



acontis technologies GmbH

Real-time Accelerator Windows LwIP

User Manual

Edition: 2024-07-10

© Copyright **acontis technologies GmbH**

Neither this document nor excerpts therefrom may be reproduced, transmitted, or conveyed to third parties by any means whatever without the express permission of the publisher. At the time of publication, the functions described in this document and those implemented in the corresponding hardware and/or software were carefully verified; nonetheless, for technical reasons, it cannot be guaranteed that no discrepancies exist. This document will be regularly examined so that corrections can be made in subsequent editions. Note: Although a product may include undocumented features, such features are not considered to be part of the product, and their functionality is therefore not subject to any form of support or guarantee.

Content

1	Introduction	4
1.1	Overview	4
1.2	Requirements	4
2	Tool usage	5
2.1	Prerequisites	5
2.2	LwIP Demo.....	6
3	Project structure.....	8
4	Sample Code	9
4.1	LwIP Demo.....	9
4.2	Architecture	9
4.3	Command line interface	10
4.4	Custom LwIP Stack Version.....	11

1 Introduction

1.1 Overview

This document describes how the “Rtacc LwIP” can be used to create your own real-time application based on our optimized link layers and lwIP (lightweight TCP/IP stack).

RtaccWinLwipDemo is a demonstration program, which is able to run:

- PingApp from the LwIP stack
- TCP client/server application to send/receive some frames over TCP
- UDP client/server application to send/receive some frames over UDP
- Performance measurement application by sending a UDP frame to an EtherCAT slave

RtaccWinLwipDemoWithWinsock is the same demonstration program but is compiled to use the original WinSock implementation from Microsoft, to demonstrate how easily and existing code can be converted.

1.2 Requirements

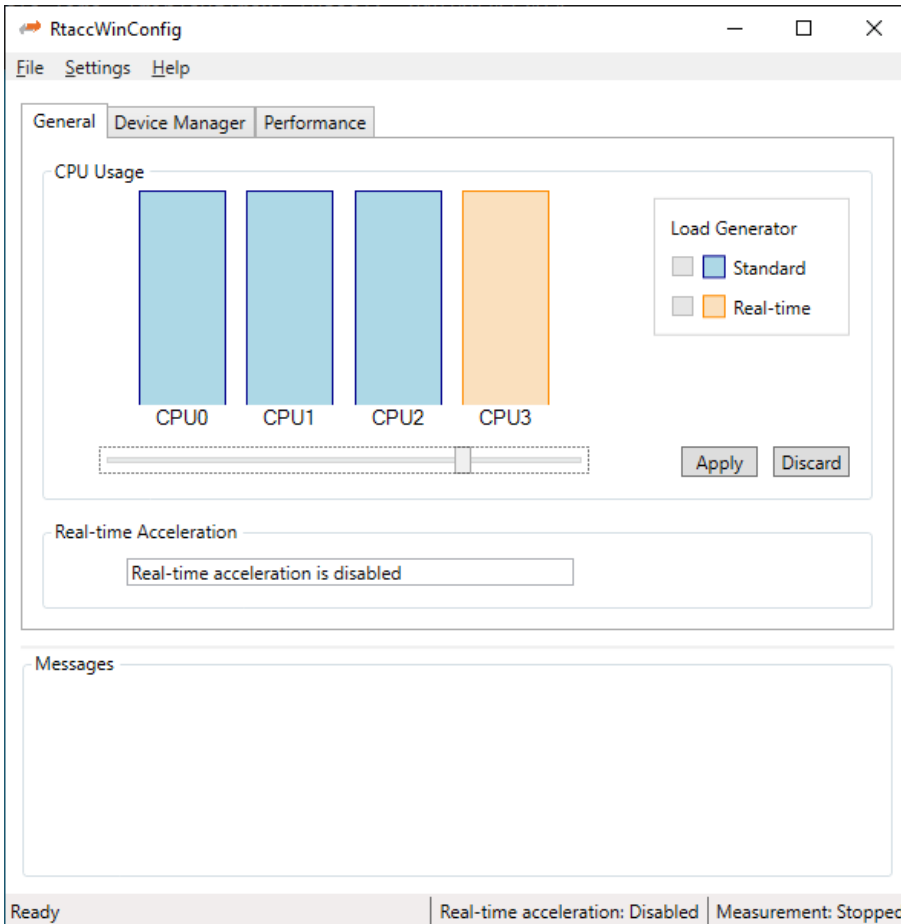
- For Developers
 - Microsoft Windows 10 and above
 - Visual Studio 2015 and above
- For Deployment
 - Microsoft Windows 10 and above
 - Microsoft .NET Framework 4 Client Profile
 - Screen resolution at least 1024x768 pixel
 - Memory as recommended for operating system

