

Dokumentation MOTCO

Dok-Rev. 1.0 vom 14.11.2007

Hardware-Rev. 2.0 vom 26.06.1996

Inhaltsverzeichnis

1	Allgemeine Hinweise	3
1.1	Handhabung	3
1.2	Installation	3
1.3	Erklärung	3
1.4	Reparaturen	3
2	Technische Daten	4
2.1	Umgebungsbedingungen	4
2.2	Mechanische Abmessungen	4
2.3	Elektrische Daten	4
3	Inbetriebnahme	5
3.1	Einbau	5
3.2	Lage der Jumper	5
3.3	Adreßbelegung	5
3.4	Interruptquellen	6
3.5	Spannungsversorgung DAC's	6
3.6	Index-Kanal	6
4	Hardwarebeschreibung	7
4.1	Steckerbelegungen	7
4.2	Inkrementalgeber	7
5	Programmierung	8
5.1	Prozeduren für den Motioncontroller LM628	8
5.1.1	Beschreibung der Funktionsbibliothek	8
5.1.2	PEARL-Beispiel	12

Revisionsliste:

Rev.	Datum	Na.	Änderung
1.0	11.07.2000	Ko	Erstellung

1 Allgemeine Hinweise

1.1 Handhabung

1. Lesen Sie bitte zuerst sorgfältig diese Dokumentation bevor Sie die Hardware auspacken und einschalten. Sie sparen Zeit und vermeiden Probleme.
2. Beachten Sie bitte die Vorsichtsmaßnahmen bei der Handhabung elektrostatisch gefährdeter Hardware.
3. Wenn die Hardware Batterien enthält, legen Sie sie nicht auf elektrisch leitfähige Unterlagen. Die Batterie könnte kurzgeschlossen werden und Schäden verursachen.
4. Achten Sie bitte darauf, daß der spezifizierete Temperaturbereich nicht verlassen wird.

1.2 Installation

1. Überprüfen Sie, ob alle Jumper entsprechend Ihrer Anwendung gesetzt sind.
2. Schalten Sie die Spannungsversorgung der externen Anschlüsse ab, bevor Sie eine Verbindung herstellen.
3. Wenn Sie sicher sind, daß alle Verbindungen korrekt installiert sind, schalten Sie die Spannungsversorgung ein.

1.3 Erklärung

Wir behalten uns das Recht vor, Änderungen, die einer Verbesserung der Schaltung oder des Produktes dienen, ohne besondere Hinweise vorzunehmen. Trotz sorgfältiger Kontrolle kann für die Richtigkeit der hier gegebenen Daten, Schaltpläne, Programme und Beschreibungen keine Haftung übernommen werden. Die Eignung des Produktes für einen bestimmten Einsatzzweck wird nicht zugesichert.

1.4 Reparaturen

Sollte das Produkt defekt sein, so senden Sie es bitte frei in geeigneter Verpackung mit folgender Beschreibung an uns zurück:

- Fehlerbeschreibung
- Trat der Fehler nur unter bestimmten Bedingungen auf?
- Was war angeschlossen?
- Wie sahen die angeschlossenen Signale aus?
- Garantiereparatur oder nicht?

2 Technische Daten

2.1 Umgebungsbedingungen

Umgebungstemperatur (Betrieb)	0-50° C
Umgebungstemperatur (Lagerung)	-20-85° C
rel. Luftfeuchte	max. 95%, nicht kondensierend
Höhe	-300m bis +3000m

2.2 Mechanische Abmessungen

Einfache Europakarte	100 x 160 mm
P-Bus Stecker	DIN 41612 C 64
Frontplatte	4 TE (=20,3mm)

2.3 Elektrische Daten

Adreß-, Daten- und Steuerleitungen siehe P-Bus Beschreibung.

