```
/*=====================================================================================*/
/* Projekt      :   Olli - eine Holzkiste lernt laufen                                 */
/*                                                                                     */
/* Hardware     :   Infineon C509                                                      */
/*                                                                                     */
/* Dateiname    :   error_1.c                                                          */
/*                                                                                     */
/* Version      :   1.4 vom 05. Februar 2004                                           */
/*                                                                                     */
/* Autoren      :   L. Kulf, J. Roos                                                   */
/*                                                                                     */
/* Datei-                                                                              */
/* beschreibung :   - Überprüfungen auf Störung                                        */
/*                  - generieren von Fehlernummern                                     */
/*                  - Reaktionen auf diverse Fehler                                    */
/*                                                                                     */
/*=====================================================================================*/


// -------------------------------------------------------------------------------------
// Dateien einbinden
#include "error_1.h"
#include "mot_outp.h"
#include "grundstellung_1.h"
#include "ba_teilauto.h"
#include "main.h"
#include "seriell_olli_seriell.h"


// -------------------------------------------------------------------------------------
// Variablendeklaration, globale Variablen
_bit error_activ=0;                 // Fehler steht an (flag für Olli-Control)
_bit first_turn_hut=0;              // Aufruf Fehler "hut" nur 1x
_bit first_turn_err=0;
_bit first_turn_neig=0;
_bit first_turn_lcd_neig=0;         // NSensor ausgelöst an LCD gesendet

extern _bit m_ba_auto;                      // Betriebsart Automatik angewählt
extern _bit m_ba_teilauto;                  // Betriebsart TeilAutomatik angew.
extern _bit m_ba_hand;                      // Betriebsart Hand angewählt
extern _bit m_grdstllg_nio;                 // keine Grundstellung
extern _bit flag_ref1_a1;                   // Flag, referieren_1 Achse 1 läuft
extern _bit flag_ref1_a2;                   // Flag, referieren_1 Achse 2 läuft
extern _bit flag_ref1_a3;                   // Flag, referieren_1 Achse 3 läuft
extern _bit flag_ref1_a4;                   // Flag, referieren_1 Achse 4 läuft
extern _bit flag_ref2_a1;                   // Flag, referieren_2 Achse 1 läuft
extern _bit flag_ref2_a2;                   // Flag, referieren_2 Achse 2 läuft
extern _bit flag_ref2_a3;                   // Flag, referieren_2 Achse 3 läuft
extern _bit flag_ref2_a4;                   // Flag, referieren_2 Achse 4 läuft
extern _bit m_all_ax_ref;                   // alle Achsen sind referiert
extern _bit m_all_ax_bb;                    // alle Achsen sind betriebsbereit
                                            // bb: z.Z. referiert
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
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
extern _bit pos_erreicht_tipp_a4_minus;     // Tippen : Position A4 ist erreicht
extern _bit pos_erreicht_ax_free_a1_plus;   // Bits aus Achse freifahren
extern _bit pos_erreicht_ax_free_a2_plus;   // Bits aus Achse freifahren
extern _bit pos_erreicht_ax_free_a3_plus;   // Bits aus Achse freifahren
extern _bit pos_erreicht_ax_free_a4_plus;   // Bits aus Achse freifahren
extern _bit pos_erreicht_ax_free_a1_minus;  // Bits aus Achse freifahren
extern _bit pos_erreicht_ax_free_a2_minus;  // Bits aus Achse freifahren
extern _bit pos_erreicht_ax_free_a3_minus;  // Bits aus Achse freifahren
extern _bit pos_erreicht_ax_free_a4_minus;  // Bits aus Achse freifahren
extern _bit m_mot_a1_plus;                  // Motor Achse 1 R+ ist angesteuert
extern _bit m_mot_a1_minus;                 // Motor Achse 1 R- ist angesteuert
