

```

1  /*=====
/* Projekt      : Olli - 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 : Befehlspuffer 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

// -----
// Variablen Deklaration, globale Variablen
30 _bit m_ba_auto=1;                                // Betriebsart Automatik angewählt
_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 -
// 5 = A3 +
// 6 = A3 -
// 7 = A4 +
// 8 = A4 -
40

45 extern unsigned char neu_anzufahrende_pos;    // Teilaautomatik : 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_s1_in[4];

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

    // BA Automatik setzen
    if((befehl_s1_in[2] == 0) && (befehl_s1_in[3] == 1))
    {
        // nur bei Grundstellung umschalten auf BA Auto zulassen, sonst Fehler-
        // bit setzen
        if ((refp[0] == 1)
            && (refp[1] == 1)
            && (refp[2] == 1)
            && (refp[3] == 1))
        {
            // Wenn am Display Automatik angefordert wird, Freigabe an Olli-Control
            // senden
            if(m_ba_auto == 0)
            {
                olli_control_enable();
            }
        }
        m_ba_hand=0;
    }
}

```

```

90             m_ba_teilauto=0;
91             m_ba_auto=1;
92         }
93     }
94
95     // Fehlerbit "keine Grundstellung" setzen
96     m_grdstllg_nio=1;
97 }
98
99 // BA Teilaautomatik setzen
100 if((befehl_s1_in[2] == 0) && (befehl_s1_in[3] == 2))
101 {
102     // nur bei Grundstellung umschalten auf BA Teilauto zulassen, sonst
103     // Fehlerbit setzen
104     if((refp[0] == 1)
105        &&(refp[1] == 1)
106        &&(refp[2] == 1)
107        &&(refp[3] == 1))
108     {
109         // Bei Bedienung über das LCD-Display Olli_Control Freigabe entziehen
110         if(m_ba_teilauto == 0)
111         {
112             olli_control_disable();
113         }
114
115         m_ba_hand=0;
116         m_ba_auto=0;
117         m_ba_teilauto=1;
118     }
119     else
120     {
121         // Fehlerbit "keine Grundstellung" setzen
122         m_grdstllg_nio=1;
123     }
124 }
125
126 // BA Hand/Tippbetrieb setzen
127 if((befehl_s1_in[2] == 0) && (befehl_s1_in[3] == 3))
128 {
129     // Umschalten auf BA Hand nicht zulassen, wenn Achsen gerade in BA
130     // Teilauto oder Automatik auf Position fahren
131     if (P4 == 0x00)
132     {
133         // Bei Bedienung über das LCD-Display Olli_Control Freigabe entziehen
134         if(m_ba_hand == 0)
135         {
136             olli_control_disable();
137         }
138
139         m_ba_teilauto=0;
140         m_ba_auto=0;
141         m_ba_hand=1;
142     }
143     else
144     {
145         // Meldung auf LCD bilden: Achse aktiv
146         msg_ax_activ();
147     }
148 }
149
150
151 //-----
152 // Teilaautomatik Fahrbefehl auslesen
153 // neuen Fahrbefehl nur akzeptieren, wenn der letzte nicht mehr aktiv ist
154 if(neu_anzufahrende_pos == 0)
155 {
156     if((befehl_s1_in[2] == 0) && (befehl_s1_in[3] == 6))
157     {
158         neu_anzufahrende_pos=1;
159     }
160
161     if((befehl_s1_in[2] == 0) && (befehl_s1_in[3] == 7))
162     {
163         neu_anzufahrende_pos=2;
164     }
165
166     if((befehl_s1_in[2] == 0) && (befehl_s1_in[3] == 8))
167     {
168         neu_anzufahrende_pos=3;
169     }
170
171     if((befehl_s1_in[2] == 0) && (befehl_s1_in[3] == 9))
172     {
173         neu_anzufahrende_pos=4;
174     }
175
176     if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 0))
177     {

```

```

180         neu_anzufahrende_pos=5;
    }

    if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 1))
    {
        // Grundstellung nur anfordern, wenn mindestens eine Achse nicht
        // auf Referenzpunktschalter steht
        if (!refp[0])
            || (!refp[1])
            || (!refp[2])
            || (!refp[3]))
        {
            // Position 6 entspricht Position Grundstellung
            neu_anzufahrende_pos=6;
            m_ba_hand=0;
            m_ba_auto=0;
            m_ba_teilauto=1;
        }
        else
        {
            error_grdstllg_ok();
        }
    }
}

// BA Hand
205
if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 6))
{
    if(m_ba_hand == 1)
    {
        ba_h_fahrbef=1;
        pos_erreicht_tipp_a1_plus=0;
    }
}

215
if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 7))
{
    if(m_ba_hand == 1)
    {
        ba_h_fahrbef=2;
        pos_erreicht_tipp_a1_minus=0;
    }
}

225
if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 8))
{
    if(m_ba_hand == 1)
    {
        ba_h_fahrbef=3;
        pos_erreicht_tipp_a2_plus=0;
    }
}

235
if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 9))
{
    if(m_ba_hand == 1)
    {
        ba_h_fahrbef=4;
        pos_erreicht_tipp_a2_minus=0;
    }
}

245
if((befehl_s1_in[2] == 2) && (befehl_s1_in[3] == 0))
{
    if(m_ba_hand == 1)
    {
        ba_h_fahrbef=5;
        pos_erreicht_tipp_a3_plus=0;
    }
}

255
if((befehl_s1_in[2] == 2) && (befehl_s1_in[3] == 1))
{
    if(m_ba_hand == 1)
    {
        ba_h_fahrbef=6;
        pos_erreicht_tipp_a3_minus=0;
    }
}

265
if((befehl_s1_in[2] == 2) && (befehl_s1_in[3] == 2))
{
    if(m_ba_hand == 1)
    {
        ba_h_fahrbef=7;
        pos_erreicht_tipp_a4_plus=0;
    }
}

```

```

270     if((befehl_s1_in[2] == 2) && (befehl_s1_in[3] == 3))
271     {
272         if(m_ba_hand == 1)
273         {
274             ba_h_fahrbef=8;
275             pos_erreicht_tipp_a4_minus=0;
276         }
277     }
278
279 //-----
280 // sonstige Befehle auslesen
281
282 // nach Endlagenfahrt über LCD Achse freifahren
283 if((befehl_s1_in[2] == 2) && (befehl_s1_in[3] == 5))
284 {
285     achse_freifahren=1;
286 }
287
288 // Befehl "Störung quittieren" auslesen
289 if((befehl_s1_in[2] == 2) && (befehl_s1_in[3] == 4))
290 {
291     err_qtt=1;
292 }
293
294 // -----
295 // Dateiende

```