number of minutes of inactivity before the screen saver turns on. The lowest wait time is one minute
- iii. '*On resume, display logon screen*': If this option is enabled, your device will automatically lock when the screen saver activates. It is necessary to enter the user password to unlock the screen saver.
- iv. Confirm the setting with '*OK*'



The Win-GRAF runtime continues to run regardless of whether the screen saver is displayed or not. Even a password-protected screen saver has no influence on runtime execution.

The screen saver is activated and appears after the set time has elapsed. If you want to display the screen saver immediately when you leave the device, press the key combination:

'Windows Logo'  key + 'L' key

8 Platform Specific Library

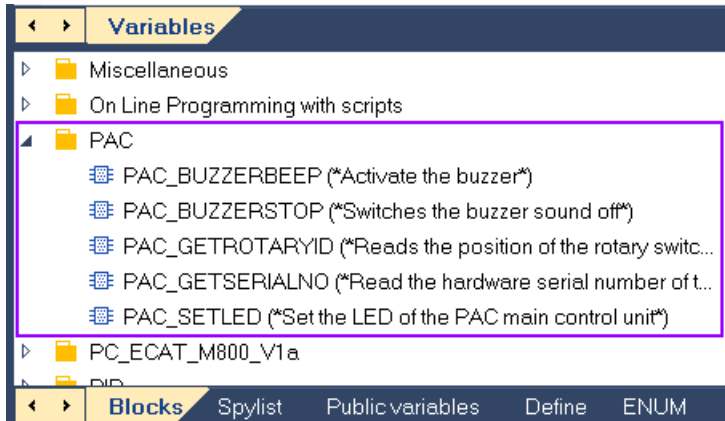


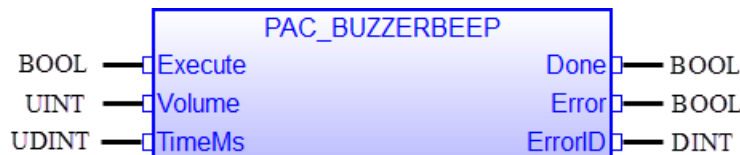
Figure 19: Main control unit library

8.1 Buzzer

Buzzer is mainly used as warning tone and alarm sound. The EMP9K has got an integrated buzzer which can be activated and disabled.

8.1.1 PAC_BuzzerBeep

This function triggers the buzzer to generates a beep tones.



Input/Output	Name	Type	Comments
VAR_INPUT	Execute	BOOL	Start the buzzer beep tone
	Volume	UINT	Set the beep tone frequency
	TimeMs	UDINT	The sleep time between beep tones
VAR_OUTPUT	Done	BOOL	True when the buzzer has been enable
	Error	BOOL	An error has occurred within the FB
	ErrorID	DINT	Error identification

Remarks:

-

8.1.2 PAC_BuzzerStop

This function stops the buzzer to generates a beep tones.



Input/Output	Name	Type	Comments
VAR_INPUT	Execute	BOOL	Stop the buzzer beep tone
VAR_OUTPUT	Done	BOOL	True after the FB successfully disabled the buzzer
	Error	BOOL	An error has occurred within the FB
	ErrorID	DINT	Error identification

Remarks:

-

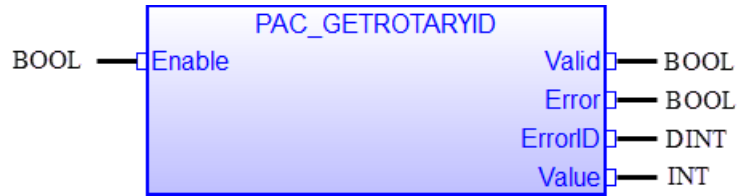
8.2 Rotary Switch

The main control unit of the EMP9K is equipped with a rotary switch. The current switch position can be determined in the PLC application.



8.2.1 PAC_GetRotaryId

This function retrieves the position number of the rotary switch.



Input/Output	Name	Type	Comments
VAR_INPUT	Enable	BOOL	TRUE: Continuously reads the rotary switch position in each cycle
VAR_OUTPUT	Valid	BOOL	True indicates that the FB output rotary switch value is valid
	Error	BOOL	An error has occurred within the FB
	ErrorID	DINT	Error identification
	Value	INT	Rotary switch position

Remarks:

-

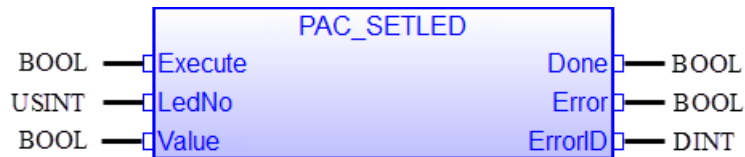
8.3 LED Control

Integrated in the main housing are three LEDs which can be set on/off programmatically.



8.3.1 PAC_SetLed

This function switches the LED on/off.



Input/Output	Name	Type	Comments
VAR_INPUT	Execute	BOOL	Trigger the FB execution
	LedNo	USINT	The LED number <ul style="list-style-type: none"> 0 - RUN 1 - L1 2 - L2
	Value	BOOL	TRUE - ON FALSE - OFF
VAR_OUTPUT	Done	BOOL	True when the LED has been set
	Error	BOOL	An error has occurred within the FB
	ErrorID	DINT	Error identification

Remarks:

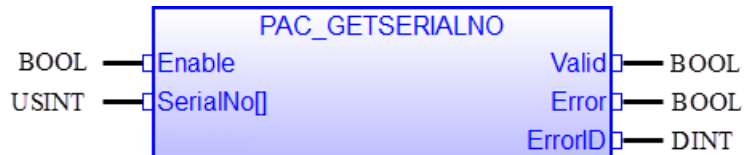
-

8.4 Serial Number

Each EMP9K device has a unique hardware number. The serial number can be used to link a PLC application directly to an EMP9K device to prevent it from running on another EMP9K device.

8.4.1 PAC_GetSerialNo

This function retrieves the serial number of the EMP9K.



Input/Output	Name	Type	Comments
VAR_INPUT	Enable	BOOL	TRUE: Continuously reads the serial number in each cycle
VAR_OUTPUT	Valid	BOOL	True if the "SerialNo[]" output is valid
	Error	BOOL	An error has occurred within the FB
	ErrorID	DINT	Error identification
	SerialNo[]	USINT	<ul style="list-style-type: none"> Serial number

			<ul style="list-style-type: none"> The serial number is eight byte long. Therefore declare the following array SerialNo : ARRAY [0 .. 7] OF USINT <p>ATTENTION: Although the parameter is shown on the input side of the function block it acts as an output parameter.</p>
--	--	--	--

Remarks:

-