Befehlsumfang MSR Steuerungen (Einführung mit Beispielen) und Wertebereiche der Encoder und Geschwindigkeitsangaben

Die Steuerung wird als **Black-Box** über eine serielle Schnittstelle (USB/RS232/UART) betrieben. Alle Befehle bestehen aus einem ASCII Zeichensatz, der von der Anlage intern verarbeitet wird und alle Abläufe steuert. Der Anwender hat keinerlei zusätzliche Programme oder dll's auf seinem Steuerrechner zu installieren.

Die Befehle sind in einer Tabelle zusammen gefasst und in einem Manual detaillierter beschrieben (siehe www.msr-engineering.de Produkte: DRS-500-ECS Link: Dokumentation im .pdf-Format).

Wenn ein Antrieb zu einem festgelegten Punkt fahren soll, dann wird die **Steuerung diesen Vorgang automatisch kontrollieren** und am Endpunkt die Achse halten, sowie eine Rückmeldung zum Host geben, dass die Aktion abgeschlossen ist.

Der Anwender muss hierfür die Positionsbestimmung berechnen, die sich aus der Auflösung des Weggebers, Getriebe und der Steigung der Welle ergeben. Die Positionsdaten sind in Hexadezimalen Werten in einem Kommando zu übergeben. Die sich aus der Anwendung ergebenden unterschiedlichen Betriebsarten (Proportionale Bewegungen, Drehmomentkonstante Bewegungen, Drehzahlkonstante Bewegungen und Positionierungen oder Joysticksteuerungen) lassen sich getrennt anwählen. Die verschiedenen Bewegungsabläufe werden danach hinsichtlich Position oder Geschwindigkeit übergeben, oder während der Bewegung die analogen Messdaten und aktuellen Positionen ausgelesen.

Für alle Geschwindigkeitsprofile steht eine Gleichungssystem zur Verfügung, dass den gewünschten Vorschub [mm/min] oder die Umdrehungszahl [Umdr/min] umrechnet und für die Befehle zur Anlage bereitstellt (siehe Beispiele unten).

Die Achsbegrenzungen können durch **Limit-Schalter** angezeigt werden. Auch hierfür gibt es eine **automatische Überwachung**, so dass die Achse nicht gegen die äußeren Begrenzungen gefahren werden kann. Es erfolgt eine automatische Bremsung, sobald die Schalter angesprochen haben und die Position kann nur in der entgegengesetzten Richtung wieder verlassen werden.

Für die Einstellung eines **Referenzpunktes** sind ebenfalls automatischen Bewegungsabläufe vorgesehen, für die entweder ein gesonderter Schaltkontakt oder einer der Limits verwendet werden kann. Alle Schalter lassen sich als Öffner oder Schließer definieren.

Die **Anlagenkonfiguration** wird durch ein mitgeliefertes Setup-Programm festgelegt. Dies wird in der Regel nur einmalig auf den Anwendungsfall bezogen angewendet. Die festgelegten Parameter werden dabei in einem EEPROM angelegt. Nach jedem Einschalten werden diese Informationen ausgelesen und stellen die Anlagenkonfiguration her (z.B. Art des Wegsensors (Inkrementeller Encoder, SSI oder Analogsensor für Sollwerte oder Istwerte), Motorverstärker für 1-Phasen oder 3-Phasen Motoren, welcher Eingang ist ein Limit-Schalter, Drehrichtungs-Informationen, etc.).

Die mitgelieferte **Bedienoberfläche für Test- und Schulungszwecke** enthält alle Befehle, die an die Anlage gesendet werden können. Hiermit kann der Anwender auch die Funktionen überprüfen und für seine eigenes Steuerprogramm überprüfen. Alle Kommandofehler werden durch detaillierte Fehlermeldungen an den Anwender zurückgegeben. Es lassen sich die Befehle auch manuell eingeben, um die Reaktion des Systems zu analysieren und damit mögliche Eingabefehler zu erkennen.

Eine Bedienoberfläche wird in der Regel aus einen INI-Datei mit den Verbindungsdaten des jeweiligen Arbeitsplatzes gestartet, bzw. erhält hiermit die Variablen Parametersätze für die Applikation.

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

Im laufenden Programm sind Übertragungsfehler praktisch ausgeschlossen und nicht bekannt.

Easy Motion Protocol

Das ***Easy Motion Protocol*** (***EMP***) definiert Nachrichten zur Ansteuerung von Motoren und zur Verarbeitung von analogen sowie digitalen I/O Daten.

Jede Nachricht des ***EMP*** besteht aus einem Nachrichtenkopf mit einer Länge von 2 Bytes und einem Nachrichtenkörper mit der Länge von sechs Bytes.

Bei serieller Übertragung wird jedes dieser acht hexadezimalen Bytes durch zwei (ASCII) Zeichen dargestellt. Da jeder Sende- und Antwortstring mit einem **EOF (0x1A = CHR(26))** abgeschlossen wird, ergibt sich eine **Gesamtlänge von 17 Zeichen (16 ASCII Daten + EOF)**.

Gültige Zeichen der hexadezimalen Bytes sind Ziffern von 0-9 und Großbuchstaben von A-F.

Das erste Byte gibt die Art der Nachricht an. Gerade Zahlen sind die Kommandos des PC's/SPS, ungerade Zahlen sind Antworten des Systems.

