GSV-PROTOKOLLSPEZIFIKATION

GSV-6 / GSV-8

Stand: 11.03.2022

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf

Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69

Mail: info@me-systeme.de
Web: www.me-systeme.de

Änderungsnachweis

Version	Status	Bearbeiter
00.00.01	initiiert	F&S, OlD
00.00.02	erstellt	F&S, SvS
00.01.00	überarbeiten	F&S, SöP
00.01.01	überarbeitet	F&S, OlD
00.02.00	geprüft	F&S, SvS
01.00.00	freigegeben	F&S, SöP
01.00.01	überarbeitet	HK
01.00.02	Überarbeitet	HK
01.00.03	überarbeitet	F&S, OlD
01.00.04	überarbeitet	TMS
01.00.05	überarbeitet & kommentiert bis Kap. 1.11.5	SW
01.00.06	überarbeitet	TMS
01.00.07	überarbeitet	SW
01.00.08	überarbeitet, korrigiert	SW
01.00.09	überarbeitet, Kap. 1.9 ergänzt	SW
01.00.10	überarbeitet, Kap. 1.9 erweitert	SW
01.00.11	Überarbeitet, Kap. 1.5	TMS
01.00.12	Umformatiert auf ME-Vorlage Ooo	SW
01.00.13	Kleinere Änderungen, Ooo Versionierung eingeführt	HK/SW
01.00.14	Read / Write Interface Setting ergänzt.	SW
01.00.15	TEDS-Kommandos ergänzt, kl. Korrekturen	SW
01.00.16	Aktualisiert, Anhang gegliedert	SW
01.00.17	Aktualisiert, korrigiert	SW
01.00.18	Aktualisiert	SW
01.00.19	Aktualisiert, korrigiert	SW
01.00.20	Aktualisiert, ergänzt	SW
01.00.21	Aktualisiert, ergänzt	SW
01.00.22	Aktualisiert, korrigiert, ergänzt	SW
01.00.23	Aktualisiert, ergänzt	SW
01.00.24	Aktualisiert, ergänzt	SW
01.00.25	Aktualisiert, ergänzt	SW
01.00.26	Aktualisiert, ergänzt	SW
01.00.27	Aktualisiert, ergänzt	SW
01.01.00	Neue Cmd. ergänzt, aktualisiert, Anhang F neu	SW
01.01.01	Aktualisiert, ergänzt	SW
01.01.02	Neue Cmd., aktualisiert, ergänzt	SW
01.01.03	Neue Cmd, aktualisiert, ergänzt, kleine Korrekturen,Anhang G neu	SW
01.01.04	Kleine Korrekturen, res. Cmd herausgenommen	SW

Dateiname: PK01079 TB03 01.01.04 GSV-Protokollspezifikation.odt
pdf-Datei: ba-gsvcom.pdf

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Inhaltsverzeichnis

Einleitung

Dieses Dokument beschreibt das Protokoll zweier verschiedener Schittstellen: Seriell/UART und CANbus mit dem proprietären ME Applikationsprotokoll.
Das serielle Interface ist bei beiden Geräten - GSV-6 und GSV-8 - implementiert, das CANbus Interface mit diesem Protokoll gibt es nur beim GSV-6.
Einige GSV-8 Modelle bieten optional ein CANopen Interface, dessen Protokollbeschreibung nicht Teil dieses Dokuments ist (sondern in ba-gsv8canopen.pdf).
Alle GSV-8 Geräte haben eine USB Schnittstelle, die dieses serielle Protokoll verwendet, und zwar über die CDC Geräteklasse, für die bei Verwendung von Windows® 10 kein Treiber installiert werden muss. Das gleiche gilt auch für die meisten Linux-Distributionen. Beide Betriebssysteme erzeugen hiermit einen vituellen COM-Port.

Der GSV-6BT hat ein zusätzliches logisches Interface "BGscript", das dieser Protokoll-Grundstruktur folgt, aber in einem gesonderten Dokument beschrieben ist (ba-gsv6bt-commands.pdf).

Für beide Schnittstellen (Serial/USB-CDC and CAN) gibt es jeweils eine Windows® DLL, um den Grätezugriff für Windows® Programmierer komfortabler zu gestalten. Für die DLLs stehen gesonderte Beschreibungen zur Verfügung.

Zweck

Dieses Dokument beschreibt die Protokollspezifikation für die Serielle/USB-CDC und die proprietäre CANbus Schnittstelle der Geräte GSV-6 und GSV-8 und deren Modellvarianten. Es werden der grundlegende Protokollstrukturaufbau und die einzelnen Befehle beschrieben.

Zielgruppe

Dieses Dokument richtet sich an Entwickler.

Konvention

Folgende beschreibenden Elemente werden verwendet:

<num>	Einzelnes Bit mit Position ‚num' gezählt von 0
<high:low>	Bit-String von Position ‚low' bis ‚high' (inklusive)
[low-high]	Wertebereich (Grenzen inklusive)
name[num]	Byte/Zeichen ‚num' aus einem Array ‚name'

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

GSV-Protokoll

Allgemein

Die Geräte GSV-6 und GSV-8 benutzen für die serielle Kommunikation ein nahezu identisches Protokoll.

Die grundlegende Struktur der Protokollblöcke ist gleich; sie unterscheiden sich bei bestimmten Befehlen in ihren Parametern und dem Befehlsverhalten (systembedingte Abweichungen).

Die Abweichungen sind in den einzelnen Befehlsbeschreibungen markiert.

Numerische Werte werden innerhalb eines Frames im Big-Endian-Format übertragen, d.h. alle Mehrbyte-Datentypen werden vom MSB zum LSB übertragen, bzw. in die Struktur eingefügt. Ebenso werden sie in dieser Reihenfolge in den Befehlsbeschreibungen aufgelistet.

Die Aufstellung der Bits erfolgt in den Struktur-Tabellen immer vom MSBit zum LSBit, bis ein Byte vollständig ist.

Protokoll-Grundstruktur

Die Grundstruktur des Protokolls sieht wie folgt aus:

# Bits	Bezeichnung	Beschreibung
(8)	Präfix	Bei einer nicht Block-Orientierten Datenübertragung (seriell) beginnt ein Paket mit dem Präfix-Byte 0xAA
2	Frame-Type	Beschreibung des übertragenen Frames: 0b00: Messwert-Frame 0b01: Befehl: Antwort 0b10: Befehl: Anfrage 0b11: <Reserviert>
2	Interface	Beschreibt das Interface über den der Frame versendet wird/wurde. 0b00: CAN 0b01: Seriell (RS232, USB, transparent via Bluetooth u.a.) 0b10: Zusätzliches logisches Interface, z.Zt. GSV-6BT: BGscript[1] 0b11: Seriell mit CRC-Checksumme
4	Länge	Dieses Feld wird abhängig von den Feldern Frame-Type und Interface unterschiedlich interpretiert. Die entsprechende Beschreibung erfolgt in den folgenden Kapiteln zu den Frame-Typen.
8	Steuerbyte / Statusbyte	Dieses Feld wird abhängig vom Feld Frame-Type unterschiedlich interpretiert. Die entsprechende Beschreibung folgt in den folgenden Kapiteln zu den Frame-Typen.
0-480 bzw. 0-120	Daten	Das Datenfeld besitzt unterschiedliche Längen und wird Frame Type-spezifisch und befehlsspezifisch interpretiert. Bei CAN wird immer ein vollständiges Paket übertragen und die Daten sind auf 48 Bit = 6 Bytes beschränkt.
(8/16)	CRC-16	Prüfsumme, nur vorhanden wenn Interface-Bits = 0b11
(8)	Suffix	Bei einer nicht Block-Orientierten Datenübertragung (seriell) endet ein Paket mit dem Suffix-Byte 0x85

Tabelle: Protokollgrundstruktur

Der Hauptunterschied in der Grundstruktur liegt bei den unterschiedlichen Schnittstellen. Bei der CAN-Schnittstelle können maximal 8 Byte in einem Paket übertragen werden. Bei einer seriellen Schnittstelle kann die Anzahl der übertragenen Werte größer gestaltet werden, da es sich um einen Stream handelt. Um hier eindeutige Paket-Grenzen zu erhalten, werden Präfix- und Suffix-Bytes übertragen.

