Driver for WAGO PLC (PFC100 and PFC200) (WagoDrv module of the REX Control System)

User guide

REX Controls s.r.o.

Version 2.50.5
Plzeň (Pilsen), Czech Republic
2017-09-06

Contents

Chapter 1

Introduction

The WAGO PLC platform represented by the PFC100 a PFC200 models is supported in REX starting with version 2.50.5. The support includes integration of Kbus for communication with WAGO I/O modules 750/753 and other standard drivers (Modbus RTU, Modbus TCP/IP, Siemens PLC communication driver, database driver), function block libraries and web user interface. The support includes basic analog and digital I/O modules, persistent memory and indication LED.

Chapter 2

Installation

The runtime modules of the REX system may be installed by replacing the operating system with an OS image from SD card or by downloading and installing a few packages directly into a running device.

Please follow the instructions in the Getting started guide for the Wago PFC platform.

2.1 Location of files

All REX runtime configuration and data files are located in the `/rex` directory. However, this directory name is in fact a symbolic link to the `/home/rex` directory on the WAGO platform due to the system drive layout. Creation of a user account named "rex" is not recommended due to possible conflicting of names.

Chapter 3

Identification and licensing

The WAGO PFC100 and PFC200 platforms are identified as WAGO PFC, the generic identification is "Generic Linux on ARM". A DEMO license is available for testing and evaluation. A proper license for required functionality has to be obtained for permanent operation. The process of obtaining a license for a device is described at `https://www.rexcontrols.com/rex-control-system-licensing`.

Chapter 4

Integration into the project

A driver named *WagoDrv* is available for the WAGO PFC platform. The driver makes accessible I/O modules and some of the LED indicators to the control algorithm. Integration of the driver into a project is done using the `MODULE` and IODRV/TIODRV function blocks. Both *module name* and *class name* parameters are set to `WagoDrv`. For details about individual function blocks of the REX system please refer to [1].

4.1 I/O variables

A simple `FROM/GOTO` or extended `INQUAD, INOCT, OUTQUAD, OUTOCT` function blocks may be used to access runtime variables provided by the driver. If the number of block's inputs/outputs exceeds the number of module channels addressed, excess outputs are set to 0 with BAD quality, excess inputs are ignored. Examples of accessing variables from a control algorithm are provided within installation of development tools. All examples for the WAGO platform are identified as 0102-xx.

The variable names for accessing the data of I/O modules are assembled in following way:

S<X>M<Y> `[input/output, double/word/bool]` – Refers to the data of a slot number X in which a module of type Y is present. The slots are numbered from 1 starting with the module next to the CPU module. Supply and interconnection modules are not numbered. The signal type is defined by the module referred – boolean for digital inputs/outputs and double for analog inputs/outputs. Example: for I/O module 750-430 or 753-430 the variable has name `WG__S1M430`. In case of a special module, the type is word and the variable contains the a raw data from the bus. A user has to process the value properly according to the documentation. A list of modules supported by the driver is attached.

S<X>M<Y>C<Z> `[input/output, double/word/bool]` – Refers to the data of a channel number Z, slot number X in which a module of type Y is present. The channels are numbered according the documentation of a module. Example: channels C1-C8 may be used when referring to the 750-430 I/O module.

S<X>M<Y>W<offset> `[vstup/výstup, word]` – Refers to the data of a slot number X in which a module of type Y is present. The offset is a word offset from the beginning of module's process image (see documentation for respective I/O module).

S<X>M<Y>OK `[vstup, bool]` – Indicates, whether a module Y in slot X is present, configured and fully functional.

A "_" (underscore) character may be added to the variable name and followed by arbitrary text. This part of variable name is intended only for user identification purpose and is ignored during the compilation process.

4.2 Controlling LEDs

Indication LEDs on the front side of PLC may be controlled from a control algorithm using variables with a name in a form:

Led<name>

where *<name>* identifies a LED. Following LEDs may be controlled : **MS, NS, USR, U1-U7**. The *U1-U7* LEDs are available only on PFC200 platform. Use the `GOTO` function block with the *GotoTag* set properly. Example: `WG__LedMS`.

A LED variable has integer type. Its value corresponds to a LED state. Following LED states are defined:

State number	LED state
0	Off
1	Red, still on
2	Green, still on
3	Yellow, still on
4	Red, flashes fast
5	Green, flashes fast
6	Yellow, flashes fast
7	Red, flashes slowly
8	Green, flashes slowly
9	Yellow, flashes slowly

Notice: The state numbers and LED states may be changed by modifying the `/etc/rexcore/cstdiagnostic.xml` file directly on a target WAGO PFC device.

