

```

1  /*=====*/
/*
/* Hardware      :   Infineon C509
/*
/*
5 /* Dateiname    :   grundstellung_1.c
/*
/* Version       :   1.3 vom 12. März 2004
/*
/* Autoren       :   L. Kulf, J. Roos
10 /*
/* Datei-
/* beschreibung  :   Überprüfung auf Transportstellung,
/*
/*                  Überprüfung auf Grundstellung,
/*                  Referenzpunktfahrt der Achsen 1-4
15 /*
/*=====*/

// -----
// Transportstellung :
20 // Endlage (-) und Refpktschalter ist angefahren

// z.Z.: sind Endlagenschalter im Ablauf nicht berücksichtigt, d.h., angefahrne
// Endlagenschalter melden Störung und haben ein Abschalten des Antriebes zur
// Folge ! Werden aber nicht zu Positionierungen genutzt !
25 // -----
// Grundstellung :
// vorgegebener Wert, z. B. 5 Spindelumdrehungen in Richtung (-) von Refpunkt-
// schalter, ist erreicht.
30 // z.Z.: gibt es keine besondere Grundstellung, als Grundstellung wird Referenz-
// punkt verwendet. Wenn die Mechanik dies später zulässt, kann dies auch so
// bleiben.

35 // -----
// Dateien einbinden
#include "grundstellung_1.h"
#include "umdrehungszaehler.h"
#include "mot_outp.h"
40 #include "geberkanal.h"
#include "regsc509.sfr"
#include "main.h"
#include "seriell_olli_seriell.h"

45 // -----
// Variablendeklaration, globale Variablen
_bit flag_ref1_a1=0;           // Flag, referieren_1 Achse 1 läuft
_bit flag_ref1_a2=0;           // Flag, referieren_1 Achse 2 läuft
_bit flag_ref1_a3=0;           // Flag, referieren_1 Achse 3 läuft
50 _bit flag_ref1_a4=0;           // Flag, referieren_1 Achse 4 läuft
_bit flag_ref2_a1=0;           // Flag, referieren_2 Achse 1 läuft
_bit flag_ref2_a2=0;           // Flag, referieren_2 Achse 2 läuft
_bit flag_ref2_a3=0;           // Flag, referieren_2 Achse 3 läuft
_bit flag_ref2_a4=0;           // Flag, referieren_2 Achse 4 läuft
55 _bit change_pm_a1=0;          // Flag, Richtungsumkehr A1
_bit change_pm_a2=0;          // Flag, Richtungsumkehr A2
_bit change_pm_a3=0;          // Flag, Richtungsumkehr A3
_bit change_pm_a4=0;          // Flag, Richtungsumkehr A4

60 _bit m_all_ax_ref=0;          // Merker, alle Achsen sind referiert
_bit m_all_ax_bb=0;            // Merker, alle Achsen sind betriebsbereit
// bb: z.Z. referiert
// eventuell später referiert+grundstellung

65 extern _bit m_mot_a1_plus;    // Merker, Ausgang Motor Achse 1 R+ ist gesetzt
extern _bit m_mot_a2_plus;    // Merker, Ausgang Motor Achse 2 R+ ist gesetzt
extern _bit m_mot_a3_plus;    // Merker, Ausgang Motor Achse 3 R+ ist gesetzt
extern _bit m_mot_a4_plus;    // Merker, Ausgang Motor Achse 4 R+ ist gesetzt
extern _bit m_mot_a1_minus;   // Merker, Ausgang Motor Achse 1 R- ist gesetzt
70 extern _bit m_mot_a2_minus;   // Merker, Ausgang Motor Achse 2 R- ist gesetzt
extern _bit m_mot_a3_minus;   // Merker, Ausgang Motor Achse 3 R- ist gesetzt
extern _bit m_mot_a4_minus;   // Merker, Ausgang Motor Achse 4 R- ist gesetzt

extern signed char s_umdr_a1;  // aktuelle Spindelumdrehungen Achse 1
75 extern signed char s_umdr_a2;  // aktuelle Spindelumdrehungen Achse 2
extern signed char s_umdr_a3;  // aktuelle Spindelumdrehungen Achse 3
extern signed char s_umdr_a4;  // aktuelle Spindelumdrehungen Achse 4

extern unsigned char refp[4];  // Refpspeicher Achse 1-4, Schalter bitadr
80 unsigned char referiert[4]={0}; // Achsen sind referiert, [0]-a1, [1]-a2, usw.