2 Tool usage

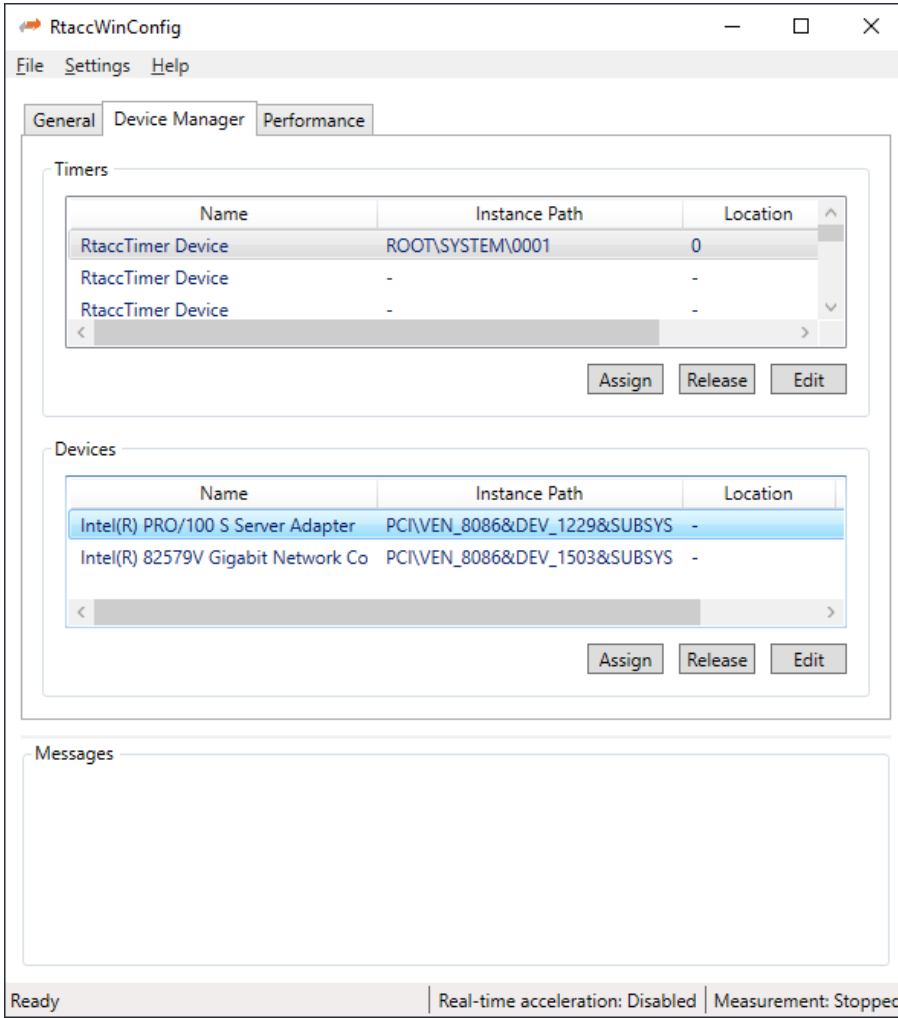
2.1 Prerequisites

Install first the “RealtimeAcceleratorSetup” from “Bin” folder. This installs the “RtaccWinConfig” tool.

