

```

1  /*=====*/
/* Projekt      :   Olli - eine Holzkiste lernt laufen          */
/*                                                     */
/* Hardware     :   Infineon C509                            */
5  /*                                                     */
/* Dateiname    :   umdrehungszaehler.c                      */
/*                                                     */
/* Version      :   1.2 vom 04. Oktober 2003                 */
/*                                                     */
10 /* Autoren     :   L. Kulf, J. Roos                        */
/*                                                     */
/* Datei-       :                                           */
/* beschreibung :   Umdrehungszählerinitialisierung (Motor, Spindel),
/*                  Umdrehungszähleraktualisierung Achsen 1-4
15 /*                                                     */
/*                                                     */
/*=====*/

20 // -----
// Dateien einbinden
#include "umdrehungszaehler.h"
#include "main.h"
#include "counter_init.h"
25 #include "mot_outp.h"

// -----
// Variablendeklaration, globale Variablen
PMBYTE m_umdr_a1 = 0;          // Umdrehungszähler Motor Achse 1
30 PMBYTE s_umdr_a1 = 0;        // Umdrehungszähler Spindel Achse 1
PMBYTE m_umdr_hilfsm_a1 = 0;    // Umdrehungszähler Motor Hilfsmerker Achse 1
PMBYTE m_umdr_a2 = 0;          // Umdrehungszähler Motor Achse 2
PMBYTE s_umdr_a2 = 0;          // Umdrehungszähler Spindel Achse 2
PMBYTE m_umdr_hilfsm_a2 = 0;    // Umdrehungszähler Motor Hilfsmerker Achse 2
35 PMBYTE m_umdr_a3 = 0;        // Umdrehungszähler Motor Achse 3
PMBYTE s_umdr_a3 = 0;          // Umdrehungszähler Spindel Achse 3
PMBYTE m_umdr_hilfsm_a3 = 0;    // Umdrehungszähler Motor Hilfsmerker Achse 3
PMBYTE m_umdr_a4 = 0;          // Umdrehungszähler Motor Achse 4
PMBYTE s_umdr_a4 = 0;          // Umdrehungszähler Spindel Achse 4
40 PMBYTE m_umdr_hilfsm_a4 = 0; // Umdrehungszähler Motor Hilfsmerker Achse 4

// Umdrehungszähler -----
45 // -----
// Umdrehungszähler Achse 1 initialisieren
void umdr_zaeher_init_a1 (void)
{
50     m_umdr_a1 = 0;
    m_umdr_hilfsm_a1 = 0;
    s_umdr_a1 = 0;

    //TL0 = 0xF0; nur diese Zeile statt "counter_init()" geht nicht !!
55     // Counterinitialisierung, um bei Drehrichtungsumkehr durch nicht voll-
    // ständige Umdrehungen keine Lageabweichungen zu bekommen.
    counter_init();
}

60 // -----
// Umdrehungszähler Achse 2 initialisieren
void umdr_zaeher_init_a2 (void)
{
65     m_umdr_a2 = 0;
    m_umdr_hilfsm_a2 = 0;
    s_umdr_a2 = 0;

    //TL0 = 0xF0; nur diese Zeile statt "counter_init()" geht nicht !!
70     // Counterinitialisierung, um bei Drehrichtungsumkehr durch nicht voll-
    // ständige Umdrehungen keine Lageabweichungen zu bekommen.
    counter_init();
}

75 // -----
// Umdrehungszähler Achse 3 initialisieren
void umdr_zaeher_init_a3 (void)
{
80     m_umdr_a3 = 0;
    m_umdr_hilfsm_a3 = 0;
    s_umdr_a3 = 0;

    //TL0 = 0xF0; nur diese Zeile statt "counter_init()" geht nicht !!
