

```

1  /*=====*/
/* Projekt      :   Olli - eine Holzkiste lernt laufen          */
/*                                                     */
/* Hardware     :   Infineon C509                            */
5 /*                                                     */
/* Dateiname    :   intrrpt_init.c                          */
/*                                                     */
/* Version      :   1.1 vom 05. Oktober 2003                */
/*                                                     */
10 /* Autoren    :   L. Kulf, J. Roos                        */
/*                                                     */
/* Datei-       :                                           */
/* beschreibung :   Interruptinitialisierung                */
/*                                                     */
15 /*                                                     */
/*                                                     */
/*=====*/

20 // -----
// Dateien einbinden
#include "intrrpt_init.h"

25 // Interrupts -----

// Interruptinitialisierung
void interrupt_init (void)
{
30     // Interruptnr. 1 (Port 3.4, Countereingang Achse 1-3)
    // Überlauf Counter Geberinkremente
    ET0=1;

    // Interrupt-Request-Bit ist TF0 und wird von Timer0 gesetzt
35     TF0=0;

    // -----
    // Interruptnr. 3 (C), Port 3.5, Countereingang Achse 4
    ET1=1;
40     TF1=0;

    // -----
    // Interrupt 4, Interruptnr. 11, Port 1.1, Endlage einer Achse angefahren
    IEX4=0;
45     EX4=1;

    // -----
    // Interrupt 5, Interruptnr. 12, Port 1.2, MotorsteuerIC - Overload/Überlast
    IEX5=0;
50     EX5=1;

    // -----
    // Serielle Schnittstelle 0
    TI0 = 0;
55     // Das Transmit Interrupt Request Flag wird gelöscht.
    // (TI0 wird nach erfolgter Sendung über S0 gesetzt).
    // TI0 wird automatisch gesetzt muß aber manuell zurückgesetzt werden.
    RI0 = 0;
    // Das Receive Interrupt Request Flag wird gelöscht.
60     // (RI0 wird nach einem Empfang über S0 gesetzt).
    // RI0 wird automatisch gesetzt muß aber manuell zurückgesetzt werden.
    ES0 = 1;
    // Interrupt Enable fuer S0 wird eingeschaltet (altern: SET_ES0)
    // Prioritätstufe sollte 2 sein (zweit Höchste)

65     // -----
    // Serielle Schnittstelle 1
    S1CON = (S1CON & 0xFC);
    // Interrupt Request Flags der S1 werden gelöscht
70     // Maskierung von Bit 0 und Bit 1 (auf "0" setzten)
    // bit 0 in S1CON ist: (RI1): Rx(receiver)Interrupt Request Flag
    // bit 1 in S1CON ist: (TI1): Tx(transmit)Interrupt Request Flag
    IEN2 = (IEN2 | 0x01);
    // Interrupt Enable fuer S1 wird eingeschaltet
75     // Im SFR IEN2 ist das Bit0 ES1.
    // Prioritätstufe sollte 2 sein (zweit Höchste)
    // Prioritäts Stufen Einstellungen
    // IP generelle Einstellungen      543210b

80     // -----
    // Prioritäten einstellen
    IP0=0x0A;    // 00001010
    IP1=0x9B;    // 10011011

85     // -----
    // Intrptnr.   Funktion                Priorität   Bit in IP0/IP1
    // 16          serielle Schnittstelle      2          IP1.0/IP0.0
    // 1           Impulszählung Geber          3          IP1.1/IP0.1
    // 3           Impulszählung Geber          3          IP1.3/IP0.3

```

```

90    // 11          Endlagenüberwachung          3          -----"-----
    // 4            serielle Schnittstelle        2          IP1.4/IP0.4
    // 12           Motoren Überlast              2          -----"-----
    // 13           Deckelüberwachung              0          IP1.5/IP0.5
    // -----
95    // um P9 einlesen zu können muss IP1.7 mit 1 beschrieben werden

    // alle Interrupts enable
    EAL=1;
}
100 // -----
    // Dateiende

```