

UE07_trap.c

```

/*****
 * Filename      : UE07_trap.c      *
 * Created on    : Nov 21, 2018    *
 * Author       : Christian Zahner  *
 *****/

#pragma compact_abi

#include "UART0.h"
#include "support_common.h" // include peripheral declarations and more;
#include "uart_support.h"   // universal asynchronous receiver transmitter,
                           // (d.h. die serielle Schnittstelle)
#include "terminal_wrapper.h"
#include "Intro.h"
#include <stdio.h>

#include "UE07_Trap.h"

#define MEMOFFSET 0x20000000

int usrstack[20];

char strtxt[] = "Start Userprogramm\r\n";
char bstxt[] = "Ausgabe via BS-Aufruf via TRAP (wechseln in Systemmode)!!!\r\n";
char endtxt[] = "Ende Programm!!!\r\n";

void trap(){

    int j = 20;
    for(j=0;j<20;j++) usrstack[j]=0xaffe;           // Userstack default init

    asm{

        bra start

////////////////////////////////////

        BS:                                // "Betriebssystem", hier nur Ausgabe eines Texts
                                           // Adresse des Textes wird als Parameter auf
                                           // dem User-Stack erwartet

        move.l a2, -(SP)                    // a2 sichern

        move    USP, a2                     // UserStackPointer nach a2 holen
        move.l (a2), -(SP)                  // Adresse des Strings von UserStack holen
                                           // und auf aktuellen Stack (=SystemStack) speichern
        jsr TERM_WriteString               // String ausgeben
        lea     4(SP), SP                   // Stack cleanup

        adda.l #4, a2                       // USP bereinigen: USP + 4,
                                           // [alternativ: lea 4(a2),a2]
        move    a2, USP                     // veränderten USP in a2 den USP übergeben

        move.l (SP)+, a2                    // a2 restaurieren

        rte                                 // Komisch da TRAP als Exception behandelt wird

////////////////////////////////////

        start:
        lea BS, a1                          // Laden der Adresse des Labels/Marke in a1
        move.l a1, MEMOFFSET+0x80           // Adresse von BS an TRAP Vektor Nr 0 hängen

```

UE07_trap.c

```

lea usrstack[20], a0      // "usrstack" in a0
move.l a0, USP            // USP um gröÙe usrstack versetzen

move.w SR,d2             // Statusregister in Dataregister laden
andi.l #0xDFFF, d2       // Maske mit Statusregister UND Verknüpfen
                          // Um Bit Nr. 13 (S Bit) auf 0 zusetzen
                          // Um Useermode zu aktivieren alle anderen
                          // Bits es Statusregisters bleiben unberührt
                          // DFFF ist 1101 1111 1111 1111
move.w d2,SR             // Verändertes Statusregister von d2 ins
                          // Statusregister kopieren jetzt ist der
                          // User Mode aktiviert

bra usrstrt

```

////////////////////////////////////

usrstrt:

```

pea strtxt               // selbst erklärend
jsr TERM_WriteString
adda.l #4, SP            // Cleanup Stack

pea bstxt                // hier wird, da im Usermode sind mittles
                          // USP auf den Stack geschrieben
                          // wichtig später brauchen wir
                          // den USP (Userstackpointer) um den
                          // String zu bekommen

TRAP #0

pea endtxt               // Was passiert hier?
                          // Ein Zauberer verrät nie seine Tricks!
jsr TERM_WriteString
add.l #4, SP            // Cleanup Stack

```

}

}