Aufbau der Datenstrings:

Datenstring	**CMD**	**DS**	**PROP**	**INFO**	**D3**	**D2**	**D1**	**D0**
	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
	MSB							LSB

Byte 0 = Datenwort 0
Byte 1 = Datenwort 1
Byte 2 = Datenwort 2
Byte 3 = Datenwort 3
Byte 4 = INFO = Success, Active, Ready und Fehlercode
Byte 5 = PROP = Eigenschaft
Byte 6 = DS = Modul-Nr. (D = bit [7…4]) und Antrieb-Nr. (S = bit [3…0])
Byte 7 = CMD = Kommandowort

EOF = 0x1A = CHR(26)

Ein EOF Zeichen bildet den Abschluss von jedem Sende- und Empfangs Kommandostring.

***MSR** ENGINEERING*
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

Beispiele aus einer VB-Bedienoberfläche:

Ein Kommandostring besteht aus 8 hexadezimalen Bytes und einem Abschlusszeichen CHR(26).

Die Bezeichnungen "Baugruppe" & "Achse" sind ein Byte mit Baugruppennummer und Achsennummer, die für die Nutzung von mehreren identischen Baugruppen, die über einen internen Bus verbunden sind. Bei einer Anlage wird immer 0x11 verwendet.

Chronologischer Befehlsumfang:

Auszug aus dem Programmierhandbuch mit Erweiterungen für den **DRS-500-ECS** Anlagentyp:

Die Anlage wird von der Bedienoberfläche mit den folgenden Befehlen gesteuert, wobei mehrere Befehle in bestimmten Arbeitsschritten verzugslos hintereinander angeordnet werden können.

Die Anlage wird diese Befehle sequenziell abarbeiten und jeden Befehl durch ein "Quittierungssignal" CHR(26) = 0x1A bestätigen.

Die gesamte Datenübertragung dauert bei 38.400 Baud ca. 10,8 ms (4,6 ms Empfang, 1,6 ms Verarbeitung in der Steuerung, 4,6 ms Rückantwort). Die interne Verarbeitung bleibt auch bei komplexeren Funktionen praktisch konstant und variiert nur unbedeutend. Bei der Analogmessung kann sich die Datenübertragungszeit allerdings auf 14 ms erhöhen, wenn die interne I2C-Busfrequenz von 96 kHz auf 15 kHz reduziert wird.

Einige Programme haben umfangreiche zusätzliche Parametereinstellungen. Dies Verfahren kann grundsätzlich vereinfacht werden, wenn der User nur bestimmte Einstellungen verwenden will. Eine gesonderte Codierung mit einem Kurzbefehl kann in diesen Fällen in der Steuerung von MSR Engineering verankert werden. Dann braucht nur dieser eine Kurzbefehl übertragen zu werden, und der gewünschte Parametersatz stellt sich mit dem Befehl ein.

***MSR** ENGINEERING*
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

00 : Programm-Version und Gerätenummer

Info, die Programmversions-Nummer und die Gerätenummer werden mit 2 Kommandos ausgelesen.

Syntax Programm:	00	DS	00					
Rückgabe:	DR	S-	50	0-	EC	_V	xx	.x

Syntax Geräte Nr.:	00	DS	01					
Rückgabe	DR	S-	50	0-	EC	_V	xx	.x

Funktion: Abfrage der Programmversion und Gerätenummer

Bemerkung: Die aktuelle Programmversion wird ausgegeben, z.B. DRS-500-ECS_V8.2 mit der Gerätenummer 650

02 : Mehrere Baugruppen EIN

Mehrere Anlagen sind über den internen-Kommunikations-Link zusammen geschaltet, und werden über eine einzige Schnittstelle gesteuert. Das erste Modul wird automatisch als Master identifiziert, alle weiteren Module werden chronologisch als Slaves behandelt.

Die Adressierung wird mit dem ersten Einschaltbefehl automatisch vergeben und richtet sich nach der jeweiligen Anordnung in der Reihenfolge des geschalteten Kommunikations-Link.

Das hat aber für die Programmierung nur die Bedeutung, dass das "DS" Byte die Kommandos unterscheidet. Alle folgenden Bytes sind bei allen Anlagen für die gleiche Funktion identisch.

Es wird lediglich in den Kommandos das DS-Byte unterschieden, indem das "D" die Modulnummer (1 bis F) repräsentiert, und das S-Byte spricht bei Anlagen mit Mehrfachachsen den jeweiligen Motor (1 bis 6) an. Damit können sich 11 bis F1 verschiedene Kombinationen ergeben, die bei einachsigen Anlagen damit insgesamt 15 Achsen steuern können.

Die Befehle werden dabei durch alle Module durchgereicht, bis das Modul mit der richtigen Adresse die Daten herausfiltert und entsprechend umsetzt sowie deren Rückantwort sendet.

Ohne diesen Einschalt-Befehl werden keine Kommandos angenommen.

Syntax :	02	DS	00					
Rückgabe:	03	DS		22				

Funktion: Baugruppe einschalten

Bemerkung: Die Baugruppen werden eingeschaltet und sind betriebsbereit.

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

02 : Einzelanlage EIN

Einzelanlage einschalten

Syntax :
Rückgabe:

02	DS	11					
03	DS		22				

Funktion: Einzelanlage / Modul einschalten

Bemerkung: Das Modul wird eingeschaltet und ist betriebsbereit.