Anfrage-Frame (0b10)

Der Anfrage-Frame wird genutzt, um Befehle (Anfragen) an das Gerät zu übermitteln. Jede Anfrage wird grundsätzlich mit einem Antwort-Frame beantwortet (bis auf 2 Ausnahmen,

1 Dieser Interface-Typ ist in einem gesonderten Dokument für den GSV-6BT beschrieben

nämlich GetValue und ResetDevice). Verschiedene Anfragen können u.a. Parameter auslesen („Read...“, „Get...“), schreiben („Write...“, „Set...“) oder Vorgänge auslösen („Set...“). Die Anfragen werden durch die Kommando-Nummer unterschieden; Parameterliste und Rückgabewerte sind diesen Kommandos konstant zugeordnet, d.h. durch diese festgelegt.

Hinweis: Es wird empfohlen, die Antwort auf eine Anfrage erst abzuwarten, bevor eine neue versendet wird.

CAN-Anfrage-Frame

# Bits	Bezeichnung	Beschreibung
2	Frame-Type	0b10: Befehl: Anfrage
2	Interface	0b00: CAN
4	Länge	Anzahl der aktiven Datenbytes [0-6]
8	Steuerbyte	Kommando-Nummer
48	Daten	Kommando-Parameter

Tabelle: CAN-Anfrage Protokoll-Frame

Hinweis: CAN-Anfragen sind immer 64 Bit / 8 Byte groß!

Seriell-Anfrage-Frame

# Bits	Bezeichnung	Beschreibung
8	Präfix	Präfix-Byte 0xAA
2	Frame-Type	0b10: Befehl: Anfrage
2	Interface	0b01: Seriell (RS232, USB ...) ohne CRC. 0b11: Seriell mit CRC-8
4	Länge	Anzahl der Datenbytes [0-15]
8	Steuerbyte	Kommando-Nummer
0-120	Daten	Kommando-Parameter
(8)	Prüfsumme	CRC-8 Checksumme, nur vorhanden mit Interface=0b11
8	Suffix	Suffix-Byte 0x85

Tabelle: Seriell-Anfrage Protokoll-Frame

Antwort-Frame (0b01)

Antwort-Frames erhalten ein Statusbyte, in dem der Fehler-Code enthalten ist. Ein fehlerfrei aufgeführter Befehl kann zusätzlich Rückgabewerte übermitteln (meist bei Leseanfragen), d.h. das Statusbyte lautet dann meistens ERR_OK =0 (Ausnahme: s. 'Seriell').
Wird hingegen ein Fehler signalisiert (Statusbyte >0, Länge =0), enthält der Anwortframe bei Interfacetypen 'Seriell' und 'CAN' keine Daten.

CAN-Antwort-Frame

# Bits	Bezeichnung	Beschreibung
2	Frame-Type	0b01: Befehlsantwort
2	Interface	0b00: CAN
4	Länge	Anzahl der aktiven Bytes in diesem Rückgabeframe [0-6]
8	Statusbyte	Fehler-Code (siehe Fehlercodes Fehlercodes)
48	Daten	Rückgabewert(e)

Tabelle: CAN-Antwort Protokoll-Frame

Hinweis: CAN-Antworten sind immer 64 Bit = 8 Bytes groß!

Seriell-Antwort-Frame

# Bits	Bezeichnung	Beschreibung
8	Präfix	Präfix-Byte 0xAA
2	Frame-Type	0b01: Befehlsantwort
2	Interface	0b01: Seriell (RS232, USB ...) ohne CRC. 0b11: Seriell mit CRC-8
4	Länge	Anzahl der Daten-Bytes in diesem Rückgabeframe [0-15] =15: Anzahl der Daten-Bytes = <Statusbyte> + 15
8	Statusbyte	Fehler-Code (siehe Fehlercodes) oder zusätzliche Länge
0-120 / 120-2160	Daten	Rückgabewert(e)
(8)	Prüfsumme	CRC-8 Checksumme, nur vorhanden mit Interface=0b11
8	Suffix	Suffix-Byte 0x85

Tabelle: Seriell-Antwort Protokoll-Frame

Wenn das Längenfeld =15 ist, steht im Statusbyte eine additive Längenangabe, d.h. die Länge des Datenfeldes ist dann: <Statusbyte> +15. Auf diese Weise kann der GSV bis zu 270 Bytes zurückgeben, statt bis zu 15, wenn <Länge> kleiner ist als 15.
Ist das Längenfeld <15, steht im Statusbyte ein Fehlercode.
Dieser lange Antwortframe wurde eingeführt beim GSV-6BT mit FW-version 3.20, für die Kommandos *Read File Extended* und *Read Value String.*

Prüfsumme bei seriellen Kommando- und Antwortframes

Wenn das Bitfeld "Interface" bei Kommandoanfragen 0b11 lautet, d.h. wenn Bit 5 des zweiten Bytes im seriellen Anfrageframe =1 ist, so wird auch die Antwort die Prüfsumme enthalten und dies ebenfalls durch Interfacebits= 0b11 signalisieren.
Diese CRC-8 Prüfsumme der Anfrage- und Antwortframes wird über die Protokollbytes <Frametype><Interface><Länge> und Statusbyte sowie alle Datenbytes berechnet, d.h. über alle Bytes im Frame ohne Präfix und ohne Suffix (und selbstverständlich ohne die Prüfsumme selbst). Details dazu im Kapitel "Prüfsummenberechnung", S.22. Die Prüfsumme wird vom GSV-8 ab Firmwareversion 1.56 und vom GSV-6 ab Version 3.35

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

unterstützt.

Messwert-Frame (0b00)

Im Messwert-Frame werden die erfassten Messwerte übermittelt. Messwert-Frames können zyklisch und autonom vom Gerät versendet werden.

CAN-Messwert-Frame

# Bits	Bezeichnung		Beschreibung	
2	Frame-Type		0b00: Messwert-Frame	
2	Interface		0b00: CAN	
4	Kanal		Kanalnummer beginnend mit 0	
8	Steuerbyte / Statusbyte		<7> Indikator ==0 <6:4> Datentyp siehe Datentyp Datentyp <3:0> Error Bits - <3:2> reserviert • <1> 6-Achsen-Fehler • <0> Übersteuerung des Eingangs	
48	Daten	16 Bit	TimeStamp	Zähler, der bei den zeitgleich erfassten Kanälen identisch ist und mit jedem Zyklus um 1 inkrementiert wird. Es besteht kein Zusammenhang zu einer Zeitbasis.
		32 Bit*	Messwert	Der erfasste Messwert des übertragenen Kanals. * Ist der Datentyp kleiner als 32Bit so werden Null-Bits angehängt.

