

# Application Note AN-0143

## CEoX Beispielprogramme

<b>1. ÜBERBLICK</b>	<b>2</b>
<b>2. DIE CEOX TASK „CEOXTASK1.DLL“</b>	<b>2</b>
2.1. Die Funktion „main(void)“	2
2.2. Die Funktion „AutoInit(void)“	2
2.3. Die Funktion „MainProc(void)“	3
<b>3. DAS WINDOWS CE PROGRAMM „CEOX_LOADEXAMPL“</b>	<b>3</b>
<b>4. AUSFÜHREN DER BEISPIELPROGRAMME</b>	<b>3</b>

## 1. Überblick

Die im folgenden beschriebenen Beispielprogramme sollen verdeutlichen, wie die Erstellung eigener Programme unter CEoX oder Windows CE funktioniert. Die Embedded Visual C Projekte zu diesen Programmen können als Vorlage für eigene Projekte verwendet werden.

Es werden in dieser Application Note zwei Programme beschrieben. Das erste Programm ist eine unter CeoX lauffähige Echtzeittask. Das zweite Programm ist ein Windows CE Programm, welches die CEoX Task lädt, die als dll Datei auf dem X-MAX-400 vorliegt.

## 2. Die CEoX Task „ceotask1.dll“

Das Programmbeispiel „ceotask1.dll“ ist eine einfache NI-Task, die eine LED des X-MAX-400 zum Blinken bringt und einen SRQ (Service-Request) über Ethernet an den Host-PC sendet. Das Programm besteht aus drei Funktionen:

- 1) Die Funktion „main(void)“ initialisiert die PDT (Programm-Deskriptor-Tabelle) der Echtzeit Task.
- 2) Die Funktion „AutoInit(void)“ kann beim Start des Programms aufgerufen werden und wird üblicherweise zur Durchführung einmaliger Initialisierungen verwendet.
- 3) Die Funktion MainProc(void) wird vom CEoX Scheduler im Fall einer NI-Task immer dann aufgerufen, wenn Windows CE sich im Idle-Zustand befindet.

### 2.1. Die Funktion „main(void)“

Diese Funktion initialisiert die PDT. Eine detaillierte Beschreibung der einzelnen Parameter kann im Handbuch zu den MAX-PCs gefunden werden (Anhang D).

### 2.2. Die Funktion „AutoInit(void)“

Die Funktion AutoInit() wird beim Start des Programms aufgerufen und führt einmalig die zur Initialisierung notwendigen Schritte aus. Um später auf die on-board LED des X-MAX-400 zugreifen zu können, wird ein Handle auf den X-MAX-400 benötigt. Dieses wird mit der Funktion „max\_connect\_cpu()“ erzeugt:

```
rcParameter.Error = max_connect_cpu(MAX_LOCAL_BOARD,  
MAX_MYSELF,0,&rcParameter.rcOsX, &rcParameter.hMyself);
```

Um SRQs an den Host-PC zu senden, wird ein Handle auf den Host-PC benötigt. Dieses wird ebenfalls mit der Funktion max\_connect\_cpu() erzeugt:

```
rcParameter.Error = max_connect_cpu (MAX_ETHERNET_HOST, 0, 0, NULL,  
&rcParameter.hHost);
```

Die Handles werden in beiden Fällen im Parameterbereich der Task gespeichert, da sie global verfügbar sein müssen.

### 2.3. Die Funktion „MainProc(void)“

Diese Funktion wird vom CEoX Ni-Scheduler aufgerufen. Nach jeweils 100.000 Durchläufen wird die on-board LED auf dem X-MAX-400 um-, also entweder ein- oder ausgeschaltet, und ein SRQ an den Host-PC versendet. Beide Funktionen nutzen die jeweiligen Handles, die in der AutoInit-Funktion erzeugt wurden.

```
rcParameter.Error = max_led_off(rcParameter.hMyself);
```

(Ausschalten der LED unter Verwendung des Handles auf den X-MAX-400)

bzw.

```
rcParameter.Error = max_send_srq(rcParameter.hHost, 0, 0x1234);
```

(Senden einer 16 Bit- Nachricht an den Host PC)

## 3. Das Windows CE Programm „ceox\_loadexampl“

Zum Laden der CEoX-Task wird ein Windows CE Programm verwendet. Das Programm ruft die Funktion max\_install\_task() auf, um das Echtzeitprogramm unter CEoX zu installieren. Alle hierfür benötigten Informationen werden in eine Variable vom Typ MAX\_PGM\_TYPE eingetragen und der Funktion „max\_install\_task()“ übergeben. Es wird vorher wieder per „max\_connect\_cpu()“ eine Verbindung zum X-MAX-400 hergestellt. Durch Setzen der entsprechenden Flags wird die Task automatisch gestartet und die Funktion „AutoInit()“ beim ersten Start aufgerufen. Der Parameter acRtFileName gibt den Pfad zum Echtzeitprogramm auf dem X-MAX-400 sowie dessen Dateinamen an.

Falls ein auf dem Host-PC vorliegendes Programm installiert werden soll, muss die Funktion „max\_transfer\_and\_install()“ verwendet werden.

