**PL „Autonomous Intelligent Systems“**

**„Programming Red Pitaya“**

**Manual**

**Created by**

Daniel Schäfer, SS 2021

Inhaltsverzeichnis

[1. Goals 3](#_Toc70584406)

[2. 3rd Party Software 4](#_Toc70584407)

[2.1. Used Software 4](#_Toc70584408)

[2.2. Software Core 5](#_Toc70584409)

[2.3. How-To 5](#_Toc70584410)

[2.4. Change the RP software 7](#_Toc70584411)

[2.5. Source code 10](#_Toc70584412)

# Goals

The goal for this project is to realize an online software for **Red Pitaya** (**RP**) System, which is able to start the ultra-sonic sensor “SRF-02” over I²C connection and to read out the reflected signal received by the sensor itself. The acquired data has to be saved temporarily on a RP[[1]](#footnote-1) System and will be analyzed internally after acquiring.

The SW on the RedPitaya is able to run as a daemon in background and deliver data to any PC in the foreground via UDP. The GUI can save data in a text-file and visualize current data from the RP.

# 3rd Party Software

All used software for the project is provided under the GNU General public license. The RP runs a Linux operating system so there a GCC [[2]](#footnote-2)(GNU compiler collection) is used to compile C(++) code. The provided makefile in the directory is created by Daniel Schäfer, but offered as a part of the software and distributed under the GNU General license.

## Used Software

To read, write, modify and save any kind of software, the following programs are used in this project:

1. Notepad++[[3]](#footnote-3)
2. WinSCP[[4]](#footnote-4)
3. Putty (portable)
4. MS Visual Studio 2019

***WinSCP*** works on windows platforms and is also available in different install-packages for windows. It uses the “secure copy protocol” to provide a remote view of a PC in the same network. The SCP server on the RP is the counter part for the client version on the windows machine.

***Notepad++*** is a very powerful software which can not only be used as a simple text-editor. The principal function based on the file-extension which is opened. In case of an opened C-file the extension “.c” is used, so the Notepad++ starts new text-parsing. Different key-words are highlighted to get the feeling of an IDE. The precompiler-defines have another color than “void” or the variable declaration “uint8\_t”. The programmer is also able to see the hierarchy of different used brackets for example the “if” or “switch” cases.

***Putty (portable)*** is used to establish connections like SSH, Telnet, FTP and RAW formats. The core idea is to have at least one single program used as an “all in one” application. Only one part out of more than 8 functions is used, the SSH connection. The RP offers an SSH connection as a server on default. The remote Secure Shell can be entered via Telnet.

**MS Visual Studio 2019** is used to create a C# project with all necessary software bundles needed for communication to / with RP.

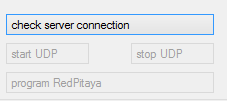
## Software Core

All Software for RP is written in C. The software for Microsoft machines is written in C#. On RP the sw runs as a daemon in background. Three threads are used to let the binary code run on the system. With the first thread main is started and two new threads are created.

The second thread is listening on port 61231 by a UDP server. The thread is in blocked mode until a message comes in via UPD. The message is parsed and gives a signal to the third thread to run.

The third thread is waiting for a signal from the server thread, which is able to deliver a command and a value. For example, “-a 1” means: “start ADC output”.

## How-To

1. Establish a WLAN connection to the access point of the Red Pitaya
2. Open the GUI
3. Check is a sensor is “online” 
4. If successful, the buttons “start UDP”, “stop UDP” and “program RedPitaya” should be enabled
   1. If enabled, the RP can be programed with the current binaries by pushing “program RedPitaya” Ein Bild, das Text enthält.

      Automatisch generierte Beschreibung. Then proceed with step 5.)
   2. If still disable, check WLAN, RP and start from beginning
5. To start the process (it is running as a daemon in the background) push the “start UDP” button. Short period later the led in the back of the sensor should blink red with 10Hz
   1. If the sensor is running, proceed with 6.)
   2. If the sensor is not running, repeat step 4.)
6. To start the local UDP server, push “start client”
7. If the client is started normally, the right part of the GUI is activeEin Bild, das Text enthält.

   Automatisch generierte Beschreibung The first command “-c” has to be sent to get information about the SW-Version and the frequency data (minimum, maximum and frequency-factor for FFT scaling).  
   After pushing “send command”, the lable “V0.XXx” should change into the actual SW version like that:
   1. If SW version is shown, the daemon is working and the UDP client is working as expected.
   2. If SW version is not shown, repeat steps from 4.) on and press the “stop UDP” button to also provide a clean shutdown of the daemon.
8. Push “start FFT”, “start ADC” or “start measure” and be inspired!

## Change the RP software

If the SW of the RP should be change, open an SCP connection by using WinSCP and navigate to the folder “/root/iic”. IP address of the RP has to be entered and the port 22 is used. For user authentication user “root” with PW “root” is used. The folder “iic/src” contains all necessary code, which can be changed and tested. In parallel a shell connection has to be opened by using a program similar to “putty”. Open a connection by also using user “root” and PW “root” to enter remote command shell (SSH).