Stromversorgung:	+5V \pm 5% / 700 mA +12-24 Volt extern
Eingangsspannung:	diff. 5 Volt oder 24 Volt
Eingangsfrequenz:	1 MHz
Ausgangsspannung:	\pm 10 Volt, gal. getrennt

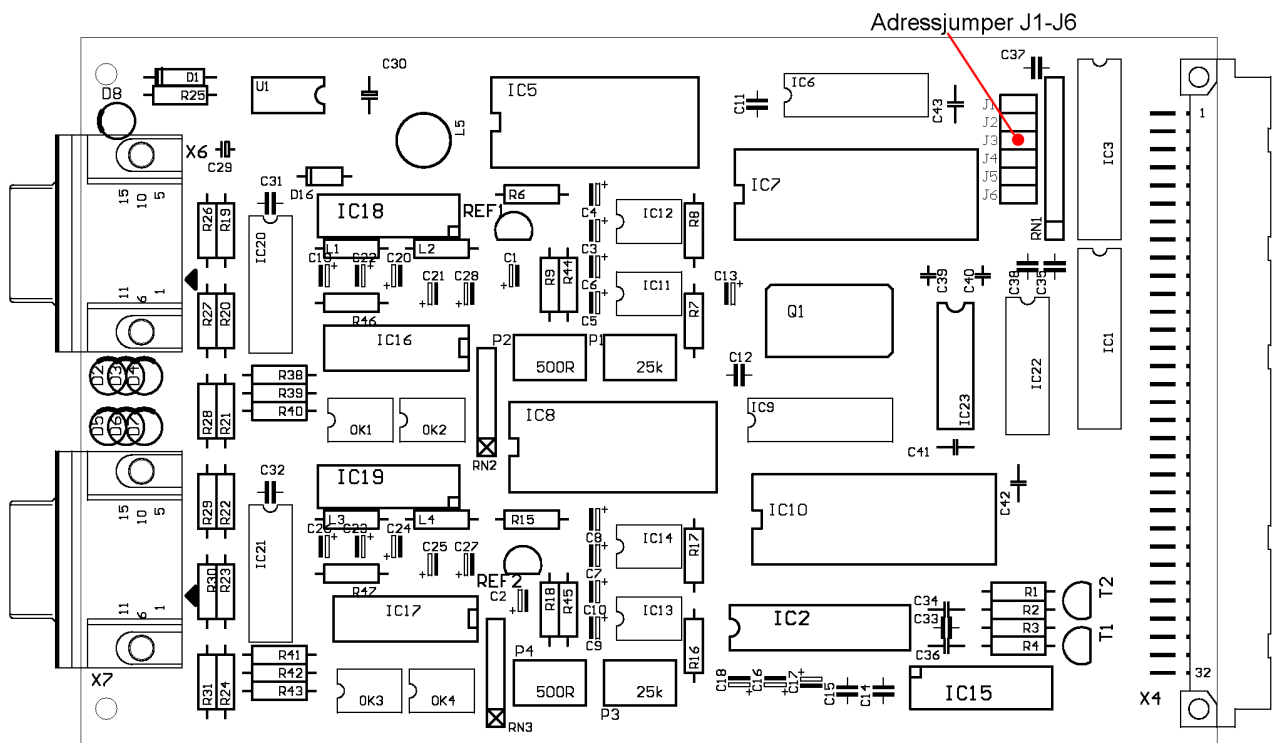
3 Inbetriebnahme

3.1 Einbau

Der MOTCO ist zum Einbau in EMV-dichte Gehäuse bestimmt. Die Verkabelung ist EMV-gerecht mit abgeschirmten Kabeln durchzuführen.

