

```

1  /*=====*/
/* Projekt      :   Olli - eine Holzkiste lernt laufen          */
/*                                                     */
/* Hardware      :   Infineon C509                            */
5  /*                                                     */
/* Dateiname     :   ba_auto.c                                */
/*                                                     */
/* Version       :   1.3 vom 17. März 2004                    */
/*                                                     */
10 /* Autoren      :   L. Kulf, J. Roos                         */
/*                                                     */
/* Datei-        :                                           */
/* beschreibung  :   BA Automatik, Ablauf                      */
/*                   BA Teilautomatik, wenn von Olli-Control angefordert */
15 /*                                                     */
/*                                                     */
/*=====*/

20 // -----
// Dateien einbinden
#include "main.h"
#include "ba_auto.h"
#include "move_order.h"
25 #include "mot_outp.h"
#include "umdrehungszaeher.h"
#include "ba_teilauto.h"
#include "error_l.h"
#include "seriell_olli_seriell.h"
30

// -----
// Variablendeklaration, globale Variablen
_bit p1_fahrt_aktiv=0;          // Fahrt auf Position 1 ist aktiv
35 _bit p2_fahrt_aktiv=0;          // Fahrt auf Position 2 ist aktiv
_bit p3_fahrt_aktiv=0;          // Fahrt auf Position 3 ist aktiv
_bit p4_fahrt_aktiv=0;          // Fahrt auf Position 4 ist aktiv
_bit p5_fahrt_aktiv=0;          // Fahrt auf Position 5 ist aktiv
_bit p6_fahrt_aktiv=0;          // Grundstellungsfahrt ist aktiv
40 _bit letzte_mess_pos=0;         // letzte Messposition wurde angefahren
_bit ba_ta_p6_fahrt=0;          // BA TA P6-Fahrt wurde angefordert

extern unsigned char move_order_pos; // anzufahrende Position
extern unsigned char refp[4];        // Referenzpunktschalter
45 extern _bit m_mot_a1_plus;
extern _bit m_mot_a1_minus;
extern _bit m_mot_a2_plus;
extern _bit m_mot_a2_minus;
extern _bit m_mot_a3_plus;
50 extern _bit m_mot_a3_minus;
extern _bit m_mot_a4_plus;
extern _bit m_mot_a4_minus;
extern _bit pos_erreicht_p1;
extern _bit pos_erreicht_p2;
55 extern _bit pos_erreicht_p3;
extern _bit pos_erreicht_p4;
extern _bit pos_erreicht_p5;
extern _bit pos_erreicht_p6;
extern signed char s_umdr_a1;
60 extern signed char s_umdr_a2;
extern signed char s_umdr_a3;
extern signed char s_umdr_a4;
extern _bit m_ba_teilauto;          // Betriebsart Teilautomatik angewählt
extern _bit m_ba_auto;              // Betriebsart Automatik angewählt
65

// -----
// Ablauf Betriebsart Automatik
void ba_auto (void)
70 {
    // -----
    // Grundstellung anfahren
    if ((!p1_fahrt_aktiv)
        &&(!p2_fahrt_aktiv)
75         &&(!p3_fahrt_aktiv)
        &&(!p4_fahrt_aktiv)
        &&(!p5_fahrt_aktiv)
        &&(move_order_pos>0)
        &&(move_order_pos<=6))
80     {
        if(refp[0] == 0)
        {
            if(m_mot_a1_minus == 0)
            {
85                 mot_a1_minus();
                p6_fahrt_aktiv=1;
            }
        }
    }
}

```

```

90     if(refp[1] == 0)
    {
        if(m_mot_a2_minus == 0)
        {
            mot_a2_minus();
95         p6_fahrt_aktiv=1;
        }
    }

    if(refp[2] == 0)
100   {
        if(m_mot_a3_minus == 0)
        {
            mot_a3_minus();
            p6_fahrt_aktiv=1;
105         }
    }

    if(refp[3] == 0)
    {
110         if(m_mot_a4_minus == 0)
        {
            mot_a4_minus();
            p6_fahrt_aktiv=1;
        }
115     }

    if(refp[0] == 1)
    {
120         mot_a1_stp();
        umdr_zaebler_init_a1();
    }

    if(refp[1] == 1)
125   {
        mot_a2_stp();
        umdr_zaebler_init_a2();
    }

    if(refp[2] == 1)
130   {
        mot_a3_stp();
        umdr_zaebler_init_a3();
        pos_erreicht_p4=0;
135     }

    if(refp[3] == 1)
    {
        mot_a4_stp();
140         umdr_zaebler_init_a4();
        pos_erreicht_p1=0;
        pos_erreicht_p2=0;
        pos_erreicht_p3=0;
        pos_erreicht_p5=0;

145         if(letzte_mess_pos == 1)
        {
            letzte_mess_pos=0;
            transmit_S0_vorbereitung(6);
150         }
    }

    if ((refp[0] == 1)
155         &&(refp[1] == 1)
        &&(refp[2] == 1)
        &&(refp[3] == 1))
    {
        p6_fahrt_aktiv=0;
160     }

    // Hilfsflag bilden, P6-Fahrt in Teilauto angefordert
    if ((move_order_pos == 6) && (p6_fahrt_aktiv == 1))
    {
165         ba_ta_p6_fahrt=1;
    }

    // Bestätigung an Olli-Control, wenn P6-Fahrt zu Ende ist (BA Teilauto)
    if((ba_ta_p6_fahrt == 1) && (p6_fahrt_aktiv == 0))
    {
170         transmit_S0_vorbereitung(6);
        move_order_pos=0;
        ba_ta_p6_fahrt=0;
    }
}

175 // Position 1: Olli Mitte, Achse 4 Mitte
if((move_order_pos == 1) && (!p6_fahrt_aktiv))
{

