



Firmware-basiertes Error Logging

**Periodische Rückmeldung
der Anzahl von CAN-Bus-Fehlern**

Software-Handbuch

(Ergänzung zum NTCAN-API Application Developers Manual)

zu Produkt C.2020.xx, C.2025.xx, C.2027.xx



Hinweis

Der Inhalt dieses Handbuches wurde mit größter Sorgfalt erarbeitet und geprüft. **esd** übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

esd hat das Recht, Änderungen am beschriebenen Produkt oder an der Dokumentation ohne vorherige Ankündigung vorzunehmen, wenn sie aus Gründen der Zuverlässigkeit oder Qualitätssicherung vorgenommen werden oder dem technischen Fortschritt dienen.

Sämtliche Rechte an der Dokumentation liegen bei **esd**. Die Weitergabe an Dritte und Vervielfältigung jeder Art, auch auszugsweise, sind nur mit schriftlicher Genehmigung durch **esd** gestattet.

© 2017 esd electronic system design gmbh, Hannover

esd electronic system design gmbh

Vahrenwalder Str. 207
30165 Hannover

Tel.: 0511/372 98-0
FAX : 0511/372 98-68
E-Mail: info@esd.eu
Internet: www.esd.eu

Trademark Hinweise

CANopen® und CiA® sind registrierte Unionsmarken des CAN in Automation e.V. .

Alle anderen hier aufgeführten Markenzeichen, Produktnamen, Firmennamen und Firmenlogos sind Eigentum des jeweiligen Rechteinhabers.

Dokument-Datei:	I:\Texte\Doku\MANUALS\PROGRAM\CAN\NTCAN-Firmware\Deutsch\Firmware-basiertes_Error-Logging_Manual_de_10.odt
Datum des Ausdrucks:	2017-03-13

Hardware Version:	n.a.
Firmware Version:	ab 0c.0f / \$093F

Änderungen in den Kapiteln

Die hier aufgeführten Änderungen im Dokument betreffen sowohl Änderungen am Produkt als auch reine Änderungen in der Beschreibung der Sachverhalte.

Rev.	Kapitel	Änderungen gegenüber Vorversion	Datum
1.0	-	Erste Version des deutschen Handbuchs.	2017-03-13

Weitere technische Änderungen vorbehalten

Inhaltsverzeichnis

1. Einleitung	6
2. Voraussetzungen	6
3. Anwendung	7
4. Beispielprogramm	8
5. Referenzen	9

Darstellungskonventionen

In diesem Handbuch werden die folgenden Darstellungen zur Unterscheidung und Hervorhebung der aufgelisteten Programmbestandteile verwendet.

Darstellung von	Beispiel
Datei- und Pfadname	<code>/dev/null</code> or <code><stdio.h></code>
Funktionsnamen	<code>open()</code>
Konstanten	<code>NULL</code>
Datentypen	<code>uint32_t</code>
Variablenamen	<i><code>Count</code></i>

1. Einleitung

Das Firmware-basierte Error-Logging unterstützt die periodische Rückmeldung der absoluten Anzahl von CAN-Fehlern, die vom CAN-Controller registriert worden sind.

Die Funktion kann z.B. genutzt werden, um Rückschlüsse auf mögliche Fehler in der CAN-Verdrahtung zu ziehen.

2. Voraussetzungen

1. Der Aufruf ist nur für PCI-Boards der 331-Serien implementiert:
CAN-PCI/331, PMC-CAN/331, CPCI-CAN/331
2. Die Karte muss sich im 2.0B Mode befinden.
Ist dies nicht der Fall, kann sie mit den folgenden Befehlen in den 2.0B Mode geschaltet werden.

Firmware-Update für die CAN-PCI/331

Files:

updc331.exe
ntcan.dll
updc331 (für Linux)

Karte in den 2.0B Mode schalten (Nur im 2.0B Mode kann der verwendete CAN Controller das gewünschte Errorlogging.)

```
Aufruf "updc331 -tb 0"
```

Treiber neu starten oder Rechner booten.

```
"updc331 0" ausführen
```

Nach einem erneuten Neustart ist die neue Firmware aktiv.

3. Anwendung

1. CAN-Baudrate setzen
2. `NTCAN_EV_USER+0x03e` "enablen".
3. Mit **`CanSend()`** eine "EVMSG" der Länge 2 mit der "evid" `NTCAN_EV_USER*0x03e` und der Periode in Millisekunden im Parameter `s[0]` an das Board senden.

**HINWEIS**

User-Events können nur mit **`CanSend()`** an das CAN-Board gesendet werden (**`canWrite()`**-Aufrufe liefern eine Fehlermeldung zurück, weil diese eine Bestätigung für gesendete CAN-Nachrichten auf dem Bus erwarten, die bei Events nicht erzeugt wird.)

4. Für jedes auf der Karte vorhandene CAN Netz wird mit der gewünschten Frequenz ein CAN EVENT mit der ID `NTCAN_EV_USER*0x03e` gesendet. Im Parameter `s[0]` wird die Anzahl der erkannten CAN-Fehler seit dem Start der Firmware übergeben.

**HINWEIS**

Sind zwei CAN-Netze vorhanden, muss an jedes Netz der Karte ein CAN Events mit der ID `NTCAN_EV_USER*0x03e` gesendet werden. Man erhält für jedes Netz einen getrennten Event: Mit dem Aufruf von **`canRead()`** (unter der Voraussetzung, dass als `len 2` übergeben wurde und unter `msg*` genug Platz für zwei CMSGs (respektive EVMSGs) vorhanden ist), werden sie empfangen.

4. Beispielprogramm

```
#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. Referenzen

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