extern _bit m_mot_a2_plus;                  // Motor Achse 2 R+ ist angesteuert
extern _bit m_mot_a2_minus;                 // Motor Achse 2 R- ist angesteuert
extern _bit m_mot_a3_plus;                  // Motor Achse 3 R+ ist angesteuert
extern _bit m_mot_a3_minus;                 // Motor Achse 3 R- ist angesteuert
extern _bit m_mot_a4_plus;                  // Motor Achse 4 R+ ist angesteuert
extern _bit m_mot_a4_minus;                 // Motor Achse 4 R- ist angesteuert

extern unsigned char neu_anzufahrende_pos;  // Teilautomatik : aktiver Fahrbefehl
extern unsigned char move_order_pos;        // anzufahrende Position
extern unsigned char befehl_s1_in[4];       // SS1
extern unsigned char referiert[4];          // Achsen (1-4) referiert
extern unsigned char ba_h_fahrbef;          // aktiver Fahrbefehl:
                                            // 1 = A1 +
                                            // 2 = A1 -
                                            // 3 = A2 +
                                            // 4 = A2 -
                                            // 5 = A3 +
                                            // 6 = A3 -
```

```
                                            // 7 = A4 +
                                            // 8 = A4 -

unsigned char number = 0;                   // die übergebene Fehlernummer in
                                            // der "Reaktionsfunktion"

// in den diversen Funktionen erzeugte Fehlernummer, die an die
// "Reaktionsfunktion" zur Auswertung übergeben wird
unsigned char err_number = 0;


// ----------------------------------------------------------------------------
// Fehlernummer: 1 + 24
// Überprüfen, ob Olli umgefallen ist
// Wenn Neigungsschalter an P1.3=true, dann wird error_break() aufgerufen
// und Fehlernummer übergeben.
void error_switch_drop(void)
{
    // Fehlernummer 1:
    // Wird gebildet, wenn alle Achsen referiert sind und Neigungssensor in
    // BA Auto oder BA TeilAuto anspricht
    if  ((m_all_ax_ref == 1) && (first_turn_neig == 0)
        &&((m_ba_auto == 1) || (m_ba_teilauto == 1)))
    {
        err_number = 1;
        error_break(err_number);
    }


    // Fehlernummer 24:
    // Wird nur gebildet, um eine Warnung auf dem LCD auszugeben, falls Olli
    // beim Aufbau zur Messung mit Spannung versorgt wird aber noch nicht in
    // Arbeitsposition steht
    // Fehlernummer ist selbstquittierend
    if((!m_all_ax_ref) && (m_ba_auto))
    {
        err_number = 24;
        error_break(err_number);
    }
}


// ----------------------------------------------------------------------------
// Fehlernummer: 2 - 13
// Funktion wird aufgerufen durch Interrupt 4 (11) an Port 1.1 in intrrpt_functions.c
// Endlagenschalter liegen an Port 7
// Endlagenschalter sind ÖFFNER-Kontakte, somit Störung bei LOW-Pegel !
// -> ACHTUNG ! Kontakte sind hardwaremäßig invertiert, somit wie Schliesser
// zu programmieren.

void error_endlage(void)
{
    // ------------------------------------------------------------------------
    // Variablendeklaration, lokale Variablen
    unsigned char achse_endlage=0;      // Eingangsbyte, Endlagenschalter Achsen

    // ------------------------------------------------------------------------

    // Portbyte mit Endlagenschalter einlesen
    achse_endlage=P7;

    // Fehlernummer: 2
    // Überprüfe auf Paarfehler, Achse 1
    // Bit 7.0+7.1 = 1 ?
    if (achse_endlage == 0x03)
    {
        err_number = 2;
        error_break(err_number);
    }

    // Fehlernummer: 3
    // Überprüfe auf Paarfehler, Achse 2
    // Bit 7.2+7.3 = 1 ?
    if (achse_endlage == 0x0c)
    {
        err_number = 3;
        error_break(err_number);
    }

    // Fehlernummer: 4
    // Überprüfe auf Paarfehler, Achse 3
    // Bit 7.4+7.5 = 1 ?
    if (achse_endlage == 0x30)
    {
        err_number = 4;
        error_break(err_number);
    }

    // Fehlernummer: 5
```

```
        /// Überprüfe auf Endlage+, Achse 1
        // Bit 7.1 = 1 ?
        if (achse_endlage == 0x02)
        {
            err_number = 7;
            error_break(err_number);
        }

        // Fehlernummer: 8
        // Überprüfe auf Endlage-, Achse 2
        // Bit 7.2 = 1 ?
