



# Firmware-based Error Logging

**Cyclically reporting the  
Number of CAN Errors**

**Software Manual**

to Product C.2020.xx, C.2025.xx, C.2027.xx



## NOTE

The information in this document has been carefully checked and is believed to be entirely reliable. **esd** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. In particular descriptions and technical data specified in this document may not be constituted to be guaranteed product features in any legal sense.

**esd** reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

All rights to this documentation are reserved by **esd**. Distribution to third parties, and reproduction of this document in any form, whole or in part, are subject to **esd**'s written approval.

© 2017 esd electronic system design gmbh, Hannover

esd electronic system design gmbh  
Vahrenwalder Str. 207  
30165 Hannover  
Germany

Phone: +49-511-372 98-0  
Fax: +49-511-372 98-68  
E-Mail: [info@esd.eu](mailto:info@esd.eu)  
Internet: [www.esd.eu](http://www.esd.eu)

### Trademark Notices

CANopen® and CiA® are registered EU trademarks of CAN in Automation e.V.

All other trademarks, product names, company names or company logos used in this manual are reserved by their respective owners.

<b>Document file:</b>	I:\Texte\Doku\MANUALS\PROGRAM\CAN\NTCAN-Firmware\Englisch\Firmware-based_Error-Logging_Manual_en_10.odt
<b>Date of print:</b>	2017-03-13
<b>Document type number:</b>	DOC0800

<b>Hardware version:</b>	n.a.
<b>Firmware Version:</b>	from 0c.0f / \$093F

## Document History

The changes in the document listed below affect changes in the software as well as changes in the description of the facts, only.

Rev.	Chapter	Changes versus previous version	Date
1.0	-	First English manual	2017-03-13

Technical details are subject to change without further notice.

# Table of contents

1. <a href="#">Introduction</a> .....	5
2. <a href="#">Requirements</a> .....	5
3. <a href="#">Application</a> .....	6
4. <a href="#">Example Program</a> .....	7
5. <a href="#">References</a> .....	8

## Typographical Conventions

Throughout this manual the following typographical conventions are used to distinguish technical terms.

Convention	Example
File and path names	<code>/dev/null</code> or <code>&lt;stdio.h&gt;</code>
Function names	<code><i>open()</i></code>
Programming constants	<code>NULL</code>
Programming data types	<code>uint32_t</code>
Variable names	<code><i>Count</i></code>

## 1. Introduction

The Firmware-based Error Logging supports the cyclically reporting of the absolute number of CAN errors detected by the CAN controller.

This function can e.g. be used to assess the wiring quality of the physical CAN net.

## 2. Requirements

1. The function is implemented in 331-series PCI boards only, e.g.:  
CAN-PCI/331, PMC-CAN/331, CPCI-CAN/331
2. The PCI board must be configured to 2.0B mode.  
If not, it can be switched to 2.0B mode using the following commands:

### Firmware Update for the CAN-PCI/331

#### Files:

updc331.exe

ntcan.dll

updc331 (for Linux)

Switch board to 2.0B mode (Error logging is only supported in 2.0B mode)

Call `"updc331 -tb 0"`

Restart driver or reboot computer:

Execute `"updc331 0"`

The firmware will be active after an additional reboot.

### 3. Application

1. Set CAN baudrate.
2. "enable" `NTCAN_EV_USER+0x03e`
3. Call ***CanSend()*** to send an "EVMSG" with length 2 and "evid" `NTCAN_EV_USER*0x03e` and the cycle time in milliseconds defined in parameter `s[0]` to the board.



#### INFORMATION

Only with ***CanSend()*** you can send User-Events to the CAN board! (***canWrite()***-calls return an error message, because they expect an acknowledge for the CAN message transmitted at the CAN bus. An event does not generate this acknowledge.)

4. A CAN event with the ID `NTCAN_EV_USER*0x03e` is send with the defined frequency for each CAN net of the board.  
Parameter `s[0]` returns the number of detected CAN errors since last firmware start.



#### INFORMATION

If two CAN nets are available a CAN event with the ID `NTCAN_EV_USER*0x03e` has to be send to each of the CAN nets.

A separate event is returned for each net: The events are received using the call ***canRead()*** (provided that parameter `len` is set to 2 and `cmsg*` has enough space for two CMSGs (respektive EVMSGs))

## 4. Example Program

```

#include <stdio.h>
#include <string.h>
#include <ntcan.h>

char *progname;

int main(int argc, char *argv[])
{
    CMSG          cmsg[32];
    NTCAN_RESULT status;
    NTCAN_HANDLE handle;

    unsigned int l;

    int32_t      len;

    progname = argv[0];

    status = canOpen(0, 0, 20, 100, 2000, 2000, &handle);
    if (status != NTCAN_SUCCESS) {
        fprintf(stderr, "%s: Can not open can handle error %d \n", progname, status);
        return(-1);
    }

    status = canSetBaudrate(handle, NTCAN_BAUD_1000);
    if (status != NTCAN_SUCCESS) {
        fprintf(stderr, "%s: Can not set baudrate error %d \n", progname, status);
        canClose(handle);
        return(-1);
    }

    canIdAdd(handle, 0x0e);
    canIdAdd(handle, 0x0f);
    canIdAdd(handle, NTCAN_EV_USER+0x3e);

    /* Insert requested timeout in milliseconds. */

    memset((void *) &cmsg, 0, sizeof(cmsg));
    cmsg[0].len = 2;
    cmsg[0].id = NTCAN_EV_USER + 0x3e + 0x40;
    cmsg[0].data[0] = (1000 >> 0) & 0x0ff;
    cmsg[0].data[1] = (1000 >> 8) & 0x0ff;

    len = 1;
    canSend(handle, &cmsg[0], &len);

    for(l=0;l<20;l++) {

        len = sizeof(cmsg)/sizeof(cmsg[0]);
        status = canRead(handle, &cmsg[0], &len, NULL);
        if ((status == NTCAN_SUCCESS) && (len > 0)) {
            unsigned int l1;

            for(l1=0;l1<len;l1++) {
                printf("Got ID: 0x%08x\n", cmsg[l1].id);

                if (cmsg[l1].id == NTCAN_EV_USER + 0x3e) {
                    unsigned int val;

                    val = cmsg[l1].data[1] & 0x0ff; val <<= 8;
                    val |= cmsg[l1].data[0] & 0x0ff;
                    printf("error counter %d\n", val);
                }
            }
        }
    }

    canClose(handle);
    return(0);
}

```

## 5. References

- /1/ esd electronic system design gmbh, *NTCAN-API Application Developers Manual*, Revision 5.0, 2016