Tabelle: CAN-Messwert-Frame

Hinweis: CAN-Messwert-Frames sind immer 64 Bit / 8 Byte groß!

Serieller-Messwert-Frame

# Bits	Bezeichnung	Beschreibung
8	Präfix	Präfix-Byte 0xAA
2	Frame-Type	0b00: Messwert-Frame
2	Interface	0b01: Seriell (RS232, USB ...) ohne CRC. 0b11: Seriell mit CRC-16
4	Anzahl Objekte	Anzahl der übertragenen Messwertobjekte (ggf. Kanäle) minus 1 GSV-6: 1-6. GSV-6BT: 1-7 GSV-8: 2-10 (High-speed Modus: bis 15)
8	Steuerbyte / Statusbyte	<7> Indikator ==1 <6:4> Datentyp siehe Datentyp <3:0> Error Bits - <3:2> reserviert - <1> 6-Achsen-Fehler - <0> Übersteuerung des Eingangs
16-512	Daten	Pro Kanal ein Messwert mit angegebenem Datentyp High-speed Modus: Bis zu 8 Kanalsequenzen, s.u.
(16)	Prüfsumme	CRC-16 Checksumme, nur vorhanden mit Interface=0b11
8	Suffix	Suffix-Byte 0x85

Tabelle: Serieller-Messwert-Frame

Datentyp

Es sind folgende Datentypen für Messwert-Frames definiert:

Bits<6:4> Statusbyte	Enum Cmd 0x80.1 / 0x81.1	Typenname
0b000	-	reserviert
0b001	1	int16
0b010	2	int24 [nur GSV-8]
0b011	3	float32
0b100	-	reserviert
0b101	-	reserviert
0b110	-	reserviert
0b111	-	reserviert

Tabelle: Messwert-Datentyp Definition

Auswertung des Messdatenframes

Die Bytes eines jeden Messwertobjektes werden in Big-Endian Reihenfolge übertragen, d.h. das höchstwertigste Byte (MSByte) kommt zuerst. Innerhalb des Datenframes bzw. einer Kanalsequenz werden niedrigere Kanalnummern zuerst übertragen. Das erste Byte des Datenfeldes ist also das MSbyte des Messwertes des Eingangskanals 1.

Normalmodus

Der Messdatenframe enthält die Messwerte gleichzeitig (bzw bei GSV-6 fast gleichzeitig) gesampelter Eingangskanäle. Das Header-Feld "Anzahl Objekte" nennt die Anzahl der zu Eingangskanälen gehörenden Messwerte -1.

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

High-Speed Modus

Beim GSV-8 ab Firmware Version 1.54 können am USB-Interface mehrere Kanalsequenzen innerhalb eines Messdatenframes übertragen werden, wenn die Messdatenfrequenz >=12000 Frames/s ist. Dadurch verringert sich der Protokoll-Overhead und der Datendurchsatz verbessert sich. Dieser Frame-Modus wird nur verwendet, wenn er durch den Host, d.h. die kommunizierende Instanz, erlaubt wird. Dies geschieht durch Setzen von Bit 2 des Übergabeparameters des Kommandos GetInterface (s. S.27). Das Header-Feld "Anzahl Objekte" nennt die Gesamtanzahl der Messdatenobjekte im Datenframe -1. Die Kanalsequenzen werden mit den ältesten Werten zuerst übertragen, innerhalb einer Sequenz kommt Kanal 1 zuerst. Zur Auswertung sollte die konfigurierte Kanalanzahl bekannt sein; diese kann durch das Kommando GetTxMapping an Index 0 gelesen werden, s.S.52. Die Kanal- und Sequenzkonfiguration kann sich nur dann ändern, wenn die Kanalanzahl oder die Messdatenrate verändert wird, oder durch Aufruf von Get Interface.

Prüfsumme

Der Messdatenframe im Normalmodus (s.o.) kann so konfiguriert werden, dass er eine CRC-16 Prüfsumme enthält. Im Auslieferungszustand hat der Messdatenframe keine Prüfsumme. Sie kann mit dem Kommando GetInterface (No. 0x01) oder SetTXmode (No. 0x81) aktiviert werden. Dieser Zustand wird nur beim Setzen durch SetTXmode nichtflüchtig gespeichert und zwar für USB und UART, wenn vorhanden. Bei Aufruf von GetInterface hingegen muss das Bit 3 des Aufrufparameters unabhängig vom Zustand des TXmode-Wertes richtig gesetzt sein; dieser Zustand ist flüchtig und gilt nur für das Interface, mit dem der Aufruf erfolgt ist. Die Prüfsumme wird vom GSV-8 ab Firmwareversion 1.56 und vom GSV-6 ab 3.35 unterstützt. Bei dieser CRC-16 Prüfsumme wird das niederwertige Byte zuerst übertragen (Little-Endian). Details dazu im Kapitel "Prüfsummenberechnung", S.22.

Interpretation der Messdaten

Die richtige Interpretation der Messwerte, d.h. die Umrechnung zu physikalischen Werten, hängt vom Messdatentyp und der korrekten Parametrierung anhand des verwendeten Sensors ab. Beim Messdatentyp float32 werden fertig normierte Messwerte übertragen. Sofern der Messverstärker richtig parametriert ist (u.a. richtige Kalibriermatrix beim Sechsachsensensor oder richtige UserScale-Werte), entsprechen diese den physikalischen Messwerten.

Bei den Integer-Datentypen müssen die Werte folgendermaßen umgerechnet werden:

1. Beim GSV-8 liegen die Rohwerte im sog. Binary Offset-Format vor. Daher muss dieser Offset zunächst subtrahiert werden, um Signed-Int Werte zu erhalten:

int16: ZwischenWert_1 = Rohwert - 0x8000
int24: ZwischenWert_1 = Rohwert - 0x800000
Beim GSV-6 wird dieser Schritt übersprungen: ZwischenWert_1 = Rohwert

2. Dieses Zwischenergebnis wird zur Dezimalzahl (d.h. Fließkommazahl) umgewandelt, dann mit dem Messbereichsoverhead 1,05 multipliziert und durch den binären (unipolaren) Messbereich dividiert:

int16: ZwischenWert_2 = ZwischenWert_1 * 1,05 / 2^15

int24: ZwischenWert_2 = ZwischenWert_1 * 1,05 / 2^23

3. So erhält man Messwerte, die stets auf +-1,0 normiert sind. Der Wert 1,0 entspricht dabei dem Nenn-Eingangsmessbereich. Ist dieser z.B. 2mV/V, so müsste der ZwischenWert_2 mit 2 multipliziert werden, um die Brückenverstimmung direkt in mV/V anzuzeigen. Ist der Eingangsmessbereich z.B. Single-Ended 10V, müsste ZwischenWert_2 mit 10 multipliziert werden, um die Eingangsspannung direkt in Volt anzuzeigen (vorausgesetzt, der Nullpunkt wurde nicht durch SetZero o.a. auf einen Wert ungleich 0 verschoben).

Diese Berechnung berücksichtigt die Übertragungsfunktion des Sensors (physikalische Größe vs. elektrische) nicht. Dies kann durch einen zusätzlichen Faktor erfolgen, mit dem der ZwischenWert_2 dann multipliziert wird. Er kann mit WriteuserScale (Cmd. Nr 0x15) im Gerät gespeichert werden; Anwendungssoftware wie GSVmulti verwendet ihn zur Darstellung physikalisch skalierter Messwerte.