        if (achse_endlage == 0x04)
        {
            err_number = 8;
            error_break(err_number);
        }

        // Fehlernummer: 9
        // Überprüfe auf Endlage+, Achse 2
        // Bit 7.3 = 1 ?
        if (achse_endlage == 0x08)
        {
            err_number = 9;
            error_break(err_number);
        }

        // Fehlernummer: 10
        // Überprüfe auf Endlage-, Achse 3
        // Bit 7.4 = 1 ?
        if (achse_endlage == 0x10)
        {
            err_number = 10;
            error_break(err_number);
        }

        // Fehlernummer: 11
        // Überprüfe auf Endlage+, Achse 3
        // Bit 7.5 = 1 ?
        if (achse_endlage == 0x20)
        {
            err_number = 11;
            error_break(err_number);
        }

        // Fehlernummer: 12
        // Überprüfe auf Endlage-, Achse 4
        // Bit 7.6 = 1 ?
        if (achse_endlage == 0x40)
        {
            err_number = 12;
            error_break(err_number);
        }

        // Fehlernummer: 13
        // Überprüfe auf Endlage+, Achse 4
        // Bit 7.7 = 1 ?
        if (achse_endlage == 0x80)
        {
            err_number = 13;
            error_break(err_number);
        }
}


    // -------------------------------------------------------------------------
    // Fehlernummer: 14
    // Überwachung und Auswertung der OCD - Ausgänge der Motoransteuerungen
    // Anschluß des ODER-Ergebnisses der OCD - Ausgänge an Port 1.2, Interrupt 5 (12)
    // Funktion wird aufgerufen durch Interrupt 5 (12) an Port 1.2 in intrrpt_functions.c
    void error_overload (void)
    {
        err_number = 14;
        error_break(err_number);
    }


    // -------------------------------------------------------------------------
    // Fehlernummer: 15
    // Überprüfen, ob Olli seinen Hut genommen hat (der Deckel abgenommen wurde)
    void error_without_hut (void)
    {
        err_number = 15;
        error_break(err_number);
    }


    // -------------------------------------------------------------------------
    // Fehlernummer: 16
    // keine Grundstellung beim Umschalten auf BA Automatik am Display
```

```
void error_grdstllg (void)
{
    err_number = 16;
    error_break(err_number);
}


// ------------------------------------------------------------------------
// Fehlernummer: 17 (Meldung)
// in BA TA wurde Grundstellung angefordert, obwohl Olli schon in Grundstellung
// ist
void error_grdstllg_ok (void)
{
    err_number = 17;
    error_break(err_number);
}
// ------------------------------------------------------------------------
// Fehlernummer: 18 (Meldung)
// in BA TA Fahrt nach P1 wurde Position erreicht
void msg_p1 (void)
{
    err_number = 18;
    error_break(err_number);
}
// ------------------------------------------------------------------------
// Fehlernummer: 19 (Meldung)
// in BA TA Fahrt nach P2 wurde Position erreicht
void msg_p2 (void)
{
    err_number = 19;
    error_break(err_number);
}
// ------------------------------------------------------------------------
// Fehlernummer: 20 (Meldung)
// in BA TA Fahrt nach P3 wurde Position erreicht
void msg_p3 (void)
{
    err_number = 20;
    error_break(err_number);
}
// ------------------------------------------------------------------------
// Fehlernummer: 21 (Meldung)
// in BA TA Fahrt nach P4 wurde Position erreicht
void msg_p4 (void)
{
    err_number = 21;
    error_break(err_number);
}
// ------------------------------------------------------------------------
// Fehlernummer: 22 (Meldung)
// in BA TA Fahrt nach P5 wurde Position erreicht
void msg_p5 (void)
{
    err_number = 22;
    error_break(err_number);
}
// ------------------------------------------------------------------------
// Fehlernummer: 23 (Meldung)
// in BA Hand nur umschalten, wenn keine Achse gerade aktiv ist
void msg_ax_activ (void)
{
    err_number = 23;
    error_break(err_number);
}


// ------------------------------------------------------------------------
// ACHTUNG !
