

Copyright (c) 2008-2013, IDTRONIC GmbH
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software SDK is a example implementation for the IDTRONIC Reader-Host-Protocol. The SDK is provided as open source and follows the license terms of above.

Version v1.08

The provide protocol handler consists of the following files:

rfe/protocol/CSrfeProtocolHandler.cs
The implementation of the protocol handler. This class represents all functions of the Reader-Host-Protocol in separate methods.

rfe/protocol/CSrfeMessageQueue.cs
Used by the protocol handler to store incoming messages.

rfe/protocol/CSrfeProtocolConstants.cs
The class contains all constants defined in the protocol description.

rfe/trace/ITraceInterface.cs
Abstract class of a trace class. Subclass this class to implement custom tracing.

rfe/device/IProtocolDeviceInterface.cs
Minimal interface that must be implemented by an IODevice that communicates with the reader. This can be a serial port, tcp client, ...
Just subclass this interface to create the custom communication.

rfe/CSrfeReaderInterface.cs
Global class with global variables and easy access to trace capability.

For test of the provided protocol handler the following files are attached:

impl/ConsoleTrace.cs
Simple implementation of the ITraceInterface. Traces everything out to the Console.

impl/SerialDevice.cs
Simple implementation of the IProtocolDeviceInterface. Implements the communication with a serial port.

Program.cs
Test class for the protocol handler with a console user interface.