Folgende Tabelle zeigt beispielhaft Rohwerte und ZwischenWert_2 für einen Nenn-Eingangsmessbereich von 2mV/V:

Sensor-auslenkung in mV/V	GSV-8		GSV-6	Lesewert MEGSV86xx.dll:: GSVread u.ä. Messwert-Lesefunktionen (ZwischenWert_2)
	Ganzzahliger Messwert, 16-Bit (Uint16) Hex	Ganzzahliger Messwert, 24-Bit (Uint24) Hex	Ganzzahliger Messwert, 16-Bit (Sint16) Hex	
<= -2,1	0x0000	0x000000	0x8000	-1,05
-2,0	0x0618	0x061862	0x8618	-1,0
0	0x8000	0x800000	0x0000	0,0
2,0	0xF9E7	0xF9E79E	0x79E7	1,0
>= 2,1	0xFFFF	0xFFFFFF	0x7FFF	1,05

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Fehlercodes

In diesem Kapitel werden alle definierten Fehler-Codes aufgelistet. Im Statusbyte des Antwort-Frame wird einer der Werte, situationsabhängig, hinterlegt.

Einige Fehlercodes sind reserviert für künftige Anwendungen.

Errorbyte im Befehlsanwortframe

Fehlerkürzel	**Wert** (Hex)	**Bedeutung**
ERR_OK	0x00	Kommando ohne Fehler ausgeführt.
ERR_OK_CHANGED	0x01	Kommando ohne Fehler ausgeführt, aber zusätzlich wurden weitere Parameter geändert.
ERR_CMD_NOTKNOWN	0x40	Das Kommando ist unbekannt.
ERR_CMD_NOTIMPL	0x41	Das Kommando ist bekannt, wird jedoch nicht unterstützt. Es kann sich hier um Kommandos handeln, die für den jeweiligen Gerätetypen (GSV-6/-8) nicht vorgesehen sind.
ERR_FRAME_ERROR	0x42	Der Frame des Kommandos ist falsch aufgebaut.
ERR_CMD_CRC	0x43	CRC-Prüfsummenfehler bei Kommandoanfrage
ERR_PAR	0x50	Ein falscher Parameter wurde übergeben.
ERR_PAR_ADR	0x51	Der übergebene Index oder die Adresse ist falsch.
ERR_PAR_DAT	0x52	Die Daten eines Datenparameters sind falsch.
ERR_PAR_BITS	0x53	Falsche Bits im Parameter.
ERR_PAR_ABSBIG	0x54	Der Wert eines Parameters ist absolut zu groß.
ERR_PAR_ABSMALL	0x55	Der Wert eines Parameters ist absolut zu klein.
ERR_PAR_COMBI	0x56	Die Parameterkombination ist fehlerhaft.
ERR_PAR_RELBIG	0x57	Ein Parameter ist in Relation zu den anderen zu groß.
ERR_PAR_RELSMALL	0x58	Ein Parameter ist in Relation zu den anderen zu klein.
ERR_PAR_NOTIMPL	0x59	Das Kommando unterstützt den Parameter nicht.
ERR_PAR_TIMEOUT	0x5A	Die Parameter zum Kommando sind nicht innerhalb von 200 ms eingetroffen.
ERR_WRONG_PAR_NUM	0x5B	Die Anzahl der Parameter passt nicht zum Kommando. Bits 3~0 im ersten Byte des Kommando Blocks zu hoch bzw. niedrig.
ERR_PAR_NOFIT_SETTINGS	0x5C	Parameter entspricht nicht der Geräteeinstellung
ERR_PAR_HW_COLLISION	0x5D	Durch Parameter angeforderte Funktion führt zu einer Hardware-Kollision
ERR_NO_DATA_AVAIL	0x60	Abgewiesen bei Leseanfrage, weil Daten nicht vorhanden
ERR_DATA_INCONSISTENT	0x61	Gespeicherte Daten falsch, bzw. inkonsistent mit Zustandsanforderung
ERR_WRONG_MOD_STATE	0x62	Befehl konnte nicht ausgeführt werden, weil Gerät bzw. Modul im ungeeignetem Zustand
ERR_NOT_SUPPORTED_D	0x63	Abgewiesen, weil angeforderte Funktionalität nicht unterstützt
ERR_FDATA_TOO_HIGH	0x64	Abgewiesen, weil Datenrate zu hoch, bzw. Prozessor

Fehlerkürzel	Wert (Hex)	Bedeutung
		wäre zu ausgelastet
ERR_MEMORY_WRONG_COND	0x6E	Abgewiesen bei Memory-Interface-write, weil Bed. nicht erf.
ERR_MEMORY_ACCESS_DENIED	0x6F	Abgewiesen bei Memory-Interface-write
ERR_ACC_DEN	0x70	Kommando wird nicht ausgeführt, weil die Berechtigung fehlt.
ERR_ACC_BLK	0x71	Kommando wurde nicht ausgeführt, da Blocking gesetzt ist.
ERR_ACC_PWD	0x72	Passwort falsch oder nicht gesetzt.
ERR_ACC_MAXWR	0x74	Die maximale Anzahl der Ausführungen ist überschritten.
ERR_ACC_PORT	0x75	Keine Schreibrechte auf dem Port.
ERR_ACC_RDONLY	0x76	Versuch, einen nur lesbaren Parameter zu schreiben
ERR_INTERNAL	0x80	Interner Ausnahmefehler.
ERR_ARITH	0x81	Interner arithmetischer Fehler.
ERR_INTER_ADC	0x82	Fehlerhaftes Verhalten des AD-Umsetzers.
ERR_MWERT_ERR	0x83	Zur Befehlsausführung ungeeigneter Messwert
ERR_EEPROM	0x84	Fehlerhaftes Verhalten des EEPROMs
ERR_EXT_HW	0x85	Nötige externe Hardware (zB SD-Karte) fehlerhaft o. nicht vorhanden
ERR_FILE	0x86	Dateisystemtreiber meldet Fehler
ERR_WRONG_DIR	0x87	Falsches Verzeichnis, bzw Verzeichniseinstellung ungeeignet
ERR_RET_TXBUF	0x91	Der Sendepuffer ist voll.
ERR_RET_BUSY	0x92	Die CPU ist zu ausgelastet, um einen Befehl auszuführen.
ERR_RET_RXBUF	0x99	Der Empfangspuffer ist voll.
GETTEDS_ERR_NOSENSOR	0xB0	TEDS: Kein Sensor angeschlossen
GETTEDS_ERR_NOTEDSEE	0xB1	Kein TEDS-fähiger Speicherbaustein angeschlossen
GETTEDS_ERR_BASICONLY	0xB2	Der TEDS-Speicher enthält nur Basic-Template Daten
GETTEDS_ERR_NOTEDSDAT	0xB3	Daten im TEDS-Speicher entsprechen nicht der Norm
GETTEDS_ERR_ENTRY_INVALID	0xB4	Eintrag im TEDS-Speicher nicht (richtig) gesetzt
GETTEDS_ERR_TOUT	0xB5	Hardware-TEDS-Gerätetreiber timeout
GETTEDS_ERR_CHKSUM	0xB6	Prüfsummenfehler der TEDS Daten
GETTEDS_ERR_UNKNOWN_TEMPL	0xB7	(Noch) nicht unterstütztes TEDS-template
GETTEDS_ERR_VERIFY_FAIL	0xB8	Leseverifizierung nach Schreiben fehlgeschlagen
BT_CONFIG_ERR	0xC0	BGscript: Fehler in der BlueTooth Anwendung