unsigned char ax_bb[4]={0};    // Speicher, einzelne Achsen betriebsbereit
unsigned int ax_wait_time=0;    // Wartezeit Richtungsumkehr referieren_1

85 void referieren_haupt (void)
{
// a1234_endl ist 0x55, wenn die Endlagen (-) der 4 Achsen angefahren sind.

// a1234_refp ist 0x0f, wenn die Refpunktschalter der 4 Achsen angefahren

```

```

// sind.
// Wenn das der Fall ist, dann werden die Refpunkte neu angefahren, um die
// Achsen in eindeutige mechanische Positionen zu bringen. Welche Achsen
// referiert sind, wird in referiert[3] festgehalten. In refp[3] steht der
95 // Schaltzustand des Schalters.
//-----
// Wenn nur Refpschalter der jeweiligen Achse angefahren ist, wird referieren_1
// gewählt:
// referieren_1 : 5 Spindelumdrehungen nach Richtung + fahren
100 // Motor aus
// Achse Richtung - fahren, bis Refpschalter angefahren ist

// Wenn kein Schalter der jeweiligen Achse angefahren ist, wird referieren_2
// gewählt:
105 // referieren_2 : Achse Richtung - fahren, bis Refpschalter angefahren ist

// Die in referieren_1 festgelegten Spindelumdrehungen werden in
// grundstellung_1.h festgelegt.

110

//-----
// Achse 1

115 if(((refp[0] == true)
    &&(!referiert[0])
    &&(!flag_ref2_a1)
    ||(flag_ref1_a1 == true))
    {referieren_1_a1(s_umdr_a1);}

120 if( (!referiert[0])
    &&(!flag_ref1_a1))
    {referieren_2_a1();}

125 //-----
// Achse 2

    if(((refp[1] == true)
        &&(!referiert[1])
        &&(!flag_ref2_a2)
        &&(referiert[0] == true)
        ||(flag_ref1_a2 == true))
        {referieren_1_a2(s_umdr_a2);}

130

135 if((!referiert[1])
    &&(!flag_ref1_a2)
    &&(referiert[0]))
    {referieren_2_a2();}

140 //-----
// Achse 3

    if(((refp[2] == true)
        &&(!referiert[2])
        &&(!flag_ref2_a3)
        &&(referiert[0] == true)
        &&(referiert[1] == true)
        ||(flag_ref1_a3))
        {referieren_1_a3(s_umdr_a3);}

145

150 if((!referiert[2])
    &&(!flag_ref1_a3)
    &&(referiert[0] == true)
    &&(referiert[1] == true))
    {referieren_2_a3();}

155 //-----
// Achse 4

160 if(((refp[3] == true)
    &&(!referiert[3])
    &&(!flag_ref2_a4)
    &&(referiert[0] == true)
    &&(referiert[1] == true)
    &&(referiert[2] == true)
    ||(flag_ref1_a4))
    {referieren_1_a4(s_umdr_a4);}

165

170 if((!referiert[3])
    &&(!flag_ref1_a4)
    &&(referiert[0] == true)
    &&(referiert[1] == true)
    &&(referiert[2] == true))
    {referieren_2_a4();}

175

// Aufruf Funktion, alle Achsen betriebsbereit und referiert abfragen
all_ax_bb_ref();

```

```

180     } // Ende referieren_haupt()

// -----
185 void referieren_1_al (signed char s_umdr_al)
{
    // Wenn Motor A1 R+ noch nicht läuft, dann einschalten
    // Geberdatenkanal wird autom. über mot_al_plus() ausgewählt
    if (!m_mot_al_plus) && (!m_mot_al_minus)
190     {
        // Motor A1 Richtung+ ein
        mot_al_plus();
    }

195     // Wenn Spindelumdrehungen = Sollwert, dann Motor stop,
    // anschließend Motor A1 R-
    if (s_umdr_al == S_U_SOLL)
    {
        mot_al_stp();

200         if (change_pm_al == 0)
        {
            change_pm_al=1;
            ax_wait();

205         }

        // Motor A1 Richtung- ein
        mot_al_minus();
    }