04 : Anlage AUS

Anlage AUS, Anlage / Modul wird abgeschaltet und nimmt keine Befehle mehr entgegen.

Syntax :
Rückgabe:

04	DS						
05	DS		22				

Funktion: Modul ausschalten
Bemerkung: Das Modul wird funktionell abgeschaltet aber nicht strommäßig getrennt. Alle Funktionen sind in stand-by.

08 : Aktiv Halten Servomotor

STOP, die Achsbewegungen werden gestoppt. In Betriebsarten mit Regler wir die aktuelle Position aktiv gehalten, d.h. der Motor erzeugt sein aktives Drehmoment.

Syntax :
Rückgabe:

08	DS						
09	DS		22				

Funktion: Die Motorachse wird auf der Position aktiv gehalten.

Bemerkung: Die Anlage geht in den Positionsmodus und hält die Achse auf der aktuellen Position bis zum vollen Drehmoment.

0A : Fehlerabfrage

Fehlercode, der letzte Fehlercode wird ausgelesen.

Syntax :
Rückgabe:

0A	DS						
0B	DS		Fehlercode				

Funktion: N/A

Bemerkung: keine

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

20 : Einstellungen – Joystick Parameter

Einstellungen "02" Für den Joystick-Betrieb werden die ADC-Parameter und Betriebseinstellungen hergestellt.

Syntax :	20	DS	02	LOOPCycles	ADCTyp	ADCMode	MULTIPlier	HYSTerese
Rückgabe:	21	DS		22				

Funktion: Betriebsparameter einstellen für den Joystick-Betrieb

Bemerkung: Der Joystickbetrieb läuft nach der Initialisierung ohne Kommandos von außen auf die Steuerung. Der geschwindigkeitswert wird dann bei jedem Programm-Durchlauf gewonnen.

Die folgenden Daten werden zugewiesen:

LOOPCycles = Anzahl der Programmschleifen nachdem ein Messwert gewonnen wird.
"01" = 10 ms, "10" = 100ms, "20 = 200ms, "30" = 300ms,
ADCTyp = verwendeter ADC Typ = "04" (ADS1112 als interner ADC)
ADCMode = ADC Messung = "00" = bipolar, "01" = unipolar
MULTIPlier = Verstärkungsfaktor des Analogsignals = "01" = Faktor 1
HYSTerese = 0-Punkt Abweichung, bevor Joystick-Ausschlag registriert wird (z.B. sind "40" = 64mV).

20 : Einstellungen – Aktuelle Position löschen

Setze Position Null, die aktuelle Position wird auf null gesetzt.

Syntax :	20	DS	05	01		00	00	00
Rückgabe:	21	DS		22		PosMSB	PosMID	PosLSB

Funktion: Die aktuelle Position wird gelöscht

Bemerkung: Für relatives Positionieren ist dieser Befehl gut anwendbar. ACHTUNG! Der Regler wird abgeschaltet und die Betriebsart muss erneut gesetzt werden.

20 : Einstellungen – Aktuelle Position setzen

Setze Position, die aktuelle Position wird auf einen gewünschten Wert eingestellt.

Syntax :	20	DS	05	01		PosMSB	PosMID	PosLSB
Rückgabe:	21	DS		22		PosMSB	PosMID	PosLSB

Funktion: Die neue Position wird definiert.

Bemerkung: ACHTUNG! Der Regler wird abgeschaltet und die Betriebsart muss erneut gesetzt werden.

***MSR** ENGINEERING*
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

20 : Einstellungen – Abtastrate

Syntax :
Rückgabe:

20	DS	19				MSB	LSB
21	DS		22				

Funktion: Abtastrate des Reglers einstellen. MSB/LSB Wertebereich [00 … FF].

Bemerkung: Werte für Abtastraten: 0xffff = 5 ms, 0x3600 = 1 ms, 0x1800 = 0,5 ms.

Weitere Einstellungen können bei den Geschwindigkeiten im PV / IV oder TP-Betrieb erfolgen.

22 : Motor Enable

Motor Enable, Setze die Betriebsart, die für die folgenden Bewegungsprofile gilt.

In der Betriebsart "Joystick" sind noch der ADC-Kanal und die Unter-Betriebsart anzugeben, die bei der Auslenkung des Joysticks verwendet werden sollen. Der Prozessor erhält dann seine Geschwindigkeitsinformationen von dem zugewiesenen ADC-Kanal, der zuvor noch weitere Parameter-Einstellungen erfordert.

Diese Parameter sind LOOPCycles + ADCTyp + ADCMode + MULTIPlier + HYSTerese, deren Bedeutung im Abschnitt "Parameter" (20 DS 02) erläutert sind.