3.2 Lage der Jumper

Bestückungsseite:



3.3 Adreßbelegung

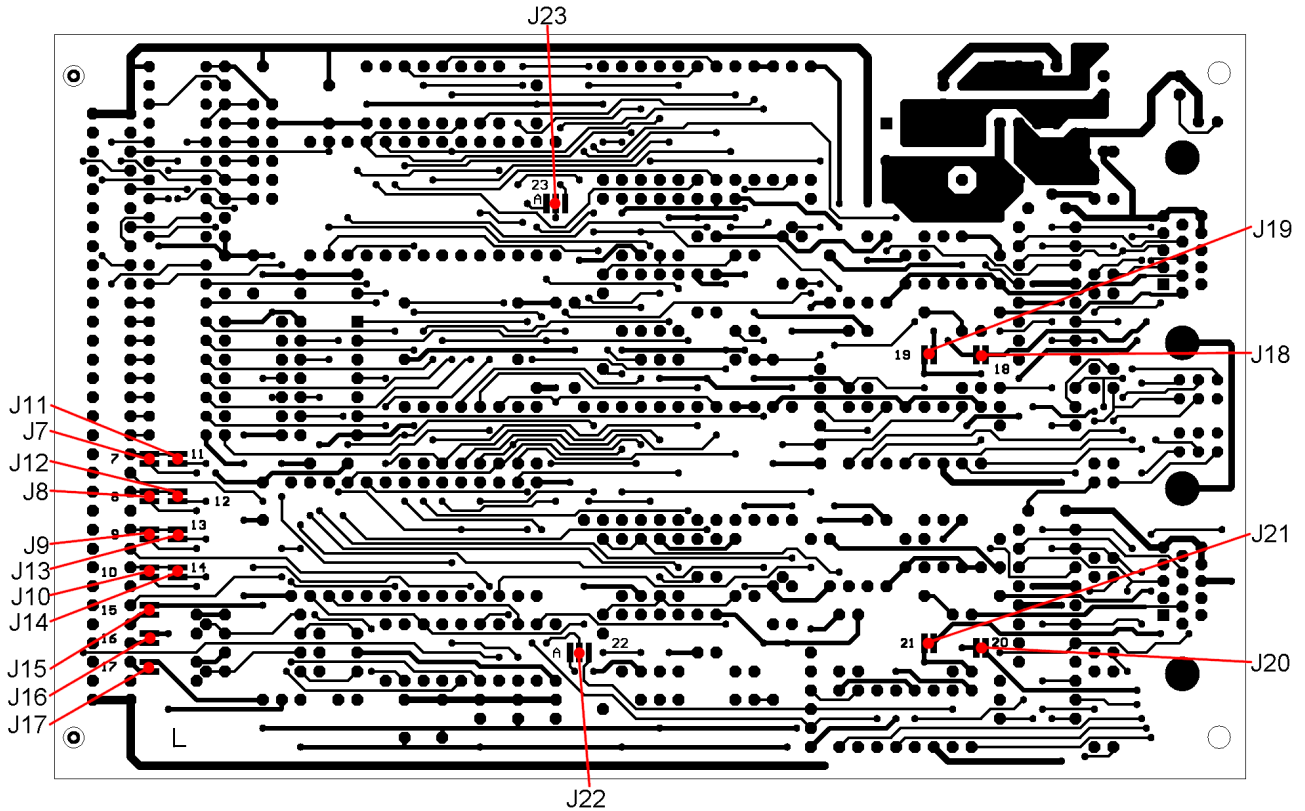
Von der MOTCO werden 4 aufeinander folgende Adressen belegt. Die Einstellung der Basis-Adresse erfolgt mit den Jumpern J1-J6. Ist der entsprechende Jumper gesteckt, wird das Adreßbit mit Low-Level verglichen, sonst mit High-Level.

Jumper J1-J6:

Adressbit	A2	A3	A4	A5	A6	A7
Jumper	J1	J2	J3	J4	J5	J6

Um z.B. die Basisadresse \$4 einzustellen, sind an Position J2, J3, J4, J5 und J6 Jumper zu stecken. Mit dieser Jumperung wird die Karte auch ausgeliefert. Kanal 1 belegt die ersten zwei Adressen, Kanal 2 die zweiten.