// Fehlernummer 24 wurde schon vergeben, siehe bei Bildung der Fehlernummer 1
// ------------------------------------------------------------------------


// ------------------------------------------------------------------------
// Reaktion auf entsprechende Fehlernummer vorbereiten
void error_break(unsigned char number)
{
    // ------------------------------------------------------------------------
    // Variablendeklaration, lokale Variablen
    _bit first_turn=0;
    unsigned char b1=0;                 // im LCD aufzurufende Makroart (N;P;T)
    unsigned char b2=0;                 // Makronummer

    // ------------------------------------------------------------------------

    // ------------------------------------------------------------------------
    if (number == 1)
    {
        first_turn_neig=1;
```

```
375         // Motoransteuerungen disable
            mot_pwr_disable();

            // Meldung bzw. Bedienerführung auf Display ausgeben:
            // Olli aufrichten !
380
            // LCD-Bild aufrufen
            // Makro 44
            b1='N';
            b2=0x2C;
385         transmit_S1_vorbereitung(b1,b2);
        }

        // ---------------------------------------------------------------------------
        // Paarfehler, Achse 1
390     if (number == 2)
        {
            // LCD-Bild aufrufen
            // Makro 20
            b1='N';
395         b2=0x14;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ---------------------------------------------------------------------------
400     // Paarfehler, Achse 2
        if (number == 3)
        {
            // LCD-Bild aufrufen
            // Makro 21
405         b1='N';
            b2=0x15;
            transmit_S1_vorbereitung(b1,b2);
        }

410     // ---------------------------------------------------------------------------
        // Paarfehler, Achse 3
        if (number == 4)
        {
            // LCD-Bild aufrufen
415         // Makro 22
            b1='N';
            b2=0x16;
            transmit_S1_vorbereitung(b1,b2);
        }
420
        // ---------------------------------------------------------------------------
        // Paarfehler, Achse 4
        if (number == 5)
        {
425         // LCD-Bild aufrufen
            // Makro 23
            b1='N';
            b2=0x17;
            transmit_S1_vorbereitung(b1,b2);
430     }

        // ---------------------------------------------------------------------------
        // Achse 1 Endlage -
        if (number == 6)
435     {
            // Interrupt 4 (Endlagenüberwachung) ausschalten
            EX4=0;

            // Ausgänge Motoren zurücksetzen
440         mot_a1_stp();

            // Motoransteuerungen disable
            mot_pwr_disable();

445         // Variable initialisieren
            init_var();

            // Ausgänge initialisieren
            init_outp();
450
            // LCD-Bild aufrufen
            // Makro 28
            b1='N';
            b2=0x1C;
455         transmit_S1_vorbereitung(b1,b2);

        }

        // ---------------------------------------------------------------------------
460     // Achse 1 Endlage +
        if (number == 7)
        {
            // Interrupt 4 (Endlagenüberwachung) ausschalten
```

```
            EX4=0;

            // Ausgänge Motoren zurücksetzen
            mot_a1_stp();

            // Motoransteuerungen disable
            mot_pwr_disable();

            // Variable initialisieren
            init_var();

            // Ausgänge initialisieren
            init_outp();

            // LCD-Bild aufrufen
            // Makro 24
            b1='N';
            b2=0x18;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ---------------------------------------------------------------------------
        // Achse 2 Endlage -
        if (number == 8)
        {
            // Interrupt 4 (Endlagenüberwachung) ausschalten
            EX4=0;

            // Ausgänge Motoren zurücksetzen
            mot_a2_stp();

            // Motoransteuerungen disable
            mot_pwr_disable();

            // Variable initialisieren
            init_var();

            // Ausgänge initialisieren
            init_outp();

