

```

1  /*=====*/
/* Projekt      :   Olly - eine Holzkiste lernt laufen      */
/*              :   */
/* Hardware     :   Infineon C509                          */
5  /*              :   */
/* Dateiname    :   Kommunikation_Auswertung_1.c           */
/*              :   */
/* Version      :   1.2 vom 07. Februar 2004               */
/*              :   */
10 /* Autoren     :   L. Kulf, J. Roos                      */
/*              :   */
/* Datei-       :   */
/* beschreibung :   Befehlspeicher der seriellen Schnittstelle auslesen.
/*              :   Hier werden die Befehle, die vom LCD-Display kommen,
15 /*              :   ausgewertet und für weitere Verwendung aufbereitet.
/*              :   */
/*              :   */
/*=====*/

20 // -----
// Dateien einbinden
#include "Kommunikation_Auswertung_1.h"
#include "achse_freifahren.h"
#include "error_1.h"
25

// -----
// Variablendeklaration, globale Variablen
_bit m_ba_auto=1; // Betriebsart Automatik angewählt
30 _bit m_ba_teilauto=0; // Betriebsart TeilAutomatik angew.
_bit m_ba_hand=0; // Betriebsart Hand angewählt
_bit err_qtt=0; // Störung quittieren von LCD angef.
_bit achse_freifahren=0; // Achse freifahren erforderlich

35 unsigned char ba_h_fahrbef=0; // aktiver Fahrbefehl:
// 1 = A1 +
// 2 = A1 -
// 3 = A2 +
// 4 = A2 -
40 // 5 = A3 +
// 6 = A3 -
// 7 = A4 +
// 8 = A4 -

45 extern unsigned char neu_anzufahrende_pos; // Teilautomatik : aktiver Fahrbefehl
extern _bit m_new_lcd_order; // neuer Befehl vom LCD vorhanden

extern _bit pos_erreicht_tipp_a1_plus; // Tippen : Position A1 ist erreicht
extern _bit pos_erreicht_tipp_a1_minus; // Tippen : Position A1 ist erreicht
50 extern _bit pos_erreicht_tipp_a2_plus; // Tippen : Position A2 ist erreicht
extern _bit pos_erreicht_tipp_a2_minus; // Tippen : Position A2 ist erreicht
extern _bit pos_erreicht_tipp_a3_plus; // Tippen : Position A3 ist erreicht
extern _bit pos_erreicht_tipp_a3_minus; // Tippen : Position A3 ist erreicht
extern _bit pos_erreicht_tipp_a4_plus; // Tippen : Position A4 ist erreicht
55 extern _bit pos_erreicht_tipp_a4_minus; // Tippen : Position A4 ist erreicht

extern _bit m_grdstllg_nio;
extern unsigned char refp[4];

60

// -----
// neuen Befehl vom LCD auslesen
void befehl_auslesen (void)
65 {
    extern unsigned char befehl_sl_in[4];

//-----
70 // BA auslesen und umschalten

    // BA Automatik setzen
    if((befehl_sl_in[2] == 0) && (befehl_sl_in[3] == 1))
    {
75        // nur bei Grundstellung umschalten auf BA Auto zulassen, sonst Fehler-
        // bit setzen
        if ((refp[0] == 1)
            &&(refp[1] == 1)
            &&(refp[2] == 1)
80            &&(refp[3] == 1))
        {
            // Wenn am Display Automatik angefordert wird, Freigabe an Olly-Control
            // senden
            if(m_ba_auto == 0)
85            {
                olly_control_enable();
            }

            m_ba_hand=0;