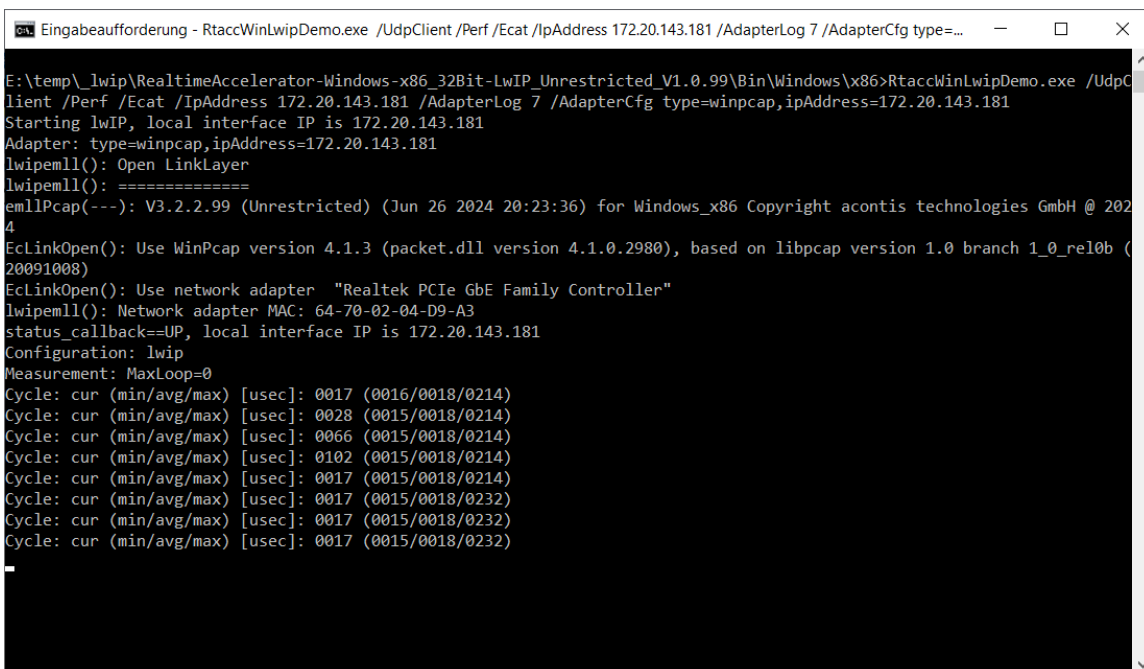
This tool can be used e.g. to isolate one or more cores for realtime usage (might be not necessary if the non-isolated mode is enough):



... or it can be used to install the required timer or device drivers:



2.2 LwIP Demo



RtaccWinLwipDemo.exe [options]

The supported options are described in chapter “4.3 Command line interface”.

3 Project structure

- \Bin
 - RealtimeAcceleratorSetup.msi
 - Rtacc Installer, which is required to install the required drivers
 - RtaccDevice.dll
 - RtaccDevice Interface Library
 - RtaccTimer.dll
 - RtaccTimer Interface Library
 - RtaccWinLwipDemo.exe
 - RtaccWinLwipDemo Application
 - RtaccWinLwipDemoWithWinsock.exe
 - RtaccWinDeviceDemo Application built to Winsock interface
- \Doc
 - Manuals
- \Examples\RtaccWinLwipDemo\RtaccWinLwipDemo.sln
 - Visual Studio 2015 Solution File for the RtaccWinLwipDemo Application
- \SDK\INC
 - Header files for LwIPstack, LwIPeml, RtaccDevice & RtaccTimer interface
- \SDK\LIB
 - Libraries files for LwIPstack, LwIPeml, RtaccDevice & RtaccTimer interface