```

```

180     if(pos_erreicht_p1 == 0)
    {
        if ((!m_mot_a4_plus)
            &&(!m_mot_a4_minus)
            &&(!m_mot_a1_minus)
            &&(!m_mot_a2_minus)
185         &&(!m_mot_a3_minus))
        {
            p1_fahrt_aktiv=1;

            // Motor A4 Richtung+ ein
190         mot_a4_plus();

            // Meldung LCD ausgeben
            ba_a_msg_p1_aktiv();
        }

195         // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
        if (s_umdr_a4 == S_U_SOLL_A4_M)
        {
            mot_a4_stp();
            p1_fahrt_aktiv=0;
            move_order_pos=0;
            transmit_S0_vorbereitung(1);

            // LCD-Meldung aufrufen
200         ba_a_msg_p1();
        }

        if(m_mot_a4_plus == 0)
        {
210         pos_erreicht_p1=1;
        }
    }
}

215 // Position 2: Achse 1 in R+, Achse 4 Mitte
if((move_order_pos == 2) && (!p6_fahrt_aktiv))
{
    if(pos_erreicht_p2 == 0)
220     {
        if (((!m_mot_a4_plus)
            &&(!m_mot_a4_minus))
            &&(!m_mot_a1_plus)
            &&(!m_mot_a1_minus)))
225         {
            p2_fahrt_aktiv=1;

            // Motor A4, A1 Richtung+ ein
            mot_a4_plus();
            mot_a1_plus();

230             // Meldung LCD ausgeben
            ba_a_msg_p2_aktiv();
        }

235         // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
        if (s_umdr_a4 == S_U_SOLL_A4_M)
        {
            mot_a4_stp();
240         }

        if (s_umdr_a1 == S_U_SOLL_A1)
        {
            mot_a1_stp();
245             // Achtung !
            // folgende 2 Zeilen müssen eventuell nach mot_a4_stp();
            // an Statt hier eingefügt werden.
            // Die Achse, die später am Ziel ist, muß diese flags auf
            // false setzen.
            p2_fahrt_aktiv=0;
            move_order_pos=0;
            transmit_S0_vorbereitung(2);

250             // LCD-Meldung aufrufen
            ba_a_msg_p2();
        }

        if((m_mot_a4_plus == 0) && (m_mot_a1_plus == 0))
260         {
            pos_erreicht_p2=1;
        }
    }
}

265 // Position 3: Achse 3 in R+, Achse 4 Mitte
if((move_order_pos == 3) && (!p6_fahrt_aktiv))

```

```

{
    if(pos_erreicht_p3 == 0)
    {
        if (((!m_mot_a4_plus)
            &&(!m_mot_a4_minus))
            &&(!m_mot_a3_plus)
            &&(!m_mot_a3_minus)))
        {
            p3_fahrt_aktiv=1;

            // Motor A4 Richtung+ ein
            mot_a4_plus();
            mot_a3_plus();

            // Meldung LCD ausgeben
            ba_a_msg_p3_aktiv();
        }

        // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
        if (s_umdr_a4 == S_U_SOLL_A4_M)
        {
            mot_a4_stp();
        }

        if (s_umdr_a3 == S_U_SOLL_A3)
        {
            mot_a3_stp();

            // Achtung !
            // folgende 2 Zeilen müssen eventuell nach mot_a4_stp();
            // an Statt hier eingefügt werden.
            // Die Achse, die später am Ziel ist, muß diese flags auf
            // false setzen.
            p3_fahrt_aktiv=0;
            move_order_pos=0;
            transmit_S0_vorbereitung(3);

            // LCD-Meldung aufrufen
            ba_a_msg_p3();
        }

        if((m_mot_a4_plus == 0) && (m_mot_a3_plus == 0))
        {
            pos_erreicht_p3=1;
        }
    }
}

// Position 4: Achse 2 in R+
if((move_order_pos == 4) && (!p6_fahrt_aktiv))
{
    if(pos_erreicht_p4 == 0)
    {
        if((!m_mot_a2_plus) && (!m_mot_a2_minus))
        {
            p4_fahrt_aktiv=1;

            // Motor A2 Richtung+ ein
            mot_a2_plus();

            // Meldung LCD ausgeben
            ba_a_msg_p4_aktiv();
        }

        // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
        if (s_umdr_a2 == S_U_SOLL_A2)
        {
            mot_a2_stp();
            p4_fahrt_aktiv=0;
            move_order_pos=0;
            transmit_S0_vorbereitung(4);

            // LCD-Meldung aufrufen
            ba_a_msg_p4();
        }

        if(m_mot_a2_plus == 0)
        {
            pos_erreicht_p4=1;
        }
    }
}

// Position 5: Achse 4 in R+ bis oben
if((move_order_pos == 5) && (!p6_fahrt_aktiv))
{
    if(pos_erreicht_p5 == 0)
    {
        if((!m_mot_a4_plus) && (!m_mot_a4_minus))

```

```

    {
        p5_fahrt_aktiv=1;
360        // Motor A4 Richtung+ ein
        mot_a4_plus();

        // Meldung LCD ausgeben
365        ba_a_msg_p5_aktiv();
    }

    // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
    if (s_umdr_a4 == S_U_SOLL_A4_O)
    {
370        mot_a4_stp();
        p5_fahrt_aktiv=0;
        move_order_pos=0;
        transmit_S0_vorbereitung(5);

375        // LCD-Meldung aufrufen
        ba_a_msg_p5();
    }

    if(m_mot_a4_plus == 0)
380    {
        pos_erreicht_p5=1;

        // flag für letzte Messposition erreicht setzen
385        letzte_mess_pos=1;
    }
}
}

390 // -----
    // Dateiende

```