## 4. Ausführen der Beispielprogramme

Um die Beispielprogramme auszuführen, kann wie folgt vorgegangen werden:

- Starten einer Telnet Sitzung mit Start-> Ausführen -> „telnet 192.168.0.221“ (unter Windows mit der voreingestellten IP-Adresse des X-MAX-400)
- Mit dem Befehl „cd“ in das Verzeichnis wechseln, in dem die Programme liegen. (üblicherweise //onboardflash/sorcus)
- Das Programm ceox\_loadexampl starten.

Zum Überprüfen, ob die Echtzeit-Task korrekt geladen wurde, kann das Programm SNW32 verwendet werden. Dieses kann unter [www.sorcus.de](http://www.sorcus.de) kostenlos heruntergeladen werden. Um mit SNW32 auf einen MAX-PC zugreifen zu können, muss der CEoX Makro Server gestartet sein. Falls dieser nicht beim Starten des MAX-PCs geladen wurde, kann er mit dem Befehl „ceox\_ms“ aus einer Telnet Sitzung heraus geladen werden. SNW32 erkennt außerdem nur angeschlossene Hardware, sofern diese unter „Einstellungen->Systemsteuerung->SORCUS boards“ richtig konfiguriert wurde.

Nachdem das CPU Modul in SNW32 ausgewählt wurde, kann über den Reiter „Tasks“ auf den Parameter- und den Datenbereich einer Echtzeit-Task zugegriffen werden. In der angehängten

Abbildung ist der Parameterbereich der Task zu sehen (Mit der Variable rcParameter.usTask in den Bytes 11 und 12 und der Variablen rcParameter.ulCounter in den Bytes 2 bis 5).

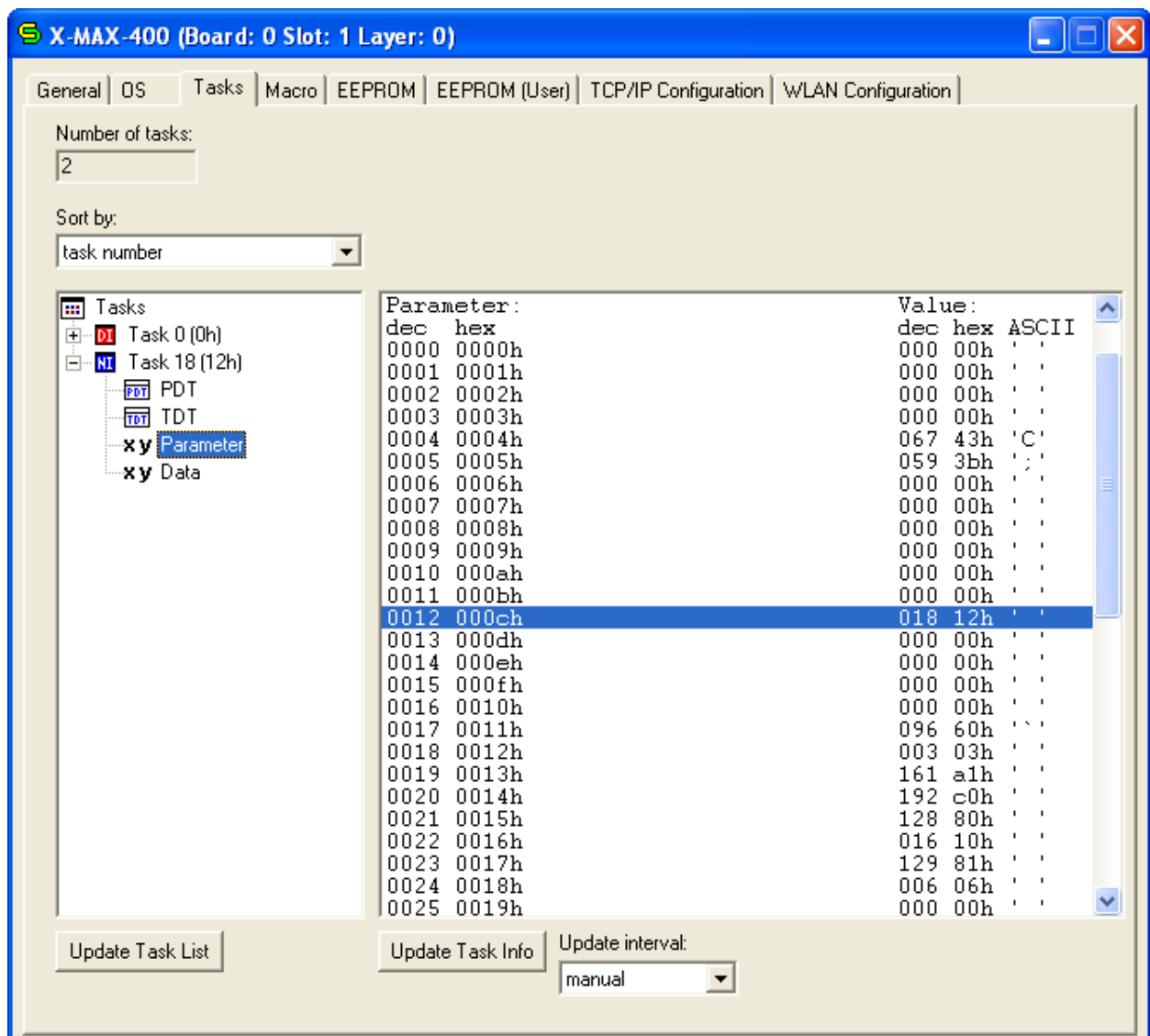


Abb. 1: Parameterbereich des Beispielprogramms