Lötseite:



3.4 Interruptquellen

Jeder LM628 kann einen Interrupt auslösen. Mit den Jumperfeldern J7-J14 kann der P-Bus Interrupt gewählt werden. Zur Zeit wird die Nutzung der Interrupts nicht empfohlen.

	EV1	EV2	EV3	EV4
Kanal 1	J7	J8	J9	J10
Kanal 2	J11	J12	J13	J14

3.5 Spannungsversorgung DAC's

Steht eine Versorgungsspannung mit ± 15 V zur Verfügung, können J15-J17 geschlossen werden. Der DC/DC-Wandler IC15 sowie C18 müssen dann entfallen.

3.6 Index-Kanal

Werden Inkrementalgeber ohne Index-Kanal eingesetzt, so sind die Eingangskanäle mit den Lötjumpfern J22 und J23 auf festen Pegel zu legen. Dazu sind die Lötbrücken auf der mit A gekennzeichneten Seite zu setzen.

4 Hardwarebeschreibung

4.1 Steckerbelegungen

Je Kanal steht eine 15-polige mini-SubD-Buchse zur Verfügung. Die Belegung ist für beide Kanäle gleich:

Pin	Belegung
1	+15 Volt (J19/J21)
2	Inkrementalgeber A+
3	Inkrementalgeber B+
4	Inkrementalgeber I+
5	GND
6	DAC Out +V
7	DAC GND
8	Inkrementalgeber A-
9	Inkrementalgeber B-
10	Inkrementalgeber I-
11	-15 Volt (J18/J20)
12	8-38 Volt
13	8-38 Volt
14	GND
15	GND

Die +8-38 Volt brauchen nur auf einer Buchse eingespeist werden, da die Anschlüsse beider Buchsen parallel geschaltet sind.

Die Anschlüsse 1 und 11 sind nur belegt, wenn der entsprechende Jumper geschlossen ist.

4.2 Inkrementalgeber

Es können Inkrementalgeber mit 2 Spuren und einem Index-Kanal angeschlossen werden. Das Vorhandensein von Impulsen wird über jeweils 3 Leuchtdioden D2 bis D4 für Kanal 1 bzw. D5 bis D7 für Kanal 2 angezeigt. Werden I-Geber mit differenziellen 5 Volt Ausgängen angeschlossen, so müssen IC20 bzw. IC21 bestückt sein. Beim Anschluß von 24 Volt Typen müssen stattdessen R26 bis R28 bzw. R29 bis R31 bestückt sein. Zur Versorgung der differenziellen Eingänge und Optokopfer ist eine Spannung von 8 bis 38 Volt nötig. Ihr Vorhandensein wird mit Leuchtdiode D8 angezeigt.

5 Programmierung

5.1 Prozeduren für den Motioncontroller LM628

Diese Beschreibung ersetzt nicht das intensive Studium des LM628 Handbuches! Die nachfolgenden Funktionen sind größtenteils ein 1:1-Abbild der LM-Command's.

Der 1. Parameter, der an die Funktion übergeben wird, ist die Hardwareadresse des gewünschten Kanals, jeder weitere Parameter entspricht der LM-Command-Struktur. Für Kanal 1 muß als Hardwareadresse die Basisadresse der MOTCO, für Kanal 2 die Basisadresse+2 übergeben werden.

Aus den meisten Funktionen erhält man den Zustand des Status-Bytes nach Ausführung der Funktion zurück.

Der Zugriff auf den LM628 ist nicht Multitasking-fest, d.h. es sollten nicht mehrere Tasks gleichzeitig versuchen, Kommandos abzusetzen.

5.1.1 Beschreibung der Funktionsbibliothek

Alle im LM-Handbuch beschriebenen Maßnahmen zur kontrollierten Bedienung des LM sind bereits in der Bibliothek enthalten, d.h. um das BUSY-Bit, die Einstellung auf 12-Bit-Wandler sowie das Timing brauchen Sie sich nicht mehr zu kümmern.