Syntax :
Rückgabe:

22	DS	Betrieb	ADC-Kanal	Sub-Betrieb			
23	DS		22	PosMSB	PosMIDH	PosMIDL	PosLSB

Funktion: Motor aktivieren

Bemerkung: Als Betriebs-Mode stehen zur Verfügung:

02 = PV Betrieb (Proportionaler Geschwindigkeits-Betrieb)
03 = IV- Betrieb (Integrierter Geschwindigkeits-Betrieb)
04 = PWM Betrieb (konstanter Drehmoment-Betrieb)
05 = Joystick Betrieb EIN (Drehzahlbetrieb mit Joystick-Betrieb)
06 = Joystick Betrieb AUS
07 = TPWM Betrieb (Positionier-Betrieb und konstanter Drehmoment-Betrieb)

Im Joystick-Betrieb müssen noch weitere Befehle vorangestellt werden, da der Analogkanal und die Unter-Betriebsart (PWM / PV / IV) für den Drehzahl-Betrieb noch zugeordnet werden müssen.

Als Analogkanäle stehen bipolar Kanal 1 mit "01" und bipolar Kanal 2 mit "02" zur Verfügung.

Als Unter-Betriebsart wird die Geschwindigkeitsprofile PWM, PV oder IV bezeichnet mit der Codierung:

02 = PV (starke Drehmomentsteuerung)
03 = IV (Drehzahlkonstante Steuerung)
04 = PWM (leichte Drehmomentsteuerung)

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

24 : Motor Disable

Syntax :
Rückgabe:

24	DS						
25	DS		22				

Funktion: Motor abschalten

Bemerkung: keine

26 : Referenzfahrt

Referenzfahrt, der Referenzschalter wird angefahren, reversiert und hält an der Referenzposition.

Steht die Achse in dem Referenzschalter, fährt die Achse im Schleichgang solange, bis der Referenzschalter ausgelöst hat.

Der Referenzschalter kann im Setup den Eingängen frei zugeordnet werden. Die Bewegungsrichtung des Antriebes ist dabei anzugeben, damit die Drehrichtungen richtig zugeordnet werden. Der Schalter kann als Öffner oder Schließer arbeiten, was auch im Setup anzugeben ist.

Syntax :
Rückgabe:

26	DS					Vfast	Vslow
27	DS		55 / 22	00	PosMSB	PosMID	PosLSB

Funktion: Referenzfahrt der Einzelachse

Bemerkung: Solange wie die Achse noch nicht Positioniert hat, wird ein "Busy" (55) zurückgegeben. Wenn der Referenzpunkt erreicht wurde kommt selbständig ein "Ready" (22).

Die Fahrt in Richtung aus den Referenzpunkt wird mit Vfast angefahren, während die Bewegung vom Referenzschalter mit Vslow gefahren wird.

ACHTUNG! Auch bei Überlaufen des Referenzschalters wird die Achse umgesteuert und bleibt erst am Referenzpunkt stehen.

2A : Move – Servomotor TPWM (01=TP / 02=PWM)

Der Motor fährt zu der angegebenen Position mit der Geschwindigkeit Vmax, wenn vorausgehend die TPWM-Betriebsart eingestellt wurde und die Codierung "01" eingetragen wird.

Syntax :
Rückgabe:

2A	DS	01	Vmax	PosMSB	PosMIDH	PosMIDL	PosLSB
2B	DS		22	PosMSB	PosMIDH	PosMIDL	PosLSB

Funktion: Die Achse wird zu der spezifizierten Position gefahren.

Bemerkung: Die Geschwindigkeit kann nach den Geschwindigkeits-Regeln mit der variablen Abtastrate eingestellt werden (siehe unten in 2A).

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

Der Motor fährt mit konstantem Drehmoment, wenn vorausgehend die TPWM-Betriebsart eingestellt wurde und die Codierung "02" eingetragen wird.

Syntax :
Rückgabe:

2A	DS	02	Vpwm				
2B	DS		22	PosMSB	PosMIDH	PosMIDL	PosLSB

Funktion: Die Achse wird mit konstantem Drehmoment gefahren wie sonst auch im PWM-Mode.

Bemerkung: keine

2A : Move – Servomotor PWM / PV / IV

Der Motor beginnt sich zu drehen mit der Geschwindigkeit Vmax.

Die Beschleunigung A wird bei diesen Geräten nicht verarbeitet und muss, falls Erforderlich über die PC-Schnittstelle mit ansteigenden Geschwindigkeiten übertragen werden.

Syntax :
Rückgabe:

2A	DS		V				
2B	DS		22	PosMSB	PosMIDH	PosMIDL	PosLSB

Funktion: Die Achse wird zu der spezifizierten Position gefahren. Die Beschleunigung A wird nicht berücksichtigt.
Bemerkung: siehe Berechnung unten, mit Geschwindigkeits-Regeln und variablen Abtastraten.

2A : Fast STOP

Der Motor wird sehr schnell abgebremst. Er kommt aber nur bei einer selbsthemmenden Achse zum Stehen.

Syntax :
Rückgabe:

2A	DS	03					
2B	DS		22	PosMSB	PosMIDH	PosMIDL	PosLSB

Funktion: Die Achse wird sofort abgebremst und kommt zum STOP. Danach ist der Motor stromlos und muss erneut mit seiner Betriebsart geladen werden.
Bemerkung: die Bewegung wird in beide Richtungen maximal abgebremst.

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

2C : Position lesen

Die aktuelle Position wird ausgelesen. Dieser Befehl kann jederzeit angewendet werden.

Syntax :

	2C	DS						
Rückgabe:	2D	DS		22	PosMSB	PosMIDH	PosMIDL	PosLSB

Funktion: Die aktuelle Position lesen. Die Abfrage kann jederzeit erfolgen.

