



# RO-ETH-CUSTOM-SAMPLE

Software-Description

2023 September

# INDEX

<b><u>1. Introduction</u></b>	<b>3</b>
1.1. Foreword	4
1.2. Customer satisfaction	4
1.3. Customer response	4
<b><u>2. Software</u></b>	<b>6</b>
2.1. Create custom firmware	7
2.1.1. Create custom firmware	7
2.1.2. Folder-structure	8
2.1.3. Custom HTML-pages	9
2.1.4. Custom c-programs	9
2.1.5. Create custom firmware	10
2.2. Firmware upload	11
2.3. Transmitting variables via web interface	13
2.3.1. Transmitting variables via web interface	13
2.3.2. HTML/Javascript programming	15
2.3.2.1. Programming example "Projekt Name" in HTML/Javascript - Readonly	15
2.3.2.2. Programming example "Test Variable 1" in HTML/Javascript - Read/Write	16
2.3.3. C-programming	17
2.4. Custom log	18
2.4.1. Custom log	18
2.4.2. Writting to log file via C-programming	20
2.5. Network configuration	21
<b><u>3. Appendix</u></b>	<b>23</b>
3.1. Revisions	24
3.2. Copyrights and trademarks	25

# Introduction

---



# **1. Introduction**

---

## **1.1. Foreword**

**Congratulations on purchasing a high quality DEDITEC product!**

Our products are developed by our engineers according to today's required quality standards. We pay attention already during the development to flexible expandability and long availability.

**We develop modular!**

Due to a modular development we shorten the development time and - what of course benefits the customer - we sell at a fair price!

**We ensure a long delivery availability!**

If used semiconductors are no longer available, we can react faster. With us mostly only modules have to be redesigned and not the whole product. This increases the delivery availability.

## **1.2. Customer satisfaction**

**A satisfied customer is our first priority!**

If something is not to your satisfaction, just contact us by phone or mail.

We will take care of it!

## **1.3. Customer response**

The best products grow with our customers. We are always grateful for any suggestions or proposals.



# Software

---



## **2. Software**

### **2.1. Create custom firmware**

#### **2.1.1. Create custom firmware**

In order to transfer custom content (software or HTML-pages) to the module, we provide a "ro-eth-custom-sample" package with all required files.

This package contains a fixed folder structure, and several shell scripts, which simplify the creation of the firmware.

### 2.1.2. Folder-structure

```
|
|-> /file-system/
|   |
|   |-> /custom/
|       |
|       |-> /bin/
|       |-> /www/
|           |
|           |-> /www/
|-> /firmware/
|   |
|   |-> /delib-linux/
|       |
|       |-> /library/
|           |
|           |-> /source/
|
|-> make_firmware_update_file.sh
```



### 2.1.3. Custom HTML-pages

In order to use custom HTML-pages via the web interface, they have to be stored in the following directory:

`/file-system/custom/www/www`

Note:

Please note, that your "start page" is named index.html, otherwise this page will not be accessible via the web interface.

If you want to transfer multiple HTML-pages to the RO-ETH module, they need no special file-name, but these pages have to be accessible from the "index.html" (e.g. via link).

### 2.1.4. Custom c-programs

In order to transfer custom c-programs to the module, all project-files have to be stored in the following directory:

`/firmware/source/`

In Addition, you can find in the folder `/firmware/` a shell script "compile\_and\_copy\_to\_nfs.sh" to compile the project under Linux.

Note:

All data, which should be compiled have to be added manually to the shell script "compile\_and\_copy\_to\_nfs.sh"

### 2.1.5. Create custom firmware

In order to transfer custom content (software or HTML-pages) to the module, you can find in the main directory a shell script called "make\_firmware\_update\_file.sh".

This shell script zips the content of the folder structure to a archive, which can be transferred to the module via firmware upload (-> see firmware upload).

Note:

Please note, that the file name of the firmware is always "RO-ETH\_CUSTOM1.tar".

## 2.2. Firmware upload

To transfer new software content to the module, proceed as follows:

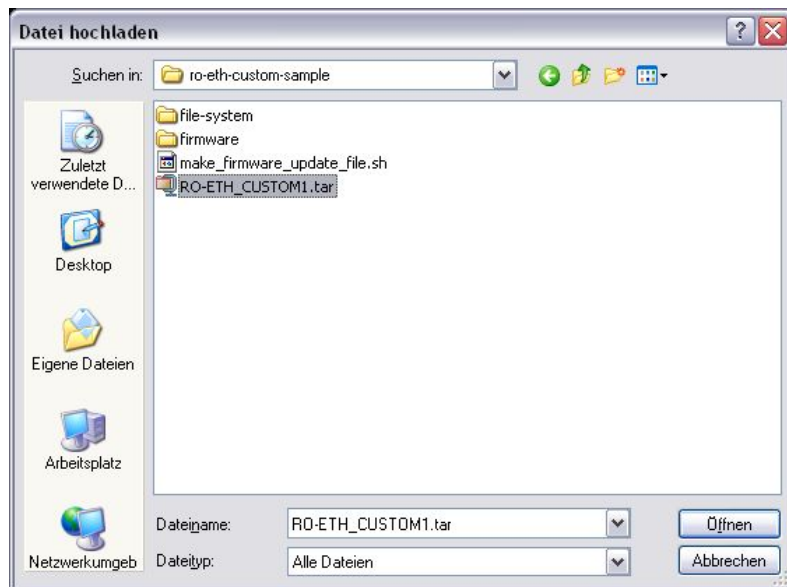
- 1) Open the web interface of the DEDITEC module.
- 2) Click on the navigation "Configuration" at the top of the page. After that, button "firmware update" in the left navigation

The following window will be opened:



- 3) Click on the button "Browse .." and select the firmware you want to transfer to the module.

Note that the firmware has the name RO-ETH\_CUSTOM1.tar.



4) Now click on the button "firmware update". The firmware will be automatically transferred to the module. Finally, a status message appears.

5) Re-open the web interface of the module by entering the IP address in the Internet browser.

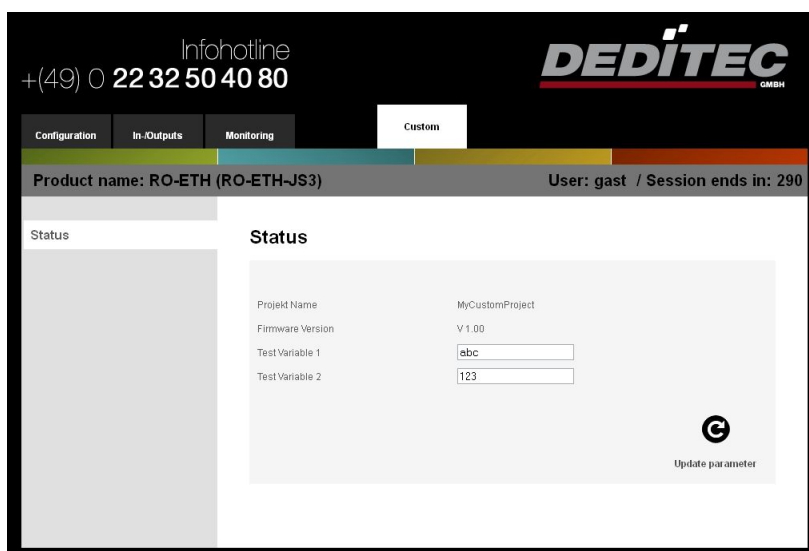
## 2.3. Transmitting variables via web interface

### 2.3.1. Transmitting variables via web interface

Start the web interface of the DEDITEC module. Therefore, type the IP address of the DEDITEC module into your Internet browser. The default IP address of the DEDITEC module is 192.168.1.1 (-> change IP address of the module)

Then click on the navigation "Custom" at the top of the page.

The following window will be opened:



In this example, the following variables can be transmit from the web interface to the C application (or vice versa).

#### Project Name (cust1\_product\_name)

Since the value of this variable is output in a label in HTML, it can not be changed via the web interface. This value can be changed only by the C program.

#### **Firmware version (cust1\_fw\_rev)**

This value can be changed only by the C program.

#### **Test variable 1 (cust1\_testvar\_text)**

This value can be changed either via the web interface, as well as the C program. In this case example, the variable is from the data type "char []"

#### **Test variable 2 (cust1\_testvar\_long)**

This value can be changed either via the web interface, as well as the C program. In this case example, the variable is from the data type "Long"

#### **Note:**

Changed parameters are transmitted only to the module after clicking the button Update parameter

## 2.3.2. HTML/Javascript programming

### 2.3.2.1. Programming example "Projekt Name" in HTML/Javascript - Readonly

```
<div id="content-left">$$C_cust1_product_name$$</div>
```

#### **Note:**

On loading the HTML page, all placeholders, which are labeled with \$\$C\_ and \$\$, are filled automatically with the according variable values.

(Example: \$\$C\_random\_value\$\$ will be filled automatically with the value of the variable random\_value, if a value is defined for this variable in the file "deditec.cgf")

#### 2.3.2.2. Programming example "Test Variable 1" in HTML/Javascript - Read/Write

```
<input type="text" name="C_cust1_testvar_text"
value="$$C_cust1_testvar_text$$" />
```

**Note:**

Please note the following notation for the input field, writing a variable in the web interface:

```
<input name="C_random_value" value="$$C_random_value$$" />
```

In this example, the value of "value" is written in the variable random\_value . As well as the read-only example (see example above), this field will be filled with the value of the variable random\_value on load of the HTML page.



