www.rexcontrols.com/rex

REX Control System Core

User guide

REX Controls s.r.o.

Version 2.50.4
2017-05-17
Plzeň (Pilsen), Czech Republic

Contents

Chapter 1

Introduction

The most important part of the REX Control System is the real-time core contained in a software component called RexCore. The RexCore is an independent process in the case of a *Windows* or a *GNU/Linux* operating systems, a group of real-time tasks in the case of a hard real-time operating system or a whole system in the case of a microcontroller but it is always the software component where a control algorithm developed by a user runs and from which a device is being handled and which initiates a communication with all other devices in the field that are involved in the control, diagnostics or provide a human-machine interface.

This document contains all the information of a configuration of the RexCore software component for Windows or GNU/Linux operating system that is necessary to put the RexCore into operation or commisioning. A developer of a control algorithm should read [1]. A user that is not yet familiar with the REX Control System should read any of getting-started tutorials, for example [2], [3], [4].

It should be noted that although a big effort has been put into making the RexCore on all supported platforms as unified as possible, it is principally not possible to provide absolutely the same functionality and performance on all devices and environments. A user should consider the targeting platform very carefully and eventually consult his needs with the REX Cotrol Systems s.r.o. company.

Chapter 2

Configuration

The RexCore itself requires only a very limited interaction with a user. In fact it needs no interaction at all when a standard installation or distributing channel is used and a default configuration is sufficient for a user. It is however often necessary to adjust the default configuration or secure the control system when commissioning or sometimes even during the development.

2.1 Location of Configuration Files

All files that are required for the proper operation of RexCore are located in a *configuration directory*. The *configuration directory* is `/rex/rexcore` in the case of GNU/Linux or in the `REX Controls` directory in the *All users profile* in the case of Windows, which should be `C:\ProgramData\REX Controls\REX<version>\RexCore` in most cases.

Following files may be present in the *configuration directory*:

- `license.txt` – A file containing licensing keys for the device. It may be placed manually by a user to the device, but using a RexDraw or RexView tools for licensing operations is most common.
- `exec.rex` – A binary file with the control algorithm configuration. A user should not touch the file unless he or she knows exactly what to do.
- `hmi.rex` – A binary file with the Human-Machine Interface configuration. A user should not touch the file unless he or she knows exactly what to do.
- `auth.rex` – A binary file with the configuration of users, groups and permissions. A user should not touch the file unless he or she knows exactly what to do.
- `rexcore.cfg` – A textual file for RexCore with all configuration options that are available and which may be adjusted by a user. A detailed options description is provided in section 2.2.

2.2 Description of Configuration Options

A configuration of RexCore is done by modifying configuration options in the `rexcore.cfg` file. Each configuration option should be on a single separate line and should have a format `option=value` with no additional spaces or tabs. A following table contains descriptions of all configuration options.

Option	`Default value`	**Description**
`server.tcp`	`:43981`	Address and port on which a diagnostic subsystem of RexCore should listen for incoming TCP connections. Set to "disabled" to disable the service. A wildcard address is used when none is specified which means that listening is performed on all network interfaces.
`server.ssl`	`:43997`	Address and port on which a diagnostic subsystem of RexCore should listen for incoming SSL connections. Set to "disabled" to disable the service. A wildcard address is used when none is specified which means that listening is performed on all network interfaces.
`server.http`	`:8008`	Address and port on which an integrated web server should listen for incoming HTTP connections. Set to "disabled" to disable the service. A wildcard address is used when none is specified which means that listening is performed on all network interfaces.
`server.https`	`:8009`	Address and port on which an integrated web server should listen for incoming HTTPS connections. Set to "disabled" to disable the service. A wildcard address is used when none is specified which means that listening is performed on all network interfaces.
`auth.enabled`	`1`	Enable/disable authentication subsystem. User has to be authenticated by a user name and password if the authentication subsystem is enabled. No authentication is required if disabled.
`auth.allowsystem`	`1`	Enables or disables the possibility of logging into the REX control system using unix accounts. This option is available on GNU/Linux systems only. See section 5.2 for more information.

auth.allowroot	1	Enables or disables of logging into the REX control system as an admin using the unix root account. This option is available on GNU/Linux systems only. See section 5.2 for more information.
auth.super.enabled	0	Enables or disables *super* accounts. See section 5.1 for more information.
auth.operator.enabled	0	Enables or disables *operator* accounts. See section 5.1 for more information.
auth.guest.enabled	0	Enables or disables *guest* accounts. See section 5.1 for more information.
log.file.enabled	0	Enables or disables logging into file defined by the option log.file.
log.file	rexcore.log	RexCore log file where all diagnostic messages from the system log are stored if the option log.file.enabled is set
hmi.file	hmi.rex	HMI file
auth.file	auth.rex	Athentication database file
exec.file	exec.rex	Algorithm configuration file
hmi.path	../www/hmi	A path for additional/static HMI files. The path is absolute or relative to RexCore configuration directory.
archive.path	../arc	A path for binary files with archive data. The path is absolute or relative to RexCore configuration directory.
data.path	../data	A path for blocks and drivers data files. The path is absolute or relative to RexCore configuration directory.
web.webroot	../www	A path for document root of static web files that are persistent on target. The path is absolute or relative to RexCore configuration directory.
rexcore.cert	rexcore.cer	RexCore certificate file used for HTTP or diagnostic connections over SSL. The file should be in PEM format.
rexcore.privkey	rexcore.key	RexCore private key file used for HTTP or diagnostic connections over SSL. The file should be in PEM format.