            // LCD-Bild aufrufen
            // Makro 29
            b1='N';
            b2=0x1D;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ---------------------------------------------------------------------------
        // Achse 2 Endlage +
        if (number == 9)
        {
            // Interrupt 4 (Endlagenüberwachung) ausschalten
            EX4=0;

            // Ausgänge Motoren zurücksetzen
            mot_a2_stp();

            // Motoransteuerungen disable
            mot_pwr_disable();

            // Variable initialisieren
            init_var();

            // Ausgänge initialisieren
            init_outp();

            // LCD-Bild aufrufen
            // Makro 25
            b1='N';
            b2=0x19;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ---------------------------------------------------------------------------
        // Achse 3 Endlage -
        if (number == 10)
        {
            // Interrupt 4 (Endlagenüberwachung) ausschalten
            EX4=0;

            // Ausgänge Motoren zurücksetzen
            mot_a3_stp();

            // Motoransteuerungen disable
            mot_pwr_disable();

            // Variable initialisieren
            init_var();
```

```
            // Ausgänge initialisieren
            init_outp();

            // LCD-Bild aufrufen
            // Makro 30
            b1='N';
            b2=0x1E;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ----------------------------------------------------------------------
        // Achse 3 Endlage +
        if (number == 11)
        {
            // Interrupt 4 (Endlagenüberwachung) ausschalten
            EX4=0;

            // Ausgänge Motoren zurücksetzen
            mot_a3_stp();

            // Motoransteuerungen disable
            mot_pwr_disable();

            // Variable initialisieren
            init_var();

            // Ausgänge initialisieren
            init_outp();

            // LCD-Bild aufrufen
            // Makro 26
            b1='N';
            b2=0x1A;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ----------------------------------------------------------------------
        // Achse 4 Endlage -
        if (number == 12)
        {
            // Interrupt 4 (Endlagenüberwachung) ausschalten
            EX4=0;

            // Ausgänge Motoren zurücksetzen
            mot_a4_stp();

            // Motoransteuerungen disable
            mot_pwr_disable();

            // Variable initialisieren
            init_var();

            // Ausgänge initialisieren
            init_outp();

            // LCD-Bild aufrufen
            // Makro 31
            b1='N';
            b2=0x1F;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ----------------------------------------------------------------------
        // Achse 4 Endlage +
        if (number == 13)
        {
            // Interrupt 4 (Endlagenüberwachung) ausschalten
            EX4=0;

            // Ausgänge Motoren zurücksetzen
            mot_a4_stp();

            // Motoransteuerungen disable
            mot_pwr_disable();

            // Variable initialisieren
            init_var();

            // Ausgänge initialisieren
            init_outp();

            // LCD-Bild aufrufen
            // Makro 27
            b1='N';
            b2=0x1B;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ----------------------------------------------------------------------
```

```
    // Achsen Überlast
    if (number == 14)
    {
        // Ausgänge Motoren zurücksetzen
        mot_a1_stp();
        mot_a2_stp();
        mot_a3_stp();
        mot_a4_stp();

        // Motoransteuerungen disable
        mot_pwr_disable();

        // LCD-Bild aufrufen
        // Makro 32
        b1='N';
        b2=0x20;
        transmit_S1_vorbereitung(b1,b2);
    }

    // ---------------------------------------------------------------------
    // Deckel nicht abgenommen
    if (number == 15)
    {
        if(!first_turn_hut)
        {
            // Motoransteuerungen disable
            mot_pwr_disable();

            // Meldung bzw. Bedienerführung auf Display ausgeben:
            // Deckel muß abgenommen werden !