4 Sample Code

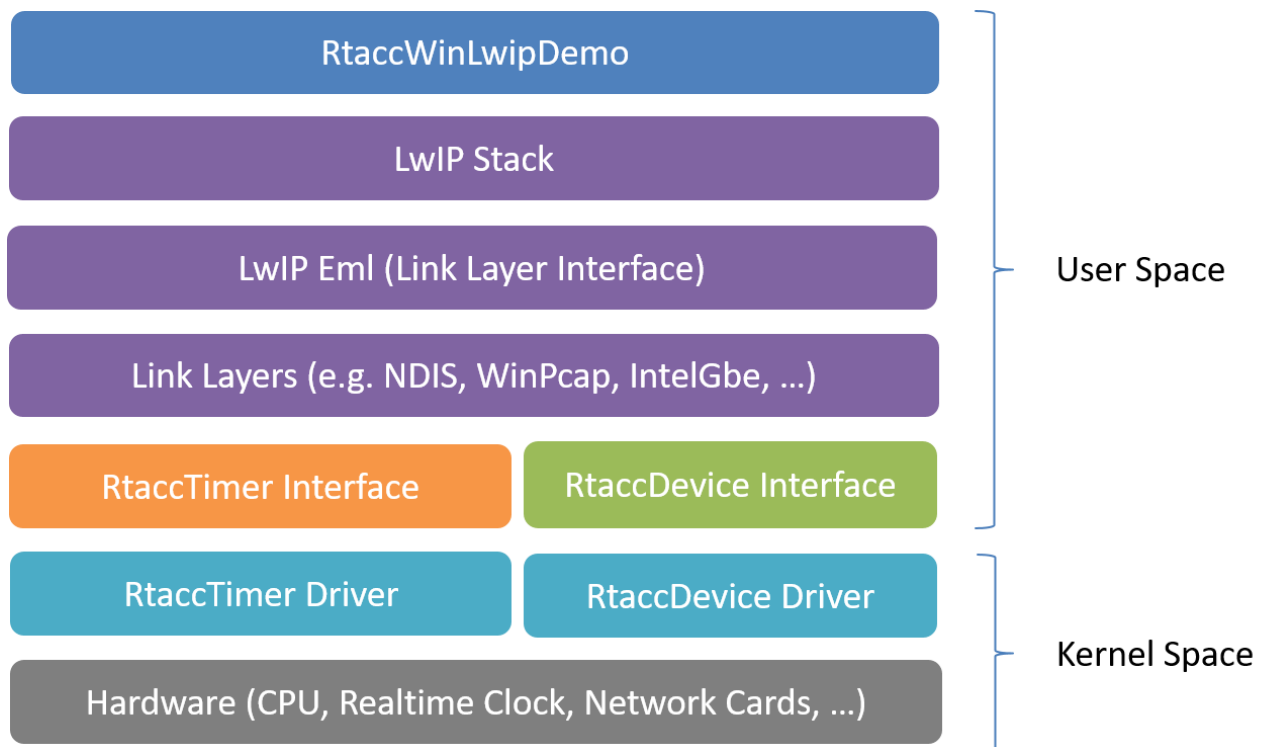
4.1 LwIP Demo

Open the Visual Studio 2015 Solution (“Examples\RtaccWinLwipDemo\RtaccWinLwipDemo.sln”) and do the following things:

- Compile the solution (Output folder of the “Debug” configuration is the root folder, where all the binaries are located)
- Run “RtaccWinLwipDemo.exe” without parameters to show usage

4.2 Architecture

SDK contains the following components:



- Sample Code
 - RtaccWinLwipDemo.exe
 - Demo application, which demonstrates how to launch LwIP stack with the optimized link layers.
- SDK
 - LwIPstack
 - lwIP - A Lightweight TCP/IP stack, version 2.2.0
 - LwIPeml
 - Link Layer Interface, which can be used to load the optimized link layers
 - Supported link layers are:
 - Winpcap (WinPcap or Npcap)
 - ndis (NDIS)
 - i8254x (Intel Pro 1000 family)
 - i8255x (Intel Pro 100 family)
 - intelgbe (Intel Gbe family)
 - rtl8169 (Realtek RTL8169)
 - ccat (Beckhoff CCAT)
 - RtaccTimer Interface (RtaccTimer.dll)
 - Provide access to RtaccTimer driver, which can be used to access a realtime timer
 - RtaccDevice Interface (RtaccDevice.dll)

- Provide access to RtaccDevice driver, which can be used to access hardware like network cards directly from user space

4.3 Command line interface

The command line interface supports the following arguments:

- /Help (or without parameters)
 - Display usage
- /PingApp
 - Enable ping app
- /TcpClient
 - Enable TCP client
- /TcpServer
 - Enable TCP server
- /UdpClient
 - Enable UDP client
- /UdpServer
 - Enable UDP server
- /Perf
 - Enable performance test
- /Ecat
 - Enable ECAT frame over UDP to an EtherCAT slave e.g. EK1100
- /Dhcp
 - Enable DHCP
- /ProcPriority
 - Set process priority to realtime
- /ProcAffinity
 - Set process affinity e.g. 7 for 8th core
- /ThdPriority
 - Set thread priority to time critical
- /Gateway
 - Gateway e.g. 192.168.1.1
- /IpAddress
 - IP address e.g. 192.168.0.200
- /NetMask
 - Netmask e.g. 255.255.255.0
- /AdapterLog
 - Adapter log level e.g. 5 (0...8)
- /AdapterCfg
 - Adapter configuration, a comma separated key value list
 - type: Link layer type
 - ipAddress: IP Address
 - instance: Instance number
 - Supported Link Layers
 - WinPcap e.g. "type=winpcap,ipAddress=192.168.1.200"
 - NDIS e.g. "type=ndis,ipAddress=192.168.1.200"
 - I8254x e.g. "type=i8254x,instance=1"
 - I8255x e.g. "type=i8255x,instance=1"
 - IntelGbe e.g. "type=intelgbe,instance=1"
 - RTL8169 e.g. "type=rtl8169,instance=1"
 - CCAT e.g. "type=ccat,instance=1"

NOTE:

All link layers will be loaded dynamically at runtime. This means the all link layers can be copied to the bin folder or the environment variable PATH can be modified in that way:

PATH=C:\Program Files\acontis_technologies\EC-Master-Windows-x86_64Bit\Bin\Windows\x64;%PATH%

Examples

- Ping app with WinPcap link layer on network adapter with IP address 192.168.0.1

- RtaccWinLwipDemo /PingApp /IpAddress 192.168.0.1 /AdapterLog 7 /AdapterCfg type=winpcap,ipAddress=192.168.0.1
- TCP client/server demo with WinPcap link layer on network adapter with IP address 192.168.0.1
 - RtaccWinLwipDemo /TcpClient /TcpServer /IpAddress 192.168.0.1 /AdapterLog 7 /AdapterCfg type=winpcap,ipAddress=192.168.0.1
- UDP client/server demo with WinPcap link layer on network adapter with IP address 192.168.0.1
 - RtaccWinLwipDemo /UdpClient /UdpServer /IpAddress 192.168.0.1 /AdapterLog 7 /AdapterCfg type=winpcap,ipAddress=192.168.0.1
- TCP client/server demo and performance measuring with WinPcap link layer on network adapter with IP address 192.168.0.1
 - RtaccWinLwipDemo /TcpClient /TcpServer /Perf /IpAddress 192.168.0.1 /AdapterLog 7 /AdapterCfg type=winpcap,ipAddress=192.168.0.1
- UDP client/server demo and performance measuring with WinPcap link layer on network adapter with IP address 192.168.0.1
 - RtaccWinLwipDemo /UdpClient /UdpServer /Perf /IpAddress 192.168.0.1 /AdapterLog 7 /AdapterCfg type=winpcap,ipAddress=192.168.0.1
- UDP client demo, sending EtherCAT frames to connected EtherCAT slave and performance measuring with WinPcap link layer on network adapter with IP address 192.168.0.1
 - RtaccWinLwipDemo /UdpClient /Perf /Ecat /ProcPriority /ProcAffinity 7 /ThdPriority /IpAddress 192.168.0.1 /AdapterLog 7 /AdapterCfg type=winpcap,ipAddress=192.168.0.1

4.4 Custom LwIP Stack Version

The SDK was built with LwIP stack version 2.2.0. It can be easily replaced by another version:

- Download CMake binary from <https://cmake.org/>
- Download LwIP stack from <https://savannah.nongnu.org/projects/lwip/>
- Unzip LwIP stack
- Open console window in LwIP directory and enter the following commands
 - Make build directory
 - mkdir build
 - Build project files for Visual Studio 2015
 - CMAKEDIR\bin\cmake .. -G "Visual Studio 2015"
 - Create configuration header file
 - Copy "contrib\examples\example_app\lwipcfg.h.example" to "contrib\examples\example_app\lwipcfg.h"
- Compare header files with the header files provide with this SDK to see which options were changed (optional)
- Open solution and compile it
- Adjust demo to use the custom LwIP stack