Tabelle: Fehlercodes des Errorbytes im Befehlsanwortframe

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Fehlercodes für „GetLastValueError" (0x43) beim GSV-8

Die 48 Bits (6 Bytes) der Error-Struktur im Einzelnen:

Bits<47:45>

Error-Type: Art des Fehlers (Kategorie)		
Wert	Name	Bedeutung
1	VALERR_TYPE_SATURATED	Analogwert am Sensoreingang gesättigt, d.h. Eingangsmessbereich überschritten
2	VALERR_TYPE_MAX_EXCEED	Maximal erlaubter physikalischer Wert überschritten (insbes. bei 6-Achsensensor)
3	VALERR_TYPE_SENSOR_BROKEN	Brückensensor oder dessen Anschlussleitung defekt
4	HWERR_TYPE_ANA_OUT	Als Stromausgang konfigurierter Analogausgang offen oder Analogausgangstreiber überhitzt
5	HWERR_TYPE_DIO	Als Ausgang konfigurierter GPIO (Digitalanschluss) kurzgeschlossen

Bits<44:16>

Error-Time: Zeit des Fehlers

Gespeichert wird die Fehlerzeit in Minuten des Gerätebetriebs, d.h. dieser Wert/60 sind die (absoluten) Betriebsstunden, zu der der Fehler auftrat.

Bits<15:0> Error-Flags Typabhängige Fehlerkodierung

Beschreibung der Error-Flags im Einzelnen:

1. ErrType = VALERR_TYPE_SATURATED:

Bits<15:8>: Wenn Bit=1: Negative Sättigung ist in einem oder mehreren Eingangskanälen aufgetreten, wobei Bit 8 Kanal 1 entspricht, Bit 9 Kanal 2, usw, bis Bit 15: Kanal 8.

Bits<7:0>: Wenn Bit=1: Positive Sättigung ist in einem oder mehreren Eingangskanälen aufgetreten, wobei Bit 0 Kanal 1 entspricht, Bit 1 Kanal 2, usw, bis Bit 7: Kanal 8.

Bemerkungen:

1. Liegt dieser Fehler aktuell vor, so leuchtet die "FUNCTION"-LED des Gerätes dauernd rot.

2. Ferner ist dann das Bit 0 des Statusbytes im Messdatenframe gesetzt.

2. ErrType = VALERR_TYPE_MAX_EXCEED:

2.1. Bei Sechsachsensensor

Bit0: Wenn Bit=1: In Fx-Richtung ist ein (positives oder negatives) Überschreiten des in den Sechachsenkalibrierdaten definierten Maximalwertes aufgetreten.

Bit1: Wenn Bit=1: In Fy-Richtung ist ein (positives oder negatives) Überschreiten des in den Sechachsenkalibrierdaten definierten Maximalwertes aufgetreten.

Bit2: Wenn Bit=1: In Fz-Richtung ist ein (positives oder negatives) Überschreiten des in den Sechachsenkalibrierdaten definierten Maximalwertes aufgetreten.

Bit3: Wenn Bit=1: In Mx-Richtung ist ein (positives oder negatives) Überschreiten des in den Sechachsenkalibrierdaten definierten Maximalwertes aufgetreten.

Bit4: Wenn Bit=1: In My-Richtung ist ein (positives oder negatives) Überschreiten des in den Sechachsenkalibrierdaten definierten Maximalwertes aufgetreten.

Bit5: Wenn Bit=1: In Mz-Richtung ist ein (positives oder negatives) Überschreiten des in den Sechachsenkalibrierdaten definierten Maximalwertes aufgetreten.

2.2. Bei PT1000 Temperatursensor:

Bits<15:0>: Wenn der Messwert das maximum von 1500°C über- oder das Minimum von -230°C unterschreitet, werden die beiden Bits [KanalNr. -1] und [KanalNr. +7] gesetzt, z.B. 0x0101 für Kanal 0, 0x0202 für Kanal 2 usw., bis 0x8080 für Kanal 8.

Bei einer Maximalwertüberschreitung wird der kommunizierte Ausgangswert auf 9999 gesetzt, bei einer Unterschreitung des Minimums wird er auf -9999 gesetzt.

Bemerkungen:

1. Liegt dieser Fehler aktuell vor, so leuchtet die "FUNCTION"-LED des Gerätes dauernd rot.

2. Ferner ist dann das Bit 1 des Statusbytes im Messdatenframe gesetzt.

3. ErrType = VALERR_TYPE_SENSOR_BROKEN:

Bit0: Wenn Bit=1: An der Ud+ Leitung des Kanals 1 lag ein Fehler vor.

Bit1: Wenn Bit=1: An der Ud- Leitung des Kanals 1 lag ein Fehler vor.

Bit2: Wenn Bit=1: An der Ud+ Leitung des Kanals 2 lag ein Fehler vor.

Bit3: Wenn Bit=1: An der Ud- Leitung des Kanals 2 lag ein Fehler vor.

Bit4: Wenn Bit=1: An der Ud+ Leitung des Kanals 3 lag ein Fehler vor.

Bit5: Wenn Bit=1: An der Ud- Leitung des Kanals 3 lag ein Fehler vor.

Bit6: Wenn Bit=1: An der Ud+ Leitung des Kanals 4 lag ein Fehler vor.

Bit7: Wenn Bit=1: An der Ud- Leitung des Kanals 4 lag ein Fehler vor.

Bit8: Wenn Bit=1: An der Ud+ Leitung des Kanals 5 lag ein Fehler vor.

Bit9: Wenn Bit=1: An der Ud- Leitung des Kanals 5 lag ein Fehler vor.

Bit10: Wenn Bit=1: An der Ud+ Leitung des Kanals 6 lag ein Fehler vor.

Bit11: Wenn Bit=1: An der Ud- Leitung des Kanals 6 lag ein Fehler vor.

Bit12: Wenn Bit=1: An der Ud+ Leitung des Kanals 7 lag ein Fehler vor.

Bit12: Wenn Bit=1: An der Ud- Leitung des Kanals 7 lag ein Fehler vor.

Bit14: Wenn Bit=1: An der Ud+ Leitung des Kanals 8 lag ein Fehler vor.

Bit15: Wenn Bit=1: An der Ud- Leitung des Kanals 8 lag ein Fehler vor.

Bemerkung: Liegt dieser Fehler aktuell vor, so leuchtet die "FUNCTION"-LED des Gerätes dauernd rot.

4. ErrType = HWERR_TYPE_ANA_OUT:

Konstanter Wert 0xFFFF: An irgendeinem der Analogausgangskanäle lag ein Fehler vor. Sonst:

Bit0: Wenn Bit=1: An Ausgangskanal 1 liegt ein offener Stromausgang vor.

Bit1: Wenn Bit=1: An Ausgangskanal 2 liegt ein offener Stromausgang vor.

Bit2: Wenn Bit=1: An Ausgangskanal 3 liegt ein offener Stromausgang vor.

Bit3: Wenn Bit=1: An Ausgangskanal 4 liegt ein offener Stromausgang vor.

Bit4: Wenn Bit=1: An Ausgangskanal 5 liegt ein offener Stromausgang vor.

Bit5: Wenn Bit=1: An Ausgangskanal 6 liegt ein offener Stromausgang vor.

Bit6: Wenn Bit=1: An Ausgangskanal 7 liegt ein offener Stromausgang vor.