            // LCD-Bild aufrufen
            // Makro 46
            b1='N';
            b2=0x2E;
            transmit_S1_vorbereitung(b1,b2);

            first_turn_hut=1;
        }
    }

    // ---------------------------------------------------------------------
    // keine Grundstellung beim Umschalten auf BA Automatik am Display
    if (number == 16)
    {
        // LCD-Bild aufrufen
        // Makro 36
        b1='N';
        b2=0x24;
        transmit_S1_vorbereitung(b1,b2);
        m_grdstllg_nio=0;
    }

    // ---------------------------------------------------------------------
    // in BA TA wurde Grundstellung angefordert, obwohl Olli schon in Grund-
    // stellung ist
    if (number == 17)
    {
        // LCD-Bild aufrufen
        // Makro 37
        b1='N';
        b2=0x25;
        transmit_S1_vorbereitung(b1,b2);
    }

    // ---------------------------------------------------------------------
    // in BA TA wurde angeforderte Position erreicht (P1)
    if (number == 18)
    {
        // LCD-Bild aufrufen
        // Makro 38
        b1='N';
        b2=0x26;
        transmit_S1_vorbereitung(b1,b2);
    }

    // ---------------------------------------------------------------------
    // in BA TA wurde angeforderte Position erreicht (P2)
    if (number == 19)
    {
        // LCD-Bild aufrufen
        // Makro 39
        b1='N';
        b2=0x27;
        transmit_S1_vorbereitung(b1,b2);
    }

    // ---------------------------------------------------------------------
    // in BA TA wurde angeforderte Position erreicht (P3)
    if (number == 20)
```

```
        {
            // LCD-Bild aufrufen
            // Makro 40
            b1='N';
            b2=0x28;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ----------------------------------------------------------------------------
        // in BA TA wurde angeforderte Position erreicht (P4)
        if (number == 21)
        {
            // LCD-Bild aufrufen
            // Makro 41
            b1='N';
            b2=0x29;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ----------------------------------------------------------------------------
        // in BA TA wurde angeforderte Position erreicht (P5)
        if (number == 22)
        {
            // LCD-Bild aufrufen
            // Makro 42
            b1='N';
            b2=0x2A;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ----------------------------------------------------------------------------
        // in BA Hand nur umschalten, wenn gerade keine Achse aktiv
        if (number == 23)
        {
            // LCD-Bild aufrufen
            // Makro 43
            b1='N';
            b2=0x2B;
            transmit_S1_vorbereitung(b1,b2);
        }

        // ----------------------------------------------------------------------------
        // Neigungsschalter hat angesprochen, bevor Olli referiert wurde
        if (number == 24)
        {
            if(!first_turn_lcd_neig)
            {
                // LCD-Bild aufrufen
                // Makro 45
                befehl_s1_in[1] = 0;
                b1='N';
                b2=0x2D;
                transmit_S1_vorbereitung(b1,b2);

                // Fehlernummer sofort zurücksetzen, da nur Meldung auf LCD
                // ausgegeben wird
                err_number=0;

                first_turn_lcd_neig=1;
            }
        }

        // ----------------------------------------------------------------------------
        // Bei Störung Freigabe von Olli-Control entziehen
        if(err_number!=0)
        {
            error_activ=1;

            if((error_activ == 1) && (first_turn_err == 0))
            {
                olli_control_disable();
                first_turn_err=1;
            }
        }
}

    // ----------------------------------------------------------------------------
    // Positionsmeldungen auf Display in BA Automatik ausgeben
    // Positionsmeldungen auf Display in BA TeilAutomatik ausgeben
// aktive Positionsfahrten auf Display ausgeben

    void ba_a_msg_p1(void)
    {
        // LCD-Bild aufrufen
        // Makro 50
        transmit_S1_vorbereitung('N',0x32);
    }

    void ba_a_msg_p2(void)
```

```
{
    // LCD-Bild aufrufen
    // Makro 51
    transmit_S1_vorbereitung('N',0x33);
}

void ba_a_msg_p3(void)
{
    // LCD-Bild aufrufen
    // Makro 52
    transmit_S1_vorbereitung('N',0x34);
}

void ba_a_msg_p4(void)