### 2.3.3. C-programming

Programming example "Projekt Name" in C - Write

```
#define CUSTOM1_PARAM_PRODUCT_NAME  "cust1_product_name"
parameter_set_text(CUSTOM1_PARAM_PRODUCT_NAME,
strlen(CUSTOM1_PARAM_PRODUCT_NAME), "MyCustomProject",
strlen("MyCustomProject"));          // write variable ("Projekt Name" @ Weboberfläche)
```

Programming example "Test Variable 1" in C - Read

```
#define CUSTOM1_TEST_PARAM_TEXT      "cust1_testvar_text"
char test_text_variable[20];
ret = parameter_get_text(CUSTOM1_TEST_PARAM_TEXT,
strlen(CUSTOM1_TEST_PARAM_TEXT), test_text_variable, strlen(test_text_variable)); // read
variable ("Test Variable 1" @ Weboberfläche)
```

Note:

When #define, you must use the same name of the variable as in the web interface.

(Example: #define CUSTOM1\_RANDOM\_VALUE "random\_value", in reference to the HTML/Javascript example on previous page)

You can find the function to read/write parameters in the file deditec\_parameter\_io.c in the directory /firmware/library/vc/

## 2.4. Custom log

### 2.4.1. Custom log

The RO-ETH module has by default a log book, in which all important status messages of the module are written to.

The log book can be viewed directly via the web interface of the module. Therefore, click on the navigation "Configuration" at the top of the page, after that the left navigation button "Log's"

The following window will be opened:

Configuration In-/Outputs Monitoring Custom

Product name: RO-ETH (DEDITEC RO-ETH Module) User: / Session ends in:

General

Network configuration

Network time (NTP)

Mail-Server

FW-Update

Status

Log's

### LOG

Modul Status: OK

Time	Application	Information
01.01.1970 00:00:38	web-log-access	log_init
01.01.1970 00:00:37	Param-Srv	read param: "product_cfg_stepp" succeeded and returns value: "0"
01.01.1970 00:00:37	Param-Srv	read param: "product_cfg_ad" succeeded and returns value: "0"
01.01.1970 00:00:37	Param-Srv	read param: "product_cfg_ar" succeeded and returns value: "0"
01.01.1970 00:00:37	Param-Srv	read param: "product_cfg_di_cnt" succeeded and returns value: "0"
01.01.1970 00:00:37	Param-Srv	read param: "product_cfg_do" succeeded and returns value: "0"
01.01.1970 00:00:37	Param-Srv	read param: "product_cfg_d" succeeded and returns value: "0"
01.01.1970 00:00:37	Param-Srv	read param: "product_cfg_d" succeeded and returns value: "0"
01.01.1970 00:00:37	Param-Srv	read param: "board_name" succeeded and returns value: "DEDITEC RO-ETH Module"
01.01.1970 00:00:37	Param-Srv	read param: "product_name" succeeded and returns value: "RO-ETH"
01.01.1970 00:00:37	webcmd	query_parameter - function begin
01.01.1970 00:00:37	webcmd	webcmd begin-->
01.01.1970 00:00:37	webcmd	log_init
01.01.1970 00:00:34	Param-Srv	writing in param: "fw_rev_parameter_webrequest", value: "1.00" (len=4 dez)
01.01.1970 00:00:34	webcmd	log_init

Get complete Log Get last 1000 Log Entries Clear Log

In order to view only entries of the custom firmware, click on the tab "Custom".



### 2.4.2. Writting to log file via C-programming

Enclosed, you can find an explanation, how to write in the log-book of the RO-ETH module via your own C-program.

#### 1) initialization of the log-book

To write to the log-book, it must first be initialized.

##### Programming sample

```
log_init("CUST-FW", 1); // initialize log
with the name "CUST-FW" * details: see
"/library/vc/deditec_log.h"
```

##### Note:

Please note to initialize the log-book that the name "CUST-FW". Only with this initialization, the log-book content can be viewed later in the web interface in the navigation "Log's" under the tab "Custom"

#### 2) Writing to the log-book

With the function "log\_write", you have the opportunity to write a text buffer to the log-book.