4.3 Meaning of LED indicators

SYS – system state: green – OK, red – ERROR,

RUN – REX runtime state: green – RUNNING, red – ERROR, yellow – STOP,

I/O – Kbus state: green – RUNNING, red – ERROR/terminal module not connected,

MS, NS, USER, U1-U7 – meaning defined by a user in a control algorithm.

4.4 Global variables

Following global variables are available:

KbusOK `[input, boolean]` – indicates a correct operation of a system bus (Kbus),

KbusStatus `[input, int]` – number of a last communication cycle error, 0 = no error; the error number is not compatible with standard REX error codes – use only for diagnostic purpose,

ModulesOK `[input, boolean]` – indicates whether all I/O modules used in the algorithm work correctly,

CycleCount `[input, int64]` – holds the total communication cycle count of the system bus from the start of operation,

KbusAllowControlMode `[output, boolean]` – switches the system bus into the configurable state. See the *WAGO IO Check* chapter for more information.

4.5 Mode selector switch

The "Mode selector switch" on a CPU unit starts/stop the control algorithm. Additionally, a control algorithm may be restarted (re-loaded from persistent storage) by pushing the switch to the "reset" position for more than 2 seconds. If the switch is hold in the "reset" for more than 8 seconds, the control algorithm is reloaded and all persistent (retentive) variables are cleared.

Notice: A control algorithm can not be started if a switch is in the "stop" position and a download operation finishes with warning.

4.6 Communication cycle

Internal communication on a system bus (Kbus) between CPU and connected I/O modules is cyclic and synchronous. The bus requires approximately 3 ms to transfer all data in a single cycle. The value may rise if more I/O modules are connected. We don't recommend to set the communication cycle of a driver to less than 5 ms. The communication cycle is derived from the main executive tick multiplied by the driver factor (tick x driver factor). The cycle length also must not be longer than 50 ms. If the cycle length is longer, the bus watchdog is activated and all I/O modules are reset (outputs are set to 0). An actual cycle length may be obtained from the diagnostic window in RexDraw.

Chapter 5

WAGO IO Check

WAGO IO Check is a software tool by the WAGO company intended for configuration of I/O modules connected to the PLC. It is possible to connect with WAGO IO Check when a REX runtime is running. A user may then perform diagnostic and/or configuration operations. (some I/O modules have configuration options, some don't offer any). Please, see the WAGO IO CHECK documentation for more information.

Notice: It is not possible to perform any configuration operations in the default operation mode of the REX runtime and only monitoring may be performed. When configuration task are necessary, the `GOTO` function block that accesses the **KbusAllowControlMode** variable has to be inserted into algorithm and set to "true". The communication cycle is not terminated when the value of the **KbusAllowControlMode** variable is changed. If set to "true", configuration operations may be performed.If set back to "false", all modules are re-initialized and the configuration is checked.

Notice: WAGO IO Check can not connect if the REX runtime or any other runtime is not running on the device.

Chapter 6

Web interface

A standard WEB interface of the REX runtime is available on port 8008 (http) and 8009 (https). A WAGO WEB interface is available on port 80 (http) and 443 (https).

Chapter 7

Security

We strongly advise to secure the device properly before commissioning including setting of user accounts and passwords, services and encryption.

7.1 User accounts

By default, following user accounts are available on the WAGO PFC platform:

1. In Linux: root/wago and admin/wago.
2. In WAGO WEB interface: admin/wago and user/wago.
3. In REX runtime: admin/*blank password*.

The accounts are not connected and must be set individually. Please, change the default password before commissioning by following these steps:

1. In Linux: connect via SSH (PuTTy) using the respective user name and change the password using the `passwd` command.
2. In WAGO WEB interface: in section Administration.
3. In REX runtime: connect using RexDraw and select Target->Change password.

See more information about user accounts and permissions documentation of RexCore [2].

7.2 Services

A WAGO IO Check connection is enabled by default. The connection does not require authentication. The service may be disabled by following command:

```
/etc/config-tools/config_iocheckport state="disabled"
```

7.3 HTTPS

Certificate and private key files are created automatically during the startup if those file are not already present. The certificate file is located in `/etc/rexcore/rexcore.cer`, the private key file is located in `/etc/rexcore/rexcore.key`. Secured WEB interface of REX runtime runs on port 8009. A connection may be established by URL in the form: *https://device-ip-address:8009*.

Chapter 8

Advanced configuration of a web interface

The WEB interface of the REX runtime may be disable – see the documentation of RexCore. Alternatively, both REX runtime and WAGO WEB interfaces may be united using a proxy WEB server. In that case, the WAGO WEB interface service ports have to be changed, for example to ports 8080 (http) and 8433 (https) — see `/etc/lighttpd/lighttpd.conf`. Then, a proxy server has to be installed. For this purpose, a nginx package is prepared. The configuration file is located in `/etc/nginx/nginx-rex.conf`.