{
    // LCD-Bild aufrufen
    // Makro 53
    transmit_S1_vorbereitung('N',0x35);
}

void ba_a_msg_p5(void)
{
    // LCD-Bild aufrufen
    // Makro 54
    transmit_S1_vorbereitung('N',0x36);
}

void ba_a_msg_p1_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 55
    transmit_S1_vorbereitung('N',0x37);
}

void ba_a_msg_p2_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 56
    transmit_S1_vorbereitung('N',0x38);
}

void ba_a_msg_p3_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 57
    transmit_S1_vorbereitung('N',0x39);
}

void ba_a_msg_p4_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 58
    transmit_S1_vorbereitung('N',0x3A);
}

void ba_a_msg_p5_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 59
    transmit_S1_vorbereitung('N',0x3B);
}

void ba_a_msg_p6_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 60
    transmit_S1_vorbereitung('N',0x3C);
}

void ba_ta_msg_p1_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 61
    transmit_S1_vorbereitung('N',0x3D);
}

void ba_ta_msg_p2_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 62
    transmit_S1_vorbereitung('N',0x3E);
}

void ba_ta_msg_p3_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 63
    transmit_S1_vorbereitung('N',0x3E);
}
```

```
void ba_ta_msg_p4_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 64
    transmit_S1_vorbereitung('N',0x40);
}

void ba_ta_msg_p5_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 65
    transmit_S1_vorbereitung('N',0x41);
}

void ba_ta_msg_p6_aktiv(void)
{
    // LCD-Bild aufrufen
    // Makro 66
    transmit_S1_vorbereitung('N',0x41);
}

// ------------------------------------------------------------------------
// Variablen initialisieren um bei Störungen bereits angeforderte
// Positionen, etc., abzulöschen
void init_var(void)
{
    m_ba_auto=0;
    m_ba_teilauto=0;
    m_mot_a1_plus=0;
    m_mot_a1_minus=0;
    m_mot_a2_plus=0;
    m_mot_a2_minus=0;
    m_mot_a3_plus=0;
    m_mot_a3_minus=0;
    m_mot_a4_plus=0;
    m_mot_a4_minus=0;
    neu_anzufahrende_pos=0;
    move_order_pos=0;
    flag_ref1_a1=0;                     // Flag, referieren_1 Achse 1 läuft
    flag_ref1_a2=0;                     // Flag, referieren_1 Achse 2 läuft
    flag_ref1_a3=0;                     // Flag, referieren_1 Achse 3 läuft
    flag_ref1_a4=0;                     // Flag, referieren_1 Achse 4 läuft
    flag_ref2_a1=0;                     // Flag, referieren_2 Achse 1 läuft
    flag_ref2_a2=0;                     // Flag, referieren_2 Achse 2 läuft
    flag_ref2_a3=0;                     // Flag, referieren_2 Achse 3 läuft
    flag_ref2_a4=0;                     // Flag, referieren_2 Achse 4 läuft
    m_new_lcd_order=0;                  // neuer Befehl vom LCD vorhanden
    ba_h_fahrbef=0;                     // aktiver Fahrbefehl
    pos_erreicht_tipp_a1_plus=0;        // Tippen : Position A1 ist erreicht
    pos_erreicht_tipp_a1_minus=0;       // Tippen : Position A1 ist erreicht
    pos_erreicht_tipp_a2_plus=0;        // Tippen : Position A2 ist erreicht
    pos_erreicht_tipp_a2_minus=0;       // Tippen : Position A2 ist erreicht
    pos_erreicht_tipp_a3_plus=0;        // Tippen : Position A3 ist erreicht
    pos_erreicht_tipp_a3_minus=0;       // Tippen : Position A3 ist erreicht
    pos_erreicht_tipp_a4_plus=0;        // Tippen : Position A4 ist erreicht
    pos_erreicht_tipp_a4_minus=0;       // Tippen : Position A4 ist erreicht

    pos_erreicht_ax_free_a1_plus=0;     // Bits aus Achse freifahren
    pos_erreicht_ax_free_a2_plus=0;
    pos_erreicht_ax_free_a3_plus=0;
    pos_erreicht_ax_free_a4_plus=0;
    pos_erreicht_ax_free_a1_minus=0;
    pos_erreicht_ax_free_a2_minus=0;
    pos_erreicht_ax_free_a3_minus=0;
    pos_erreicht_ax_free_a4_minus=0;
}

// ------------------------------------------------------------------------
// Initialisierung der Ausgänge, die Motoren, bzw. Bewegungen ansteuern
void init_outp(void)
{
    P4=0x00;
}

// ------------------------------------------------------------------------
// tool Olli-Control Freigabe erteilen
void olli_control_enable(void)
{
    transmit_S0_vorbereitung(8);
}

// ------------------------------------------------------------------------
// tool Olli-Control Freigabe entziehen
void olli_control_disable(void)
{
    transmit_S0_vorbereitung(7);
}
```

```
    // ------------------------------------------------------------------------
1000 // Dateiende
```