85     // Counterinitialisierung, um bei Drehrichtungsumkehr durch nicht voll-
    // ständige Umdrehungen keine Lageabweichungen zu bekommen.
    counter_init();
}

// -----
// Umdrehungszähler Achse 4 initialisieren

```

```

90 void umdr_zaeher_init_a4 (void)
{
    m_umdr_a4 = 0;
    m_umdr_hilfsm_a4 = 0;
    s_umdr_a4 = 0;
95
    //TL0 = 0xF0; nur diese Zeile statt "counter_init()" geht nicht !!
    // Counterinitialisierung, um bei Drehrichtungsumkehr durch nicht voll-
    // ständige Umdrehungen keine Lageabweichungen zu bekommen.
    counter_init();
100 }

// -----
// Umdrehungszähler Achse 1 aktualisieren
// wird aufgerufen durch Interrupt 1 aus Überlauf Counter 0
105 void count_umdr_a1 (void)
{
    // Umdrehungszähler Motor inkrementieren
    // jeden 16. Impuls wird Umdrehungszähler Motor akt.
    m_umdr_a1++;
110 m_umdr_hilfsm_a1++;

    // Umdrehungszähler Spindel inkrementieren
    // alle GEAR_N Motorumdrehungen wird Umdrehungszähler Spindel aktualisiert.
    // Hilfsmerker Umdrehungszähler Motor wird gelöscht.
115 if (m_umdr_hilfsm_a1 == GEAR_N_23 )
    {
        s_umdr_a1++;
        m_umdr_hilfsm_a1=0;
    }
120 }

// -----
// Umdrehungszähler Achse 2 aktualisieren
// wird aufgerufen durch Interrupt 1 aus Überlauf Counter 0
125 void count_umdr_a2 (void)
{
    // Umdrehungszähler Motor inkrementieren
    // jeden 16. Impuls wird Umdrehungszähler Motor akt.
    m_umdr_a2++;
130 m_umdr_hilfsm_a2++;

    // Umdrehungszähler Spindel inkrementieren
    // alle GEAR_N Motorumdrehungen wird Umdrehungszähler Spindel aktualisiert.
    // Hilfsmerker Umdrehungszähler Motor wird gelöscht.
135 if (m_umdr_hilfsm_a2 == GEAR_N_23 )
    {
        s_umdr_a2++;
        m_umdr_hilfsm_a2=0;
    }
140 }

// -----
// Umdrehungszähler Achse 3 aktualisieren
// wird aufgerufen durch Interrupt 1 aus Überlauf Counter 0
145 void count_umdr_a3 (void)
{
    // Umdrehungszähler Motor inkrementieren
    // jeden 16. Impuls wird Umdrehungszähler Motor akt.
    m_umdr_a3++;
150 m_umdr_hilfsm_a3++;

    // Umdrehungszähler Spindel inkrementieren
    // alle GEAR_N Motorumdrehungen wird Umdrehungszähler Spindel aktualisiert.
    // Hilfsmerker Umdrehungszähler Motor wird gelöscht.
155 if (m_umdr_hilfsm_a3 == GEAR_N_23 )
    {
        s_umdr_a3++;
        m_umdr_hilfsm_a3=0;
    }
160 }

// -----
// Umdrehungszähler Achse 4 aktualisieren
// wird aufgerufen durch Interrupt 3 aus Überlauf Counter 1
165 void count_umdr_a4 (void)
{
    // Umdrehungszähler Motor inkrementieren
    // jeden 16. Impuls wird Umdrehungszähler Motor akt.
    m_umdr_a4++;
170 m_umdr_hilfsm_a4++;

    // Umdrehungszähler Spindel inkrementieren
    // alle GEAR_N Motorumdrehungen wird Umdrehungszähler Spindel aktualisiert.
    // Hilfsmerker Umdrehungszähler Motor wird gelöscht.
175 if (m_umdr_hilfsm_a4 == GEAR_N_23 )
    {
        s_umdr_a4++;
        m_umdr_hilfsm_a4=0;
    }

```

```
} / Dateiende
```