`dgn.messages`	`0xc333370`	Diagnostic messages that are shown in system log. User should not set the option directly but rather use RexDraw or RexView to configure the system log. See chapter 3 for more information.

Chapter 3

System Log

The RexCore has an integrated system log to which all diagnostic messages are stored. Diagnostic messages are errors, warnings or informational messages that may have impact on execution of a control algorithm.

Each message that is written into the system log has an information about a date and time when it was written and flags that indicate a source subsystem and a severity of the message.

3.1 Sources and Severities

Following sources of diagnostic messages exists in the REX Control System:

- `OS` - operating system abstraction layer,
- `CORE` - real-time core,
- `DIAG` - diagnostic subsystem,
- `BLOCK` - function block algorithms,
- `IODRV` - input/output a communication drivers.

Following severities are distinguished:

- `Error`,
- `Warning`,
- `Information`,
- `Verbose information`.

See [5] for more information on how to configure the system log.

Chapter 4

Web Interface

Starting with version 2.50 of the REX Control System, a highly optimized web server is integrated directly into RexCore. The server listens on port 8008 for HTTP connection and on port 8009 for HTTPS connections by default. A certificate and a private key must be installed to enable HTTPS connections. See chapter 6 for more information.

Although a big effort has been put into stabilization and testing of the integrated web server, an unexpected error may possibly have an impact on the execution of a control algorithm. If that is not acceptable for your critical application, please, consider disabling the integrated web server by setting the options `server.http` and `server.http` to "disabled" and wait for the next version of the REX Control System where the integrated web server will be put into another completely separated process.

4.1 Available Services

Following services are handled by the integrated web server:

- sending of static file content from the executive configuration files,
- sending of static file content from the directory specified by the `web.webroot` configuration option,
- providing dynamic content over the REST API,
- providing dynamic content over a WebSocket connection,
- handling diagnostic connections over WebSocket.

See [6] and [7] for more information.

Chapter 5

Authentication

An authentication subsystem has been integrated into the version 2.50 of the REX Control System. Users, roles and permissions of the roles are defined by the authentication subsystem. The authentication subsystem may also be attached to standard accounts from passwd/shadow files in the GNU/Linux environment. An engine for PAM and LDAP authentication mechanisms is planned but not yet supported.

User accounts and roles are fixed in the version 2.50. A configuration interface will be added into RexDraw and RexView in a next service release.

5.1 Users and Roles

Following roles and permissions are defined:

- *admin* - full permission set, a user with this role may perform any operation and has full control over the target device.
- *super* - a user with this role can not perform any operation that has an impact on the target device ie. reboot the device, set system clock etc., but still may modify the running executive,
- *operator* - a user with this role can not change the running executive but may read and change signal values,
- *guest* - a user with this role can only read signal values and can not make any modifications.

Only the *admin* role is enabled and the roles super, operator and guest are disabled by default. A user may enable those roles by setting options `auth.super.enabled`, `auth.operator.enabled` and `auth.guest.enabled`. For each role a user account with the respective name is present in the version 2.50.

The authentication subsystem is enabled by default. A user may disable the authentication by the option `auth.enabled`. The admin has an empty password by default and a user should always set the password after the first login.

A user must provide authentication credentials (name and password) before connecting to the target device from RexDraw, RexView or RexHMI Designer or when using the integrated web interface. An error is returned and the connection is closed if invalid credentials are submitted.

5.2 Using Unix Accounts

In the GNU/Linux environment, the authentication subsystem may be attached to standard accounts defined by passwd/shadow files by enabling configuration option `auth.allowsystem`. To put a user into a specific role, following system group must be present and corresponding user account must a member of that group:

- *admin* - `rex-admin`,
- *super* - `rex-super`,
- *operator* - `rex-operator`,
- *guest* - `rex-guest`.

In the GNU/Linux environment, a root system account may also be bound to the admin user of the REX Control system by enabling configuration option `auth.allowroot`.

Example: To bound a "pi" system account with the "admin" role of the REX Control System in the Raspberry Pi device, just create a system group "rex-admin" and add the user "pi" as a member with a shell command "`usermod -a -G rex-admin pi`". It is then possible to login into the REX Control system using the user name "pi". The password may be changed with a shell command "`passwd`".

Chapter 6

Security

A SSL protocol should always be the choice when communicating with the target device over Internet or untrusted network. The only operation user has to perform to enable the SSL protocol in RexCore is to place a certificate and a private key into the target device.

A self-signed certificate is created during the installation process on GNU/Linux and there is nothing that has to be done if a self-signed certificate is sufficient. On Windows, a *RexSecurityConfig* application must be used to generate a self-signed certificate.

If a custom certificate and a private key is available, a user has to put them into the RexCore configuration directory. A file with the certificate should be named `rexcore.cer` and a file with the private key should be named `rexcore.key`. Both files should be in the PEM file format. The RexCore should be restarted every time a file with certificate or private key is changed.

List of Figures

Bibliography

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

[2] REX Controls s.r.o.. *Getting started with REX*, 2009.

[3] REX Controls s.r.o.. *Getting started with REX on Raspberry Pi*, 2013.

[4] REX Controls s.r.o.. *Getting started with REX on Intellisys PIO*, 2014.

[5] REX Controls s.r.o.. *RexDraw – User manual*, 2017.

[6] REX Controls s.r.o.. *REST API of the REX Control System – Reference Manual*, 2017.

[7] REX Controls s.r.o.. *RexHMI – User manual*, 2017.

Documentation reference number: 8269