Bemerkung: Es ist jedoch bei Positionierungen zu beachten, welche Vorgeschichte an Befehlen vorliegt, bzw. ob auf das Signal "Ready" gewartet wird.

40 : I/O Read

Der Zustand der digitalen Eingänge wird gelesen. der Zustand der Eingänge wird direkt angezeigt. Bei "EIN" liegt 24V am Eingang an.

Syntax :

	40	DS	01					
Rückgabe:	41	DS	Limits	22			I4 I3	I2 I1

Funktion: Lesen der digitalen Eingänge. "1" = EIN = 24V, "0" = AUS = 0V.

Bemerkung: keine

42 : I/O Write

Die digitalen Ausgänge werden geschrieben. Bei "EIN" liegt 24V am Ausgang an.

Syntax :

	42	DS	01				O4 O3	O2 O1
Rückgabe:	43	DS	Limits	22			11...00	11...00

Funktion: Schreiben der digitalen Ausgänge. "1" = EIN = 24V, "0" = AUS = 0V.

Bemerkung: keine

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

50 : ADC Read

Analog Lesen, der analoge Kanal wird gelesen.

Syntax :
Rückgabe:

50	DS	Typ		Mode			
51	DS		22			DatLSB	DatMSB

Funktion: ADC lesen mit Spezifizierung des ADC-Typs.

Typ: "04" = ADS1112 – mit Auflösung von wahlweise 12-bit (250 SPS) bis 16-bit (15 SPS).

Mode: Gibt den Kanal, die Auflösung und den Verstärkungsfaktor an, wobei im bipolaren Betrieb "x" den Kanal1 mit "0" und den Kanal2 mit "2" anwählt. Im unipolaren Betrieb ist Kanal1 mit "4", Kanal2 mit "6" und Kanal3 mit "2".

x0 = 12-bit, Faktor 1	x1 = 12-bit, Faktor 2	x2 = 12-bit, Faktor 4	x3 = 12-bit, Faktor 8
x4 = 14-bit, Faktor 1	x5 = 14-bit, Faktor 2	x6 = 14-bit, Faktor 4	x7 = 14-bit, Faktor 8
x8 = 15-bit, Faktor 1	x9 = 15-bit, Faktor 2	xA = 15-bit, Faktor 4	xB = 15-bit, Faktor 8
xC = 16-bit, Faktor 1	xD = 16-bit, Faktor 2	xE = 16-bit, Faktor 4	xF = 16-bit, Faktor 8

Bemerkung:

Die Umschaltung zwischen unipolaren und bipolaren Betrieb muss außerdem noch durch einen JUMPER im Gerät manuell umgesteckt werden.

Der Messbereich wird bei den Faktoren >1 entsprechend niedriger. Im bipolaren Mode ist bei Faktor 1 der Messbereich ±2V, bei Faktor 2 ist der Messbereich ±1V, bei Faktor 4 ist der Messbereich ±500mV, bei Faktor 8 ist der Messbereich ±250mV. Im unipolaren Betrieb gelten nur die positiven Werte. Andere Verstärkungsfaktoren gibt es nicht.

Der Verstärkungsfaktor wird als "PGA" in die Umrechnung eingehen mit den Faktoren 1, 2, 4 oder 8.

Die Auflösung kann zwischen 12/14/15/16-Bit gewählt werden. Der Umrechnungsfaktor "MinCode" ist dabei wie folgt:

12-Bit, MinCode = 2048 - 14-Bit, MinCode = 8192 - 15-Bit, MinCode = 16384 - 16-Bit, MinCode = 32768

Die folgende Umrechnung ist im PC erforderlich, wobei die "DezZahl" die umgerechnete Dezimalzahl des hexadezimalen Wertes aus der Anlage darstellt. Die Anlage liefert mit Byte 7 den LSB-Messwert und mit Byte 8 den MSB-Messwert.

Die Umrechnung auf mV erfolgt dann mit der folgenden Formel (siehe Datenblatt ADS1112):

```
If DezZahl < 32768 Then                          'Datenumrechnung vom 2-er Komplement
   PosWert = 2048 * DezZahl / (MinCode * PGA)    'Positiver analoger Wert
Else
   NegWert = -2048 * (65535 - DezZahl) / (MinCode * PGA)   'Negativer analoger Wert
End If
```


MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

54 : RTC

Real-Time-Clock (Datum und Uhrzeit) einstellen oder lesen.

RTC lesen

Syntax :

54	DS	00		WT	HH	MM	SS
55	DS		22				

Rückgabe:

54	DS	01		TT	MM	JJ	JJ
55	DS		22				

Funktion: Datum und Uhrzeit lesen

Bemerkung: -

```
'RTC LESEN
Sendestring.Text = setrtc & Baugruppe & Achse & "000000000000"
SendeCommDatWaitEOF()          'WT HH:MM:SS
Label1.Text = lbl5.Text + "  " + lbl6.Text + ":" + lbl7.Text + ":" + lbl8.Text

Sendestring.Text = setrtc & Baugruppe & Achse & "010000000000"
SendeCommDatWaitEOF()          'TT:MM:JJJJ
Label1.Text = lbl8.Text + "." + lbl7.Text + "." + lbl5.Text + lbl6.Text + "    " + Label1.Text

Tag.Text = Microsoft.VisualBasic.Left(Label1.Text, 2)        'Tag
Monat.Text = Mid(Label1.Text, 4, 2)                          'Monat
Jahr.Text = Mid(Label1.Text, 7, 4)                           'Jahr
Wochentag.Text = Mid(Label1.Text, 14, 2)                     'Wochentag
Uhrzeit.Text = Mid(Label1.Text, 18, 8)                       'Std:Min:Sek
```