- Reset Auslösen des Reset

```
SPC Reset ENTRY( FIXED(31) /* HW-Adr */ )  
                RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

Es wird ein Reset ausgelöst. Wenn der Reset vom LM628 nicht akzeptiert wird, erfolgt die Ausgabe der Fehlermeldung:

```
WRONG COMMAND TIMING
```

und es wird ein Status von -1 zurückgegeben. War der Reset erfolgreich, wird auf 12 Bit D/A-Wandler geschaltet und die Home-Position gesetzt.

- DFH DeFine Home

```
SPC DFH ENTRY( FIXED(31) /* HW-Adr */ )  
            RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

Die aktuelle Position wird als "home" definiert, d.h. als absolut 0.

- MOFF Motor OFF

```
SPC MOFF ENTRY( FIXED(31) /* HW-Adr */ )  
           RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

Setzt Bit 8 im Trajectory-Control-Word zurück, d.h. der Wandlerausgang wird Null (Bremsen ohne Gegenstrom!).

- RDSIG ReaD SIGnal Register

```
SPC RDSIG ENTRY( FIXED(31) /* HW-Adr */ )  
                RETURNS( FIXED ) /* Sig-Reg */ GLOBAL ;
```

Auslesen des Signal-Registers.

- LFIL Load FILter

```
SPC LFIL  ENTRY( FIXED(31), /* HW-Adr */  
                FIXED(15), /* kp      */  
                FIXED(15), /* ki      */  
                FIXED(15), /* kd      */  
                FIXED(15)  /* il      */  
                ) RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

Laden der Filterparameter, entsprechend der im LM-Handbuch beschriebenen Funktion.

- UDF UpDate Filter

```
SPC UDF   ENTRY( FIXED(31) /* HW-Adr   */ )  
                RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

Nachdem die neuen Filter-Parameter geladen wurden, werden sie mit diesem Kommando aktiviert.

- LTRJ Load TRaJectory parameters

```
SPC LTRJ  ENTRY( FIXED(31), /* HW-Adr */  
                FIXED(15), /* CTRL   */  
                FIXED(31), /* Pos    */  
                FIXED(31), /* Acc    */  
                FIXED(31)  /* Vel    */  
                ) RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

Entspricht der im LM-Handbuch beschriebenen Funktion. Zu beachten ist, daß das CTRL-Word entsprechend dem Trajectory-Control-Word (s. Handbuch) initialisiert wird.

- LTRJM Load TRaJectory parameters Motor

```
SPC LTRJM ENTRY( FIXED(31), /* HW-Adr */  
                FIXED(15)  /* CNTL   */  
                ) RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

Es wird im Gegensatz zu LTRJ nur das CTRL-Word übergeben. Mit dieser Funktion kann z.B. ein schneller STOP ohne Parameter ausgelöst werden.

- RDDP ReaD Desired Position

```
SPC RDDP  ENTRY( FIXED(31) /* HW-Adr */ )  
                RETURNS( FIXED(31) ) /* Position */ GLOBAL ;
```

Entspricht der im LM-Handbuch beschriebenen Funktion.

-
- RDRP ReaD Real Position


```
SPC RDRP  ENTRY( FIXED(31)      /* HW-Adr  */ )
           RETURNS( FIXED(31) ) /* Position */ GLOBAL ;
```

 Entspricht der im LM-Handbuch beschriebenen Funktion.
 - RDDV ReaD Desired Velocity


```
SPC RDDV  ENTRY( FIXED(31)      /* HW-Adr  */ )
           RETURNS( FIXED(31) ) /* Geschw. */ GLOBAL ;
```

 Entspricht der im LM-Handbuch beschriebenen Funktion.
 - RDRV ReaD Real Velocity


```
SPC RDRV  ENTRY( FIXED(31) /* HW-Adr  */ )
           RETURNS( FIXED ) /* Geschw. */ GLOBAL ;
```

 Entspricht der im LM-Handbuch beschriebenen Funktion.
 - RDSUM ReaD integration-term SUMmation Value