Bit7: Wenn Bit=1: An Ausgangskanal 8 liegt ein offener Stromausgang vor.

Bit8: Wenn Bit=1: An Ausgangskanal 1 ist der Ausgangstreiber überhitzt

Bit9: Wenn Bit=1: An Ausgangskanal 2 ist der Ausgangstreiber überhitzt

Bit10: Wenn Bit=1: An Ausgangskanal 3 ist der Ausgangstreiber überhitzt

Bit11: Wenn Bit=1: An Ausgangskanal 4 ist der Ausgangstreiber überhitzt

Bit12: Wenn Bit=1: An Ausgangskanal 5 ist der Ausgangstreiber überhitzt

Bit13: Wenn Bit=1: An Ausgangskanal 6 ist der Ausgangstreiber überhitzt

Bit14: Wenn Bit=1: An Ausgangskanal 7 ist der Ausgangstreiber überhitzt

Bit15: Wenn Bit=1: An Ausgangskanal 8 ist der Ausgangstreiber überhitzt

Bemerkungen:

1. Der Überhitzung des Ausgangstreibers liegt möglicherweise ein Kurzschluss des dementsprechenden Spannungsausgangs zugrunde.

2. Liegt dieser Fehler aktuell vor, so blinkt die "FUNCTION"-LED des Gerätes langsam (ca 1x/Sek) rot.

5. ErrType = HWERR_TYPE_DIO:

Bits<15:0> Wenn Bit=1: An der entsprechneden DIO-No. ist ein Kurzschluss aufgeteten, d.h. wenn dieser als Ausgang und auf High geschaltet ist, ist er mit GNDD kurzgeschlossen, oder wenn er auf Low geschaltet ist, ist eine Spannung >=3V angeschlossen. Bit 0 entspricht dabei DIOno 1 (Group1: 1.1.), Bit 1 DIOno 2 (Group1: 1.2.), usw bis Bit 15: DIOno 16 (Group4: 4.4.)

Bemerkung:

Liegt dieser Fehler aktuell vor (Index 1, s.o.), so blinkt die "FUNCTION"-LED des Gerätes schnell (ca 3x/Sek) rot.

Fehlercodes für „GetLastValueError" (0x43) beim GSV-6

Bits <5:0> / Byte 0: Maximal erlaubter physikalischer Wert bei 6-Achsensensor überschritten. Bit 0: Überschreitung an Kanal 1 (Fx), usw. bis Bit 5: Überschreitung an Kanal 6 (Mz).

Bits <13:8> / Byte 1: Analogwert am Sensoreingang gesättigt, d.h. Eingangsmessbereich überschritten. Bit 8: Sättigung an Kanal 1, usw. bis Bit 13: Sättigung an Kanal 6

Prüfsummenberechnung

Wenn bei seriellen Frames die Bits <5:4> des zweiten Protokollbytes =0x3 lauten, enthält der Frame eine CRC Prüfsumme. Diese wird über die Information enthaltenen Protokoll-Bytes 2 und 3 und ggf. über alle Datenbytes berechnet. Sie liegt in den Frames nach den Nutzdaten, direkt vor dem Suffix. Das Vorhandensein der Prüfsumme wird für das Befehlsinterface und die Messdatenframes unabhängig voneinander verwaltet. Bei Kommandos gilt: Hat die Anfrage den CRC-8, so wird auch die Antwort damit versehen.

Bei den Messdatenframes gilt:
1. Wenn der CRC-16 mit dem Kommando SetTXmode eingeschaltet wurde, so sendet der GSV-8 ab dem nächsten Neustart zunächst bei allen seriellen Interfaces Messdatenframes mit CRC-16. Dieser Zustand wird nichtflüchtig gespeichert.
2. Dieser Zustand kann temporär (d.h. flüchtig) durch das Kommando GetInterface

überschrieben werden. Ist beim Übergabeparameter von GetInterface das Bit 3 gesetzt, so werden Frames mit CRC-16 gesendet, ist es nicht gesetzt, Frames ohne CRC-16. Letzteres gilt auch dann, wenn es in TxMode aktiviert ist. Auf diese Weise ist der GSV-8 mit älterer (bzw bestehender) Anwendungssoftware kompatibel, die die Windows-DLL MEGSV86w32.dll verwendet, zB GSVmulti (denn diese sendet bei der Initialisierung stets GetInterface mit Bit3=0).

CRC-16 bei Messdatenframes

Beim Messdatenframe ist die Prüfsumme ein 16-Bit Wert, bei dem ausnahmsweise das niederwertigere Byte zuerst kommt (Little-Endian). Sie entspricht dem CRC16 des Modbus-Protokolls, d.h. das Generator Polynom lautet:
$x^{16}+x^{15}+x^2+1$ oder 0x8005. Eingangs- und Ausgangsdaten sind in der Bitreihenfolge umgekehrt (Bit-reversed) und der CRC-Startwert lautet 0xFFFF.
Beispiel (hex):
`AA 37 B0 C1 C7 CD 38 3F E6 19 7E 3F C0 B6 0B BF 49 7E 95 40 22 DD 1D 3F B2 11 53 3E E6 C3 72 3F 92 65 3B E7 6E 85`
Die Prüfsumme wird hier über die 34 Bytes von 0x37 bis 0x3B berechnet und das Ergebnis ist 0x6EE7. Ein Beispiel für eine Berechnungsroutine in C ist im Anhang G, S.104

CRC-8 bei Kommandos

Bei Kommandoanfragen und -antworten ist die Prüfsumme ein 8-Bit Wert, der jeweils auf die gleiche Art berechnet wird. Das Generatorpolynom lautet:

x^8+x^2+x+1 oder 0x07. Eingangs- und Ausgangsdaten sind hierbei in der Bitreihenfolge regelmäßig, d.h. nicht umgekehrt und der CRC-Startwert lautet 0x00.
Beispiele (hex):

`AA B1 01 08 AC 85` (Get Interface Anfrage mit Aktivierung des CRC-16 im Messdatenframe)

`AA 74 00 C8 73 00 02 B9 85` (Get Interface Antwort darauf)

`AA B0 23 A6 85` (StopTX zum Stoppen des Messdatenstroms)

`AA 70 00 A2 85` (Allgemeine Antwort: OK)

Ein Beispiel für eine Berechnungsroutine dieses CRC-8 in C ist im Anhang G zu finden.

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Befehlsinterface

Im folgenden Unterkapitel wird die Struktur der Befehls-Dokumentation erläutert, danach folgt das Unterkapitel mit der Beschreibung jedes Befehls.

Allgemeine Struktur der Beschreibungen

Bei der Beschreibung der einzelnen Befehle wird folgende Tabellenstruktur genutzt:

Nr.	**Richtung (R / W) oder A: löst Aktion aus**	**Name**	**Gerätemodel**
Beschreibung			
Nr. Bytes	Nr. Parameter		
Offset P1	Parameter1 Typ	Paramter1 Name	Parmeter1 Beschreibung
Offset P2	Parameter2 Typ	Paramter2 Name	Paramter2 Beschreibung
Nr. Bytes	Nr. Rückgaben		
Offset R1	Rückgabe 1 Typ	Rückgabe1 Name	Rückgabe1 Beschreibung

Tabelle: Allgemeine Befehlsbeschreibung

Wenn unter "Richtung" **W** angegeben ist, wird der Wert nichtflüchtig gespeichert. Bei **A** wird nur eine Aktion ausgelöst, ohne nichtflüchtiges Speichern. "**W / A"** löst eine Aktion aus und speichert das Ergebnis nichtflüchtig. "**W (RAM)**" bedeutet, dass der Wert zunächst nur im RAM gespeichert wird und zum nicht-flüchtigen Speichern ein anderes Kommando vorgesehen ist.