RTC (Uhrzeit und Datum) schreiben

Syntax :

54	DS	02		WT	HH	MM	SS
55	DS		22				

Rückgabe:

54	DS	03		CC	JJ	MO	TT
55	DS		22				

Legende: WT=Wochentag, HH=Stunden, MM=Minuten, SS=Sekunden
CC=Jahrhundert, JJ=Jahr, MO=Monat, TT=Wochentag

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

44 : EEPROM Read/Write über I2C-Schnittstelle

Ein EEPROM wird über die I2C-Schnittstelle des HMI-Anschlusses byteweise beschrieben und gelesen. Jeder Befehl überträgt nur ein Datenbyte.

EEPROM Lesen, das externe EEPROM der HMI-Schnittstelle wird über eine I2C-Schnittstelle ausgelesen.

Syntax :
Rückgabe:

44	DS	08	MemoMSB	MemoLSB	EEPROMAdr		
45	DS		22				Data

Funktion: Über den I2C Bus eine Adresse lesen.

Bemerkung: Die 2-Byte Speicheradresse wird in "MemoMSB" und "MemoLSB" angegeben und die Adresse des externen EEPROMs mit 1-Byte in "EEPROMAdr". Das datenbyte wird in der Antwort mit "Data" gesendet.

Beispiel:

```
Sendestring.Text = EEPROMDaten & Baugr & Achse & "08" + MemorY + EEpromAdr + "0000"
SendeCommDatWaitEOF()
```

EEPROM Schreiben, das externe EEPROM der HMI-Schnittstelle wird über eine I2C Schnittstelle beschrieben.

Syntax :
Rückgabe:

44	DS	09	MemoMSB	MemoLSB	EEPROMAdr		MemoDat
45	DS		22				

Funktion: Über den I2C Bus eine Adresse schreiben.

Bemerkung: Die 2-Byte Speicheradresse wird in "MemoMSB" und "MemoLSB" angegeben und die Adresse des externen EEPROMs mit 1-Byte in "EEPROMAdr". Das zu schreibende Byte wird in "MemoDat" geschrieben.

Beispiel:

```
TextSendestring.Text = EEPROMDaten & Baugr & Achse & "09" + MemorY + EEpromAdr + "00" + MemDat
SendeCommDatWaitEOF()
```


MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

Beispiel für Programmierung von Bewegungsprofilen u.a.

Beispiel 1:

Die folgenden Funktionen werden der Anlage mitgeteilt um die Achse mit konstanten Drehmomenten zu drehen:

1.	Anlage einschalten	➔02DS110000000000	
2.	Betriebsart mit konstanten Drehmoment festlegen	➔22DS040000000000	
3.	Drehmoment in Höhe von "10" auf den Motor geben	➔2ADS001000000000	RL10
4.	Drehmoment in Höhe von "15" auf den Motor geben	➔2ADS001500000000	RL15
5.	Drehmoment in Höhe von "F0" auf den Motor geben	➔2ADS00F000000000	LL10

Beispiel 2:

Die Achse in Betriebsart IV mit nicht-skalierten Geschwindigkeiten V = 2/5/-2/-5 drehen:

1.	Betriebsart IV auswählen	➔22DS030000000000	
2.	Geschwindigkeit von 2 übergeben	➔2ADS000200000000	RL2
3.	Geschwindigkeit von 5 übergeben	➔2ADS000500000000	RL5
4.	Geschwindigkeit von -2 übergeben	➔2ADS00FE00000000	LL2
5.	Geschwindigkeit von -5 übergeben	➔2ADS00FB00000000	LL5

Die Achse in Betriebsart IV mit skalierten Geschwindigkeiten V = 2/-2/5 Umdr/min drehen mit Getriebeuntersetzung = 14:1, Spindelsteigung = 5mm/Umdr:

6.	Betriebsart IV auswählen	➔22DS030000000000	
7.	Abtastrate berechnen und übergeben	➔20DS190000001A28	Abtastrate einstellen
8.	Geschwindigkeit von 2 Umdr/min	➔2ADS000100000000	RL2 U/min
9.	Geschwindigkeit von -2 Umdr/min	➔2ADS00FF00000000	LL2 U/min
10.	Abtastrate berechnen und übergeben	➔20DS190000000A76	Abtastrate einstellen
11.	Geschwindigkeit von 5 Umdr/min	➔2ADS000100000000	RL5 U/min

Die Abtastraten variierten im Bereich 5-6 kHz, wobei der Geschwindigkeitswert mit "01" berechnet wurde.

Beispiel 3:

Die aktuelle Position während der Achsbewegung lesen.

1.	Position lesen	➔2CDS000000000000
2.	Position lesen	➔2CDS000000000000

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

Beispiel 4:

Die Achse auf eine Position 500 mm mit einer Geschwindigkeit von 10 mm/min positionieren und dann auf Position 200 mm mit 5 mm/min positionieren.