```
SPC RDSUM ENTRY( FIXED(31) /* HW-Adr */ )
           RETURNS( FIXED ) /* I-Sum  */ GLOBAL ;
```

 Entspricht der im LM-Handbuch beschriebenen Funktion.
 - RDIP ReaD Index Position


```
SPC RDRP  ENTRY( FIXED(31)      /* HW-Adr  */ )
           RETURNS( FIXED(31) ) /* Index Pos. */ GLOBAL ;
```

 Entspricht der im LM-Handbuch beschriebenen Funktion.
 - STT STArT motion control


```
SPC STT   ENTRY( FIXED(31) /* HW-Adr  */ )
           RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

 Entspricht der im LM-Handbuch beschriebenen Funktion.
 - RIST Reset InterrupTs


```
SPC RIST  ENTRY( FIXED(31) /* HW-Adr  */ )
           RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

 Entspricht der im LM-Handbuch beschriebenen Funktion RSTI. (Mußte geändert werden, weil identisch mit Assembler-Command.)
 - SIP Set Index Position


```
SPC SIP   ENTRY( FIXED(31) /* HW-Adr  */ )
           RETURNS( FIXED ) /* Stat-Byte */ GLOBAL ;
```

 Entspricht der im LM-Handbuch beschriebenen Funktion.
-

-
- LPEI Load Position Error for Interrupt

```
SPC LPEI  ENTRY( FIXED(31), /* HW-Adr          */
                FIXED(15) /* Error-Pos. IR */
                ) RETURNS( FIXED ) /* Stat-Byte */ GLOBAL;
```

Entspricht der im LM-Handbuch beschriebenen Funktion.
 - LPES Load Position Error for Stopping

```
SPC LPES  ENTRY( FIXED(31), /* HW-Adr          */
                FIXED(15) /* Error-Pos. Not-Stop */
                ) RETURNS( FIXED ) /* Stat-Byte */ GLOBAL;
```

Entspricht der im LM-Handbuch beschriebenen Funktion.
 - SBPA Set BreakPoint Absolut

```
SPC SBPA  ENTRY( FIXED(31), /* HW-Adr          */
                FIXED(31) /* Pos.-Breakpoint abs */
                ) RETURNS( FIXED ) /* Stat-Byte */ GLOBAL;
```

Entspricht der im LM-Handbuch beschriebenen Funktion.
 - SBPR Set BreakPoint Relativ

```
SPC SBPR  ENTRY( FIXED(31), /* HW-Adr          */
                FIXED(31) /* Pos.-Breakpoint rel */
                ) RETURNS( FIXED ) /* Stat-Byte */ GLOBAL;
```

Entspricht der im LM-Handbuch beschriebenen Funktion.
 - MSKI MaSK Interrupts

```
SPC MSKI  ENTRY( FIXED(31), /* HW-Adr    */
                FIXED(15) /* IR-Maske */
                ) RETURNS( FIXED ) /* Stat-Byte */ GLOBAL;
```

Entspricht der im LM-Handbuch beschriebenen Funktion.
 - RDSTAT
Wurde nicht separat abgebildet, da in RDSIG enthalten.
-

5.1.2 PEARL-Beispiel

```
SetMotion : PROC
/*****/
( Achse CHAR(1)   ,
  Vorz  CHAR(1)   ,
  Acc   FIXED(15) ,
  Velo  FIXED(15)
) GLOBAL ;

      DCL (Null, VelH, AccH) FIXED(31) ;

      REQUEST Controller ;

      IF Achse EQ 'X' THEN
        IF Velo EQ 0 THEN
          Stat = LTRJM( X_Achse, TOFIXED('0400'B4) ) ;
          Stat = STT( X_Achse ) ;
        ELSE
          VelH = MF_rnd3155( EncLines_mm_X * 341E-6(55) *
                             Velo          * 65536(31)
                           ) ;
          AccH = MF_rnd3155( EncLines_mm_X * 116.281E-9(55) *
                             Acc          * 65536(31)
                           ) ;
          Stat = MOFF( X_Achse ); /* Wegen Acc.-Aenderung */