First, enable the HTTP service in the WAGO WEB interface. Then, install the nginx server by following commands:

```
wget http://download.rexcontrols.cz/files/wago/<version>/nginx_1.10.3_arm.ipk
```

```
ipkg install nginx_1.10.3_arm.ipk
```

HTTPS protocol may be enabled in the configuration file `/etc/nginx/nginx-rex.conf` by uncommenting the lines that follow the "uncomment to enable SSL" line. The nginx or the whole device have to be restarted after the configuration is changed. The nginx proxy server uses the same certificate and private key as the REX runtime by default.

Chapter 9

Other tips

Complete identification of a device:

```
/etc/config-tools/get_typelabel_value -a
```

or at *http://device-ip-address/api/:platinfo*

Miscellaneous information about a device:

```
/etc/config-tools/get_coupler_details -help
```

Please, send all questions related to the WAGO PFC platform or the REX system to our support team at support@rexcontrols.com.

Appendix A

Supported I/O modules

- **Full** – Individual signals are provided in units of measurement (volts, milliamps, etc.).
- **Partial** – Individual signals are provided as data for further processing (using function blocks LIN, GAIN, CNDR, INTSM, etc.). For details about individual function blocks of the REX system please refer to [1].
- **Raw** – Raw process image data of the module is available. It is up to the user to parse the data.
- **N/A** – Module is not supported at all. This status also applies to modules which are not listed here. Contact us for further details.

Module	Type	Support status
400	Digital input	Full
401	Digital input	Full
402	Digital input	Full
403	Digital input	Full
405	Digital input	Full
406	Digital input	Full
408	Digital input	Full
409	Digital input	Full
410	Digital input	Full
411	Digital input	Full
412	Digital input	Full
414	Digital input	Full
415	Digital input	Full
418	Digital input	Full
419	Digital input	Full

The list continues on the next page...

Module	Type	Support status
421	Digital input	Full
422	Digital input	Full
423	Digital input	Full
424	Digital input	Full
425	Digital input	Full
427	Digital input	Full
428	Digital input	Full
430	Digital input	Full
431	Digital input	Full
432	Digital input	Full
433	Digital input	Full
434	Digital input	Full
435	Digital input	Full
436	Digital input	Full
437	Digital input	Full
438	Digital input	Full
440	Digital input	Full
450	Analog input	Partial
451	Analog input	Partial
452	Analog input	Full
453	Analog input	Full
454	Analog input	Full
455	Analog input	Full
456	Analog input	Full
457	Analog input	Full
459	Analog input	Full
460	Analog input	Partial
461	Analog input	Partial
462	Analog input	Partial
465	Analog input	Full
466	Analog input	Full
467	Analog input	Full
468	Analog input	Full
469	Analog input	Full
472	Analog input	Full
474	Analog input	Full
475	Analog input	Full
476	Analog input	Full
477	Analog input	Full
478	Analog input	Full

The list continues on the next page...

Module	Type	Support status
479	Analog input	Full
480	Analog input	Full
481	Analog input	Full
483	Analog input	Full
485	Analog input	Full
491	Analog input	Full
492	Analog input	Full
501	Digital output	Full
502	Digital output	Full
504	Digital output	Full
506	Digital output	Full
507	Digital output	Full
508	Digital output	Full
509	Digital output	Full
512	Digital output	Full
513	Digital output	Full
514	Digital output	Full
516	Digital output	Full
517	Digital output	Full
519	Digital output	Full
522	Digital output	Full
523	Digital output	Full
530	Digital output	Full
531	Digital output	Full
532	Digital output	Full
534	Digital output	Full
535	Digital output	Full
536	Digital output	Full
537	Digital output	Full
540	Digital output	Full
550	Analog output	Full
552	Analog output	Full
553	Analog output	Full
554	Analog output	Full
555	Analog output	Full
556	Analog output	Full
557	Analog output	Full
559	Analog output	Full
560	Analog output	Full
562	Analog output	Full

The list continues on the next page...

Module	Type	Support status
563	Analog output	Partial
585	Analog output	Full
638	Special	Partial
1415	Digital input	Full
1416	Digital input	Full
1417	Digital input	Full
1418	Digital input	Full
1420	Digital input	Full
1421	Digital input	Full
1422	Digital input	Full
1423	Digital input	Full
1425	Digital input	Full
1500	Digital output	Full
1501	Digital output	Full
1502	Digital output	Full
1504	Digital output	Full
1505	Digital output	Full
1506	Digital output	Full
1515	Digital output	Full
1516	Digital output	Full
???	Unlisted modules	N/A

Bibliography

[1] REX Controls s.r.o.. *Function blocks of the REX Control System – reference manual*, 2017.

[2] REX Controls s.r.o.. *RexCore – User manual*, 2017.

Documentation reference number: 8614