```

```

90         m_ba_teilauto=0;
           m_ba_auto=1;
       }
       else
       {
95           // Fehlerbit "keine Grundstellung" setzen
           m_grdstllg_nio=1;
       }
   }

100   // BA Teilautomatik setzen
   if((befehl_sl_in[2] == 0) && (befehl_sl_in[3] == 2))
   {
       // nur bei Grundstellung umschalten auf BA TeilAuto zulassen, sonst
       // Fehlerbit setzen
105       if ((refp[0] == 1)
           &&(refp[1] == 1)
           &&(refp[2] == 1)
           &&(refp[3] == 1))
       {
110           // Bei Bedienung über das LCD-Display Olli_Control Freigabe entziehen
           if(m_ba_teilauto == 0)
           {
               olli_control_disable();
           }

115           m_ba_hand=0;
           m_ba_auto=0;
           m_ba_teilauto=1;
       }
       else
       {
           // Fehlerbit "keine Grundstellung" setzen
           m_grdstllg_nio=1;
       }
125   }

   // BA Hand/Tippbetrieb setzen
   if((befehl_sl_in[2] == 0) && (befehl_sl_in[3] == 3))
   {
130       // Umschalten auf BA Hand nicht zulassen, wenn Achsen gerade in BA
       // TeilAuto oder Automatik auf Position fahren
       if (P4 == 0x00)
       {
           // Bei Bedienung über das LCD-Display Olli_Control Freigabe entziehen
135           if(m_ba_hand == 0)
           {
               olli_control_disable();
           }

140           m_ba_teilauto=0;
           m_ba_auto=0;
           m_ba_hand=1;
       }
       else
145       {
           // Meldung auf LCD bilden: Achse aktiv
           msg_ax_aktiv();
       }
   }

150   //-----
   // Teilautomatik Fahrbefehl auslesen
   // neuen Fahrbefehl nur akzeptieren, wenn der letzte nicht mehr aktiv ist
155   if(neu_anzufahrende_pos == 0)
   {
       if((befehl_sl_in[2] == 0) && (befehl_sl_in[3] == 6))
       {
160           neu_anzufahrende_pos=1;
       }

       if((befehl_sl_in[2] == 0) && (befehl_sl_in[3] == 7))
       {
165           neu_anzufahrende_pos=2;
       }

       if((befehl_sl_in[2] == 0) && (befehl_sl_in[3] == 8))
       {
170           neu_anzufahrende_pos=3;
       }

       if((befehl_sl_in[2] == 0) && (befehl_sl_in[3] == 9))
       {
175           neu_anzufahrende_pos=4;
       }

       if((befehl_sl_in[2] == 1) && (befehl_sl_in[3] == 0))
       {

```

```

neu_anzufahrende_pos=5;
180     }

    if((befehl_sl_in[2] == 1) && (befehl_sl_in[3] == 1))
    {
        // Grundstellung nur anfordern, wenn mindestens eine Achse nicht
185        // auf Referenzpunktschalter steht
        if (!refp[0])
            ||(!refp[1])
            ||(!refp[2])
            ||(!refp[3]))
190        {
            // Position 6 entspricht Position Grundstellung
            neu_anzufahrende_pos=6;
            m_ba_hand=0;
            m_ba_auto=0;
195            m_ba_teilauto=1;
        }
        else
        {
            error_grdstllg_ok();
200        }
    }
}

//-----
// BA Hand
205
    if((befehl_sl_in[2] == 1) && (befehl_sl_in[3] == 6))
    {
        if(m_ba_hand == 1)
        {
210            ba_h_fahrbef=1;
            pos_erreicht_tipp_a1_plus=0;
        }
    }

215    if((befehl_sl_in[2] == 1) && (befehl_sl_in[3] == 7))
    {
        if(m_ba_hand == 1)
        {
220            ba_h_fahrbef=2;
            pos_erreicht_tipp_a1_minus=0;
        }
    }

225    if((befehl_sl_in[2] == 1) && (befehl_sl_in[3] == 8))
    {
        if(m_ba_hand == 1)
        {
230            ba_h_fahrbef=3;
            pos_erreicht_tipp_a2_plus=0;
        }
    }

    if((befehl_sl_in[2] == 1) && (befehl_sl_in[3] == 9))
    {
235        if(m_ba_hand == 1)
        {
            ba_h_fahrbef=4;
            pos_erreicht_tipp_a2_minus=0;
        }
    }
240

    if((befehl_sl_in[2] == 2) && (befehl_sl_in[3] == 0))
    {
        if(m_ba_hand == 1)
        {
245            ba_h_fahrbef=5;
            pos_erreicht_tipp_a3_plus=0;
        }
    }

250    if((befehl_sl_in[2] == 2) && (befehl_sl_in[3] == 1))
    {
        if(m_ba_hand == 1)
        {
255            ba_h_fahrbef=6;
            pos_erreicht_tipp_a3_minus=0;
        }
    }

260    if((befehl_sl_in[2] == 2) && (befehl_sl_in[3] == 2))
    {
        if(m_ba_hand == 1)
        {
265            ba_h_fahrbef=7;
            pos_erreicht_tipp_a4_plus=0;
        }
    }
}

```

```
270     if((befehl_sl_in[2] == 2) && (befehl_sl_in[3] == 3))
    {
        if(m_ba_hand == 1)
        {
            ba_h_fahrbef=8;
            pos_erreicht_tipp_a4_minus=0;
275     }
    }
```

```
280 //-----
// sonstige Befehle auslesen
```

```
    // nach Endlagenfahrt über LCD Achse freifahren
    if((befehl_sl_in[2] == 2) && (befehl_sl_in[3] == 5))
    {
285         achse_freifahren=1;
    }
```

```
    // Befehl "Störung quittieren" auslesen
    if((befehl_sl_in[2] == 2) && (befehl_sl_in[3] == 4))
    {
        err_qtt=1;
    }
295 }
```

```
// -----
// Dateiende
```