          IF Vorz EQ '+' THEN
            Stat = LTRJ( X_Achse, TOFIXED('182A'B4),
                        AccH, VelH, 0(31)
                      ) ;
          ELSE
            Stat = LTRJ( X_Achse, TOFIXED('082A'B4),
                        AccH, VelH, 0(31)
                      ) ;
          FIN;
          Stat = STT( X_Achse ) ;
        FIN;
      ELSE
        IF Achse EQ 'Y' THEN
          IF Velo EQ 0 THEN
            Stat = LTRJM( Y_Achse, TOFIXED('0400'B4) ) ;
            Stat = STT( Y_Achse ) ;
          ELSE
```

```

    VelH = MF_rnd3155( EncLines_mm_Y * 341E-6(55) *
                      Velo * 65536(31)
                      ) ;
    AccH = MF_rnd3155( EncLines_mm_Y * 116.281E-9(55) *
                      Acc * 65536(31)
                      ) ;
    Stat = MOFF( Y_Achse ) ; /*Wegen Acc.-Aenderung*/

    IF Vorz EQ '+' THEN
        Stat = LTRJ( Y_Achse, TOFIXED('182A'B4),
                    AccH, VelH, 0(31)
                    ) ;
    ELSE
        Stat = LTRJ( Y_Achse, TOFIXED('082A'B4),
                    AccH, VelH, 0(31)
                    ) ;
    FIN;
    Stat = STT( Y_Achse ) ;
FIN;
FIN;
FIN;
RELEASE Controller ;
END; /* PROC SetMotion -----*/ ;

INIT_LM : TASK
/******/
;

X_Achse = TOFIXED( '00F00004'B4 ) ; /* Basisadresse X-Achse MOCS */
Y_Achse = TOFIXED( '00F00006'B4 ) ; /* Basisadresse Y-Achse MOCS */

/* Controller-Reset */
Stat = Reset( X_Achse ) ;
Stat = MSKI( X_Achse, TOFIXED('0000'B4) ) ;
/* Reset Interrupts */
Stat = RIST( X_Achse, TOFIXED('0000'B4) ) ;
Stat = LFIL( X_Achse, X_KP, X_KI, X_KD, X_IL ) ;
Stat = UDF( X_Achse ) ;
Stat = LPEI( X_Achse, PEI_X ) ;
Stat = LPES( X_Achse, PES_X ) ;
Stat = DFH( X_Achse ) ;
Stat = MSKI( X_Achse, TOFIXED('0036'B4) ) ;
END; /* TASK Init_LM -----*/ ;

```

```

LM_X_Show : TASK
/******/
;
  PUT 'X_Achse:'
    TO Terminal BY A,SKIP;
  PUT 'Reglervorgabe-Position: ', RDDP( X_Achse )
    TO Terminal BY A,F(12,0),SKIP;
  PUT 'Realposition:          ', RDRP( X_Achse )
    TO Terminal BY A,F(12,0),SKIP;
  PUT 'Reglervorgabe-Geschw.: ', RDDV( X_Achse )
    TO Terminal BY A,F(12,0),SKIP;
  PUT 'Realgeschwindigkeit:   ', RDRV( X_Achse )
    TO Terminal BY A,F(12,0),SKIP;
  PUT 'Indexposition:         ', RDIP( X_Achse )
    TO Terminal BY A,F(12,0),SKIP;
  PUT 'Integrierer-Summe:     ', RDSUM( X_Achse )
    TO Terminal BY A,F(12,0),SKIP;
  PUT 'Status:                ', TOBIT( RDSIG( X_Achse ) )
    TO Terminal BY A,B1,SKIP;
END; /* TASK LM_X_Show -----*/ ;

```