Zu beachten ist, dass die angegebene Anzahl der Parameter/Rückgabe-Bytes und die Offsets sich nur auf die (Lese/Schreib-) Daten im Anfrage-/Antwort-Frame beziehen. Bei Rückgabe eines Standard-Anwortframes ohne Nutzdaten wird "0 Rückgaben" genannt.

Sollten sich die Befehle von GSV-6 und GSV-8 grundlegend unterscheiden (z.B. in Datentypen), werden separate Beschreibungen aufgeführt. Ebenso bei größeren Kommunikationsweg-Unterschieden (CAN, Seriell).

Als Gerätemodell können folgende Einträge (auch in Kombination) auftreten:

Gerätemodel	**Beschreibung**
GSV-6	GSV-6 allgemein
GSV-6 (Reset)	GSV-6 Befehl benötigt zur Aktivierung der Änderung einen Reset/Powercycle.
GSV-6BT	GSV-6 Bluetooth
GSV-6BTanalog	Bluetooth-Transceiver mit Analogausgängen, für GSV-6BT (reserviert)
GSV-8	GSV-8 allgemein
GSV-8 CAN	GSV-8 mit CANopen Schnittstelle

Tabelle: Gerätemodelle

Die auftretenden Datentypen in Parameter- und Rückgabetyp sind:

Bezeichner	**Bytes**	**Beschreibung**
uint8_t	1	Vorzeichenloser 8-Bit Integer
uint16_t	2	Vorzeichenloser 16-Bit Integer
U24	3	Vorzeichenloser 24-Bit Integer
S24	3	Vorzeichenbehafteter 24-Bit Integer
uint32_t	4	Vorzeichenloser 32-Bit Integer
int32_t	4	Vorzeichenbehafteter 32-Bit Integer
S7.24	4	Vorzeichenbehaftete Fixpunkt-Zahl (Vorzeichen-Bit, 7 Vor-, 24 Nachkommabits)
float	4	32-Bit Fließkommazahl

Bezeichner	Bytes	Beschreibung
char[x]	x	Zeichenkette bzw Byte-Array

Tabelle: Parameter und Rückgabe Datentypen

Bei den Befehlen treten diverse Elemente als Parameter bzw. Rückgabetyp häufig auf, diese logischen Typen sind in der folgenden Tabelle kurz beschrieben:

Bezeichner	Beschreibung
Kanal	Kanal-Nummer werden i.a. mit 1 beginnend gezählt, bis auf die dokumentierten Ausnahmen. Der GSV-6 hat 6 Kanäle mit Kanal-Nr. 1-6, der GSV-6BT bis zu 7. Der GSV-8 hat 8 Analogkanäle 1-8 und bis zu 2 Counter/Frequenzkanäle. Als Sonderwert kann bei vielen Schreibbefehlen (Set*, Write*) der Wert 0 übergeben werden, so dass der entsprechende Wert für alle Kanäle gesetzt wird.
Index, Sub Index	Meist ein Wert mit dem auf ein Array zugegriffen, bzw. ein durch Kanal/Adresse spezifiziertes Element ausgewählt wird.
Flags	Bit-Felder, deren Bit-Position einer Konfiguration entsprechen.

Tabelle: Häufige Parameter- bzw. Rückgabewert-Bezeichner

Befehlskategorien

Kommandos werden grundsätzlich mit einem Antwortframe beantwortet[2], sofern bei der Anfage die Protokoll-Grundstruktur eingehalten ist (z.B. Präfix 0xAA vorhanden), s.o. Der Antwortframe enthält einen der oben aufgelisteten Fehlercodes.

Parametrierung, Speicherverhalten

Im allgemeinen speichern die Geräte konfigurierte Parameter im nichtflüchtigen Speicher, deshalb muss eine Parametrierung mit gewünschten Geräteeinstellungen nur einmalig erfolgen. Der nichtflüchtige Speicher ist ein EEPROM oder ein Flash-ROM mit Schreibmanagement (wear-levelling). Dennoch ist die maximale Anzahl der Schreibvorgänge begrenzt, so dass es nicht empfehlenswert ist, Parameter in einem Programm pauschal stets zu setzen. Das ist insbesondere beim GSV-6 ungünstig, da dieser nach jedem Schreibbefehl die Parameter ins Flash überträgt. Stattdessen ist es besser, Parameter erst zu lesen, und diese nur dann zu schreiben, wenn sie vom gewünschten Wert abweichen.

Datenfluss konfigurieren

Die autonome permanente Datenübertragung kann mit *StartTransmission* und *StopTransmission* gestartet und gestoppt werden. Die Datenübertragung eines einzelnen Messwertes wird mit *GetValue* ausgelöst.

Der mit *StartTransmission* und *StopTransmission* eingestellte Zustand ist flüchtig und wird beim Einschalten zurückgesetzt.

Das Verhalten des Messverstärkers bezüglich der Datenübertragung nach dem Einschalten kann mit SetTXmode gesetzt werden (Messverstärker sendet permanent Daten = default; oder sendet keine Daten).

Die Anzahl der Messwerte-Datenframes, die intern erfasst und pro Sekunde übertragen werden (falls permanete datenübertragung aktiv) wird mit *WriteDataRate* eingestellt.

2 Mit Ausnahme der Kommandos ResetDevice (No. 0x78) und GetValue (0x3B), siehe dort.

Nullpunkt verschieben

Die aktuelle Messwertanzeige lässt sich durch Anwendung des Kommandos *SetZero* auf 0 verschieben. Diese Funktion wird auch durch den Steuereingang „Tara“ ausgelöst.

Beim GSV-6 kann über den Parameter des Kommandos *WriteZero* festgelegt werden, auf welchen Wert die Messwertanzeige bei Anwendung des Kommandos *SetZero* verschoben werden soll. Mit *ReadZero* lässt sich dieser Parameter abfragen.

Die Funktion *SetZero* wirkt unmittelbar auf die „Rohwerte“ des Analog-Digital-Umsetzers und somit noch vor jeglicher Weiterverarbeitung der Messwerte durch Anwendung von Benutzer-Skalierungen, wie zum Beispiel *SetUserOffset* und *SetUserScale*.

Skalierung einstellen

Mit den Kommandos *WriteUserOffse*t und *WriteUserScale* kann die Skalierung der Messwerte auf physikalische Werte eingestellt werden; diese Einstellung hat keine Auswirkung auf den/die Analogausgang(e).

Die Gesamtskalierung wird außerdem durch die Eingangsbeschaltung des Messverstärkers beeinflusst. Beim GSV-6 kann die Eingangsempfindlichkeit des Messverstärkers über das Kommando *MEwriteInputRange* gesetzt werden, zusätzlich kann mit *WriteAoutScale* die Eingangsverstärkung stufenlos angepasst werden, so dass die resultierende Eingangsempfindlichlkeit <Physikalischer Eingangsbereich / OverallScaling ist.
Beim GSV-8 wird zum Ändern des Eingangstyps *SetInputType* verwendet.