Getriebeuntersetzung = 14:1, Spindelsteigung = 5mm/Umdr, Encoder = 1000 Inkr/Umdr:

1. Betriebsart TPWM festlegen	➔22DS070000000000
2. Geschwindigkeit von 10 mm/min berechnen	Abtastrate = 1A28 mit V = 01
3. Position von 500 mm berechnen	500 mm = 100.000d = 186A0h Inkremente
4. Abtastfrequenz an Anlage senden	➔20DS190000001A28 Abtastrate
5. Bewegungskommando an Anlage senden	➔2ADS0001000186A0 V und Posi
6. warten, bis Achse positioniert hat	Anlage sendet Info-byte "22"
7. Position lesen (wenn gewünscht)	➔2CDS000000000000
8. Geschwindigkeit von 5 mm/min berechnen	Abtastrate = 3450 mit V = 01
9. Position von 200 mm berechnen	200 mm = 40.000d = 9C40h Inkremente
10. Abtastfrequenz an Anlage senden	➔20DS190000003450 Abtastrate
11. Bewegungskommando an Anlage senden	➔2ADS000100009C40 V und Posi
12. warten, bis Achse positioniert hat	Anlage sendet Info-byte "22"
13. Position lesen (wenn gewünscht)	➔2CDS000000000000

MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

Beispiel 5:

Joystick Zuschalten und wieder Abschalten

1. I2C-Systemfrequenz auf 15 kHz einstellen ➔ 44 DS 11 ….. ADCFreq …
2. ADC Parameter einstellen ➔ 20 DS 02 ……
3. Analogen Kanal 1 lesen ➔ 50 DS ADCTyp … ADCRange
4. Joystick EIN mit Sub-Betriebsart einstellen ➔ 22 DS 05 ADCKanal BetrArt …..
5. Wenn IV, dann Regler einstellen ➔ 20 DS 19 …… AbtastFreq
6. Wenn IV, dann Regler vom PC aus initialisieren ➔ 2A DS 00 01….

7. Joystick AUS ➔ 22 DS 06 ….

Joystick Zuschalten (1-5):

```
'I2C Systemfrequenz auf 15kHz einstellen
EEPROMDaten = "44" :  ADCFreq = "26"
TextSendestring.Text = EEPROMDaten & Baugruppe & Achse & "11" + "000000" + ADCFreq + "00"
SendeCommDatWaitEOF()

'ADC Parameter einstellen mit Cmd: 20 DS 02 Cycle ADCTyp Mode Multi Hyst
Parameter = "20" : LOOPCycles = "0C" : ADCTyp = "04" : ADCMode = "00" : MULTIPlier = "01" : HYSTerese = "40"
TextSendestring.Text = Parameter & Baugruppe & Achse & "02" + LOOPCycles + ADCTyp + ADCMode +
MULTIPlier + HYSTerese
SendeCommDatWaitEOF()

'Analog Lesen CH1
AnalogLesen = "50" : ADCTyp = "04" :  ADCRange = "00"
TextSendestring.Text = AnalogLesen & Baugruppe & Achse & ADCTyp & "00" & ADCRange & "00" & "00" & "00"
SendeCommDatWaitEOF()

'Betriebsart JOYSTICK mit Sub-Betriebsart PWM, PV oder IV zuschalten (2A DS 05 ADCH BETR…..)
MotorEnable = "22" : ADCKanal = "01" : BetriebsArt = "03"
TextSendestring.Text = MotorEnable & Baugruppe & Achse & "05" + ADCKanal + BetriebsArt + "000000"
SendeCommDatWaitEOF()

'MotorMove wird intern erzeugt! wenn aber im IV-Betrieb, dann MotorMove vom PC mit V=01 initiieren.
'Bei den Sub-Betriebsarten PV und PWM nicht erforderlich!
If BetriebsArt = "03" Then
   Parameter = "20" : AbtastFrequ =  "C350" : MotorMove = "2A" 'mit "00 01"
    TextSendestring.Text = Parameter & Baugruppe & Achse & "19" + "00" + "0000" + AbtastFrequ
    SendeCommDatWaitEOF()
    TextSendestring.Text = MotorMove & Baugruppe & Achse & "00" & "01" & "00000000"
    SendeCommDatWaitEOF()
   'Der Motor fährt kurz an und bleibt dann stehen.
End If
'Poti-Betrieb ist jetzt zugeschaltet. Alle Sollwerte werden vom Joystick-Potentiometer erhalten.
```

Joystick Abschalten (6):

```
'Poti-Betrieb Abschalten = Joystick AUS (2A DS 06 …..)
MotorEnable = "22"  'mit Codierung "06" = AUS
TextSendestring.Text = MotorEnable & Baugruppe & Achse & "06" + "0000000000"
SendeCommDatWaitEOF()
```


MSR ENGINEERING
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

Berechnung der Abtastrat mit Vorgabe von Geschwindigkeit/Vorschub

Mit den folgenden schematisch dargestellten Programmteilen werden die Abtastraten [Reload Value] und Geschwindigkeiten [00h-7Fh] bei vorgegebenen Achsen-Geschwindigkeiten [mm/min] und Vorschübe [mm/min] berechnet.