210     // Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse "ist
    // referiert" setzen und Motor ausschalten
    if ((refp[0] == 1) && (m_mot_al_minus))
    {
215         // Achse 1 ist referiert
        referiert[0]=1;

        // Flag "referieren_1_Achse_1_läuft" rücksetzen
        flag_ref1_al=0;

220         // Achse 1 ist betriebsbereit setzen
        ax_bb[0]=1;

        // Motor A1 ausschalten
        mot_al_stp();

225         // alle Zähler rücksetzen
        umdr_zaeher_init_al();
    }
    else
    {
230         // Flag "referieren_1_Achse_1_läuft" setzen
        flag_ref1_al=1;
    }

235 }

// -----
void referieren_2_al (void)
{
240     // falls Motor in R- noch nicht läuft, einschalten
    if(m_mot_al_minus == 0)
    {
        // Motor A1 Richtung- ein
        mot_al_minus();

245     }

    // Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse ist
    // referiert setzen und Motor ausschalten
250     if (refp[0] == 1)
    {
        referiert[0]=1;

        // Achse 1 ist betriebsbereit setzen
        ax_bb[0]=1;
        flag_ref2_al=0;
        mot_al_stp();

255         // alle Zähler rücksetzen
        umdr_zaeher_init_al();
    }

    // flag setzen, daß nicht referieren1_al angewählt wird
    // rücksetzen von flag_ref2_al erst, wenn referieren 1/2 _a2 läuft
265     flag_ref2_al=1;
}

// -----

```

```

void referieren_1_a2 (signed char s_umdr_a2)
270 {
    if ((!m_mot_a2_plus) && (!m_mot_a2_minus))
    {
        // Motor A2 Richtung+ ein
        mot_a2_plus();
275     }

    // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
    if (s_umdr_a2 == S_U_SOLL)
    {
280         mot_a2_stp();

        if (change_pm_a2 == 0)
        {
            change_pm_a2=1;
285             ax_wait();
        }

        // Motor A2 Richtung- ein
        mot_a2_minus();
290     }

    // Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse ist
    // referiert setzen und Motor ausschalten
295     if ((refp[1] == 1) && (m_mot_a2_minus))
    {
        referiert[1]=1;
        flag_ref1_a2=0;

300         // Achse 2 ist betriebsbereit setzen
        ax_bb[1]=1;

        mot_a2_stp();

305         // alle Zähler rücksetzen
        umdr_zaeher_init_a2();
    }
    else
    {
310         flag_ref1_a2=1;
    }
}

// -----
315 void referieren_2_a2 (void)
{
    if(m_mot_a2_minus == 0)
    {
        // Motor A2 Richtung- ein
320         mot_a2_minus();
    }

    // Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse ist
    // referiert setzen und Motor ausschalten
325     if (refp[1] == 1)
    {
        referiert[1]=1;

        // Achse 2 ist betriebsbereit setzen
        ax_bb[1]=1;
        flag_ref2_a2=0;
        mot_a2_stp();

330         // alle Zähler rücksetzen
        umdr_zaeher_init_a2();
    }
    flag_ref2_a2=1;
}

// -----
340 void referieren_1_a3 (signed char s_umdr_a3)
{
    if ((!m_mot_a3_plus) && (!m_mot_a3_minus))
    {
345         // Motor A3 Richtung+ ein
        mot_a3_plus();
    }

    // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
350     if (s_umdr_a3 == S_U_SOLL)
    {
        mot_a3_stp();

        if (change_pm_a3 == 0)
355         {
            change_pm_a3=1;
            ax_wait();
        }
    }
}

```

```

    }