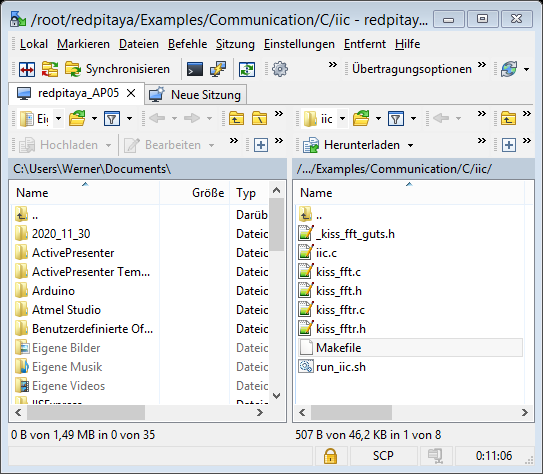


Figure 1 WinSCP connection to RP

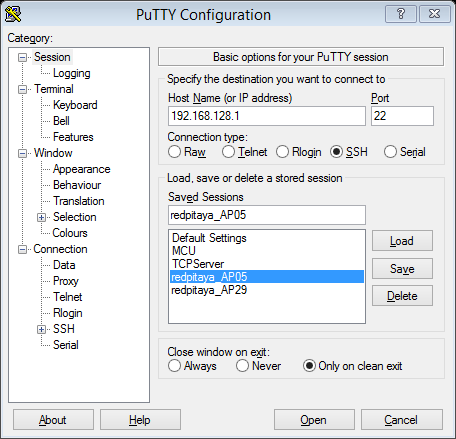


Figure Putty configuration

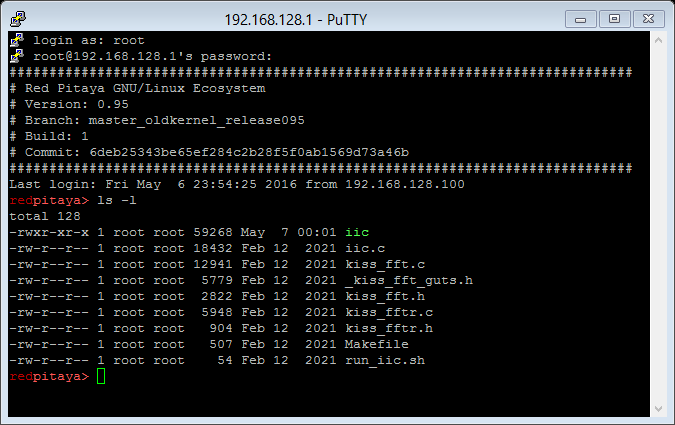


Figure 3 Command line in RP

If the command “ls -l” is executed, the content of the current folder is shown. By typing in “  
**cd “/root/iic”**

The correct folder is entered. Now the command “make” can be executed like below.

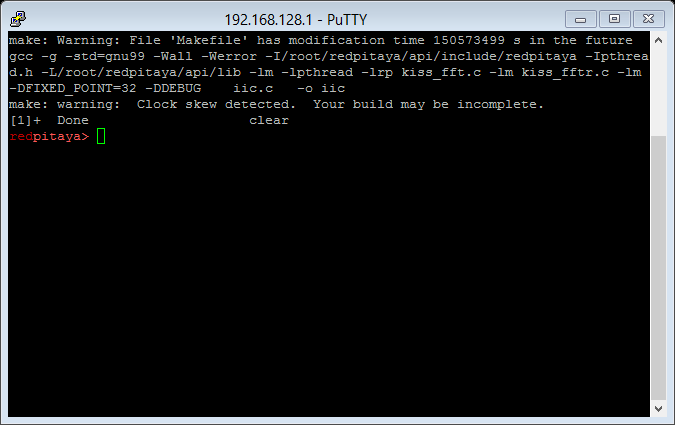


Figure 4 Successful compiling of the SW

One warning can appear, but no doubts about that. The RP has no chance to establish a connection to any time-server (NTP-server), so the time starts at 1900 or something like that.

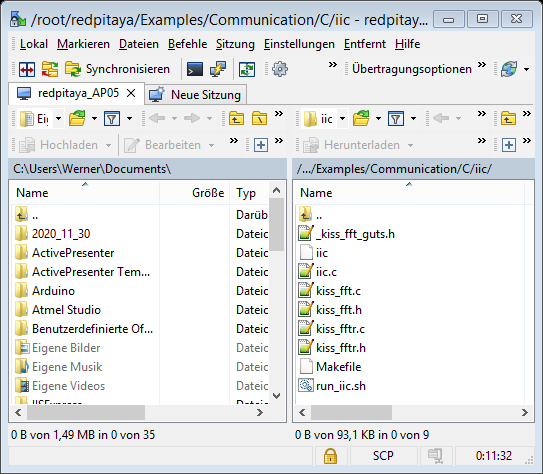


Figure 5 WinSCP shows iic

In WinSCP, after updating the screen, the binary file “iic” is built and can be seen.

Now, the GUI is able to start the new iic by pushing button “start UDP”.

That´s it!

## Source code

The source code can be found in a 7-zip format as an additional appendix on the platform Moodle. It is uploaded with the filename “Source-Code\_AIS201920\_SchaeferD.7z”

Following data is placed inside (under “/bin/Debug/proc”)

* inc
  + \_kiss\_fft\_guts.h
  + iic.h
  + kiss\_fft.h
  + kiss\_fftr.h
  + main.h
  + save\_data.h
  + udp\_server.h
* lib
  + librpxx.so
* src
  + iic.c
  + kiss\_fft.c
  + kiss\_fftr.c
  + main.c
  + save\_data.c
  + udp\_server.c
* cronjob
* Makefile

1. https://www.redpitaya.com [↑](#footnote-ref-1)
2. https://gcc.gnu.org [↑](#footnote-ref-2)
3. https://notepad-plus-plus.org [↑](#footnote-ref-3)
4. https://winscp.net/eng/index.php [↑](#footnote-ref-4)