Bei Anwendung mit einer vorgegebenen Abtastrate ist die Genauigkeit geringer als bei einer variablen Abtastrate, die sich 16-Bit genau einstellt.

vRL und vLL sind die zu übertragenden Geschwindigkeiten für Rechtslauf und Linkslauf, während RelVal die Abtastfrequenz der Steuerung einstellt.

Die Programme sind von Visual Basic abgeleitet und müssen entsprechend der verwendeten Hochsprache umgesetzt werden.

Die Dezimalwerte für "RelVal" und "RelValDezimal" sind wie folgt:

Frequenz	**RelVal_dez**	**RelVal_hex**
250 Hz	50.000	0xC350
400 Hz	31.250	0x7A12
500 Hz	25.000	0x61A8
1 kHz	12.500	0x30D4
2 kHz	6.250	0x186A
3 kHz	4.167	0x1046
4 kHz	3.125	0x0C35
6 kHz	2.083	0x0823

Der Reload-Wert wird errechnet mit:

$$RelVal = TimeOutPeriod * \frac{Systemfrequenz}{4}$$

$$TimeOutPeriod = \frac{1}{Abtastfrequenz}$$

Die Systemfrequenz beträgt 50 MHz. Alle übrigen Werte ergeben sich aus der mechanischen Untersetzung (Getriebe), der Spindelsteigung und dem verwendeten Drehgeber hinsichtlich Inkr/Umdrehung. Die Faktoren 4 ergeben sich aus der Quadratur des Encodersignals und dem Vorteiler der Systemfrequenz des Prozessors. Die 60 ergibt sich aus 60 sek/min. Die Grundfrequenz liegt im Bereich 6 bis 54. Die programmierte Linkslauf Geschwindigkeit ist vLL = 255 – vRL.

Berechnung der Geschwindigkeit und Abtastrate:

```
 Public Sub BerechnungUpDownIV()        'BERECHNUNG RELOADVALUE UND GESCHWINDIGKEIT
'INPUT: Vorschub, MinAbtastRate
'OUTPUT: Geschwindigkeit, AbtastFrequ

Grundfrequenz = MinAbtastRate
Geschwindigkeit = "128"

Do While Val(Geschwindigkeit) > 127
   Geschwindigkeit = (Vorschub * Untersetzung / (Spindelsteigung * Grundfrequenz))
   Grundfrequenz = Val(Grundfrequenz) + 1
Loop

Geschwindigkeit = (Int(Val(Geschwindigkeit)))
AbtastFrequ = ((60 * (Systemfrequenz) * 1000000.0 * (Geschwindigkeit)) / ((Encoderauflösung) * 4 * (Vorteiler) *
((Vorschub) * (Untersetzung) / (Spindelsteigung))))          'ReloadValue in dezimal

If Val(AbtastFrequ) > 65535 Then
   'MsgBox("Werteüberschreitung für die berechnete Abtastrate") : Messung = False
End If

End Sub
```


***MSR** ENGINEERING*
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de

Datenübergabe an die Steuerung für Abtastfrequenz und Fahrdaten:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

        AbtastFrequ = RelVal.Text          'Abtastfrequenz einstellen mit Parameter in [HEX]
        Sendestring.Text = Parameter & Baugruppe & Achse & "19" + "000000" + AbtastFrequ
        SendeCommDatWaitEOF()              'Einstellung der Abtastfrequenz

      'vRL = Fahrdaten mit MotorMove in [HEX]
        Sendestring.Text = MotorMove & Baugruppe & Achse & BeschlA & vRL & "00000000"
        SendeCommDatWaitEOF()              'Übertragung der Fahrdaten mit Geschwindigkeit

    End Sub
```

Alle übrigen Befehle bestehen aus einfachen Datensätzen, die keiner detaillierteren Berechnung unterliegen. Die Berechnung der Geschwindigkeit kann auch einfacher mit konstanter Abtastfrequenz erfolgen, dabei entfällt der ganze Rechnungsgang für den RelVal.

Durch die variable Abtastfrequenz wird der Dynamikbereich für die Geschwindigkeit erhöht und die Genauigkeit verbessert.

Wertebereiche der Encoder und Geschwindigkeitsangaben

Die Standard Positionsdaten sind im 25-byte Wertebereich festgelegt (25-bit SSI-Encoder), die Geschwindigkeitsdaten im Bereich 0x00 bis 0x7F (Rechtslauf) und 0x80 bis 0xFF (Linkslauf).

Die Auflösung ist bei den Geschwindigkeiten sehr viel höher, da die Abtastrate mit 16-bit Daten auf die jeweilige Geschwindigkeit in [mm/min] oder die Drehzahlen in [Umdr/min] abgestimmt werden.

Alle Daten werden in Hexadezimalen Zahlen an die Steuerung übergeben.

01.00.00.00h 01.FF.FF.FFh		**00h**	**00.00.00.01h 00.FF.FF.FFh**	
LINKSLAUF (LL)	**80h ... FFh**	**00h**	**01h ... 7Fh**	RECHTSLAUF (RL)

Drehrichtung bei Blick auf die Welle

***MSR** ENGINEERING*
Dipl.-Ing. Heiko Evers e.K.
Gut Weilen 8
28759 Bremen

Telefon (0421) 62 43 72
Telefax (0421) 62 43 74
e-mail info@msr-engineering.de
Internet: www.msr-engineering.de