##### Programming example

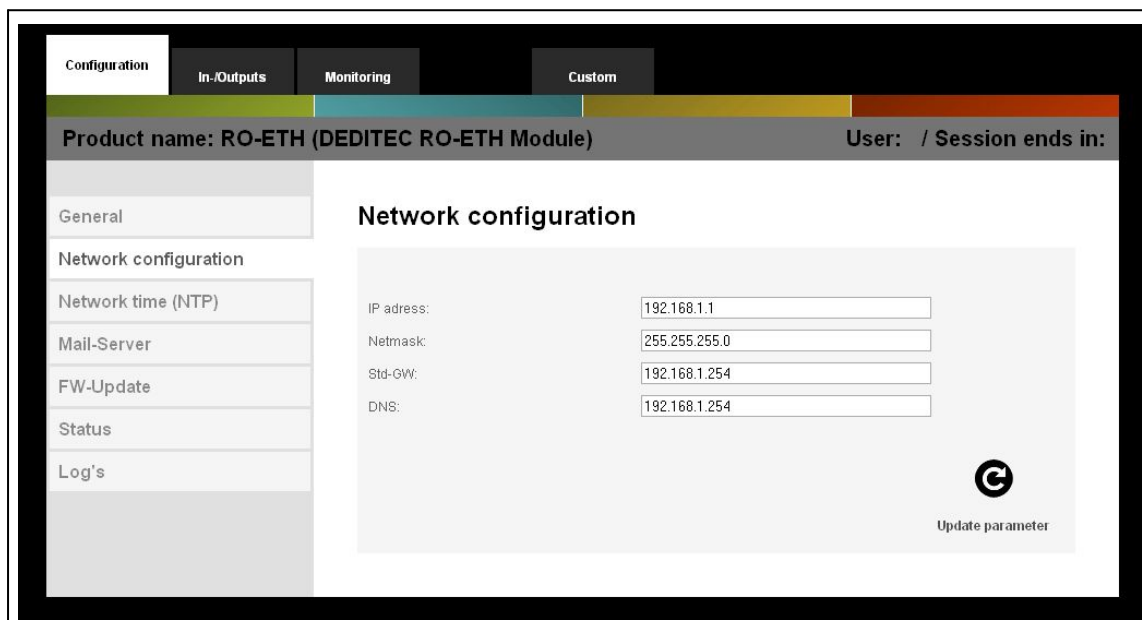
```
char msg[200];

sprintf(msg, "Starting custom1 firmware"); // writing a
String in the buffer "msg"
log_write(msg, 1); // writing the
buffer "msg" to LOG * details: see
"/library/vc/deditec_log.h"
```

## 2.5. Network configuration

The network configuration of the RO-ETH module can be made via the web interface. Therefore, click on the navigation "Configuration" at the top of the page, after that the left button 'Network configuration'.

The following window will be opened: (in this example with the default network configuration):



The screenshot displays the DEDITEC web interface for network configuration. At the top, there is a navigation bar with tabs: Configuration, In-/Outputs, Monitoring, and Custom. Below this, a status bar shows 'Product name: RO-ETH (DEDITEC RO-ETH Module)' and 'User: / Session ends in:'. On the left side, a sidebar menu lists options: General, Network configuration (selected), Network time (NTP), Mail-Server, FW-Update, Status, and Log's. The main content area is titled 'Network configuration' and contains four input fields: IP address (192.168.1.1), Netmask (255.255.255.0), Std-GW (192.168.1.254), and DNS (192.168.1.254). At the bottom right of the main area, there is a circular refresh icon and the text 'Update parameter'.

In this section you can configure the settings for the IP address, subnet mask, default gateway and DNS server.

**Note:**

Please note that a default gateway is set. This is also, if the module is located in a separate network (eg. RO-ETH module + PC via a switch/hub).

You should set as default gateway, always the network ID + 254 as host ID (example: IP address of RO-ETH module: 192.168.20.123 -> default gateway: 192.168.20.254).

Note also that the IP addresses with network IDs which ending on 0, are not available. (Ex. 192.168.0.x, 195.123.0.x, 223.45.0. x, ...)

# Appendix



## **3. Appendix**

### **3.1. Revisions**

Rev 3.00

DEDITEC Design Update

Rev 2.00

First issue



### **3.2. Copyrights and trademarks**

Linux is a registered trademark of Linus Torvalds.

USB is a registered trademark of USB Implementers Forum Inc.

LabVIEW is a registered trademark of National Instruments.

Intel is a registered trademark of Intel Corporation.

AMD is a registered trademark of Advanced Micro Devices, Inc.

ProfiLab is a registered trademark of ABACOM Ingenieurbüro GbR.

ispVM System is a registered trademark of Lattice Semiconductor Corporation.

Windows, Visual-C/C++, -C#, -Basic, -Basic.NET and Visual-Studio are registered trademarks of Microsoft Corporation.

Delphi is a registered trademark of Borland Software Corporation.

Java is a registered trademark of Oracle Corporation.