360     // Motor A3 Richtung- ein
    mot_a3_minus();
}

// Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse ist
365 // referiert setzen und Motor ausschalten
if ((refp[2] == 1) && (m_mot_a3_minus))
{
    referiert[2]=1;
    flag_ref1_a3=0;

370     // Achse 3 ist betriebsbereit setzen
    ax_bb[2]=1;

    mot_a3_stp();

375     // alle Zähler rücksetzen
    umdr_zaeher_init_a3();
}
else
380 {
    flag_ref1_a3=1;
}
}

385 // -----
void referieren_2_a3 (void)
{
    if(m_mot_a3_minus == 0)
    {
390         // Motor A1 Richtung- ein
        mot_a3_minus();
    }

    // Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse ist
395 // referiert setzen und Motor ausschalten
    if (refp[2] == 1)
    {
        referiert[2]=1;

400         // Achse 3 ist betriebsbereit setzen
        ax_bb[2]=1;
        flag_ref2_a3=0;
        mot_a3_stp();

405         // alle Zähler rücksetzen
        umdr_zaeher_init_a3();
    }
    flag_ref2_a3=1;
}

410 // -----
void referieren_1_a4 (signed char s_umdr_a4)
{
    if((!m_mot_a4_plus) && (!m_mot_a4_minus))
415 {
        // Motor A4 Richtung+ ein
        mot_a4_plus();
    }

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

425         if (change_pm_a4 == 0)
        {
            change_pm_a4=1;
            ax_wait();
        }

430         // Motor A4 Richtung- ein
        mot_a4_minus();
    }

    // Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse ist
435 // referiert setzen und Motor ausschalten
    if ((refp[3] == 1) && (m_mot_a4_minus == 1))
    {
        referiert[3]=1;
        flag_ref1_a4=0;

440         // Achse 4 ist betriebsbereit setzen
        ax_bb[3]=1;

445         mot_a4_stp();
    }
}

```

```

        // alle Zähler rücksetzen
        umdr_zae_hler_init_a4();
    }
450     else
    {
        flag_ref1_a4=1;
    }
}
455 // -----
void referieren_2_a4 (void)
{
    if(m_mot_a4_minus == 0)
460     {
        // Motor A4 Richtung- ein
        mot_a4_minus();
    }

465     // Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse ist
    // referiert setzen und Motor ausschalten
    if (refp[3] == 1)
    {
        referiert[3]=1;
470
        // Achse 4 ist betriebsbereit setzen
        ax_bb[3]=1;
        flag_ref2_a4=0;
        mot_a4_stp();
475
        // alle Zähler rücksetzen
        umdr_zae_hler_init_a4();
    }
    flag_ref2_a4=1;
480 }

// -----
// Merkerbit für "alle Achsen sind referiert" setzen + rücksetzen
// Merkerbit für "alle Achsen betriebsbereit" setzen + rücksetzen
485 void all_ax_bb_ref(void)
{
    if ((referiert[0])
        &&(referiert[1])
        &&(referiert[2])
490         &&(referiert[3]))
    {
        m_all_ax_ref=1;
        change_pm_a1=0;
        change_pm_a2=0;
495         change_pm_a3=0;
        change_pm_a4=0;
    }

    if ((m_all_ax_ref)
500         &&(ax_bb[0])
        &&(ax_bb[1])
        &&(ax_bb[2])
        &&(ax_bb[3]))
    {
505         m_all_ax_bb=1;

        // Bild Betriebsartenwahl aufrufen
        transmit_S1_vorbereitung('N',0x03);
    }
510 }

// -----
// Wartezeit für Richtungsumkehr der Spindel bei referieren_1 zur
515 // Strombegrenzung
void ax_wait(void)
{
    for(ax_wait_time=0;ax_wait_time<=20000;ax_wait_time++)
    {}
520 }

// -----
// Dateiende

```