Die Rohmesswerte werden stets auf einen Bereich von ±1 normiert übertragen. Der Wert 1 entspricht 100% der eingestellten Eingangsempfindlichkeit bzw. des eingestellten elektrischen Eingangsbereiches (zum Beispiel +10V oder 2mV/V). Bei einer Eingangsempfindlichkeit von zum Beispiel 2 mV/V muss der Skalierungsfaktor mit *WriteUserScale* auf 2 eingstellt werden, um eine richtige Anzeige in mV/V zu erhalten.

Der Skalierungsfaktor kann mit *WriteUserScale* so eingestellt werden, dass mit dem angeschlossenen Sensor physikalisch richtig skalierte Messwerte ausgegeben werden, da der auf ±1 normierte Rohwert hiermit multipliziert wird:

Physikalischer Messwert mit Datentyp Float = Rohwert * Skalierungsfaktor

Zur Skalierung von Mehrkomponentensensoren (Kraft-/Drehmoment Sensoren, Force-/Torque Sensoren, F/T Sensoren) führt der Messverstärker eine Matrizenmultiplikation durch. Die Matrizenmultiplikation wird aktiviert, indem ein Bit in einem Modus Register mit dem Befehl *SetMode* gesetzt wird, nachdem die Kalibrierdaten im Gerät gespeichert wurden. Wenn der Modus für F/T Sensoren aktiviert ist, sind die Skalierungen *WriteUserOffset* und der durch *WriteUserScale* gespeicherte Offset für einachsige Sensoren unwirksam. Das Nullsetzen mit *SetZero* ist weiterhin wirksam, sollte dann allerdings auf alle Kanäle angewendet werden. Beim GSV-8 wird das Kommando bei Anwendung auf einzelne Kanäle 1 bis 6 dann abgewiesen.

Das Setzen der Einheit mit *SetUnitNo* hat grundsätzlich keinen Einfluss auf die Skalierung, sondern ist ausschließlich für die Darstellung in der Anzeige von Bedeutung.

Beim GSV-8 erfolgt die Skalierung des Analogausgangs nach der Skalierung der Messwerte; sie kann mit dem Kommando *WriteAoutScale* gesetzt werden.

Schnittstellen konfigurieren

Der CANbus wird mit dem Kommando *SetCANSetting* konfiguriert.

Sonderfunktionen konfigurieren

Es stehen eine Reihe von Sonderfunktionen zur Verfügung, wie zum Beispiel die Konfiguration eines IIR Filters vierter Ordnung, das Auslesen von Fehlerspeichern, eine Rauschunterdrückung im Bereich des Nullwertes, das Auslesen von Minimal- und Maximalwertspeichern, und vieles mehr.

Zugangsschutz, Schreibschutz

Einige Befehle haben für ihr Funktionieren das Setzen der Benutzer-ID, d.h. des richtigen Passwortes zur Voraussetzung. Dies geschieht mit dem Kommando *MEsetID (0x19)*

In der Beschreibung ist dies dann vermerkt ("Benutzer-ID erforderlich"). Beim GSV-8 kann dieser Benutzer-ID-String mit dem Kommando *Set Password* (0x58) geändert werden.

Ferner können mit dem Kommando *Switch Blocking* (0x92) alle Schreibbefehle gesperrt oder entsperrt werden.[3] Das Entsperren ist nur mit gesetzter Benutzer-ID möglich. Darüber hinaus kann beim GSV-8 ein Schreibbefehl abgewiesen werden, weil ein anderes Kommunikationsinterface das Schreibrecht hat. Dies ist z.B. bei Geräten mit Feldbus (EtherCat, CANopen) dann der Fall, wenn der Feldbus aufgrund des „States“ das Schreibrecht besitzt (z.B. ab Pre-Operational State).

Bei GSV-8 Modellen mit mehreren seriellen Schnittstellen, die gleichzeitig verwendet werden, d.h. UART und USB, gilt in Bezug auf Schreibbefehle (markiert mit "W" in den Beschreibungen) folgendes:
Dasjenige Interface, mit dem *zuerst* ein Schreibbefehl ausgeführt wird, hat das Schreibrecht. An der jeweils anderen Schnittstelle werden Schreibbefehle dann mit ERR_ACC_PORT (Code 0x75) abgewiesen. Eine Kommunikationssession kann mit Aufruf des Befehls ReleaseInterface (No. 0x7A) beendet werden, so dass die andere Schnittstelle dadurch Schreibrecht erhält, falls ihr dies vorher verwehrt war.

3 Der Schreibschutz ist beim GSV-6 erst ab Firmware-version 3.30 sinnvoll nutzbar

Befehlsbeschreibungen

ResetStatus (0x00)

0x00	W / A	ResetStatus	GSV-6 / GSV-8
Alle Protokoll-Fehlerzustände löschen. GSV-8: Fehler der Messapplikation (siehe *GetLastValueError)* werden ebenfalls gelöscht, aber nur dann, wenn deren Ursache nicht mehr vorliegt.			
0 Byte	0 Parameter		
0 Byte	0 Rückgaben		

GetInterface (0x01)

0x01	R	GetInterface	GSV-6 / GSV-8
Abfrage der Schnittstellenbeschreibung. Mit diesem Befehl wird nicht nur die aktuelle Schnittstelle u.a. abgefragt, sondern mit einem Parameter auch die gewünschte Messdatenübertragung gesteuert.			
1 Byte	1 Parameter		
0	uint8_t (1)	Request-Flags	<1:0> Einzustellende Messdatenübertragung. - 0b00: Keine Änderung • 0b01: Übertragung AUS (OFF) • 0b10: Übertragung AN (ON) <2>: =1: Erlaube High-Speed Messdatenframes =0: Nur normale frames; s.S.14 <3>: =1: Messwertframes mit CRC16 Checksumme. =0: Keine CRC <7:4> Reservierte Bits (=0)
4 Bytes	4 Rückgaben		
0	uint8_t (1)	Protokoll und Gerätemodel	<7:6> Protokoll-Typ: 0b01: Messdatenframe ohne CRC 0b11: Messdatenframe mit CRC16 <5:0> Gerätemodel - 0x00: Unbekannt - 0x06: GSV-6 - 0x08: GSV-8
1	uint8_t (1)	Messwertframe-Info	<7:4> Anzahl der Messwertframe-Objekte minus 1 <3> Aktuelle Einstellung zur Messdatenübertragung (ON/OFF) <2:0>Messwertdatentyp (siehe Datentyp) - 0x01: int16 - 0x02: S24 (nur GSV-8) - 0x03: float
2	uint8_t (1)	Schreibschutz	<7> =1: Schnittstellen-spezifischer

0x01	R	GetInterface	GSV-6 / GSV-8
			Schreibschutz aktiv <6> =1: Genereller-Schreibschutz ist aktiv. <5:0> Nummer der Schnittstelle, über die gerade kommuniziert wird (0...N-1)
3	uint8_t (1)	Deskriptor Anzahl	Anzahl der vorhandenen Schnittstellen N. Bestimmt auch den Indexbereich der "Basic settings" für die Kommandos 0x7B/0x7C. Wenn =0: Diese Kommandos werden (noch) nicht unterstützt.

ReadZero (0x02)

0x02	R	ReadZero	GSV-6 / GSV-8
Aktuellen Tara-Wert auslesen			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Abzufragender Kanal GSV-6: [1-6] GSV-8: [1-8]
4 Bytes	1 Rückgaben		
0	int32_t (4)	Tara-Wert	Aktueller Tara-Wert Anmerkung: GSV-6 besitzt ‚nur' einen 16-Bit Tara-Wert, der hier auf 32-Bit erweitert zurückgegeben wird, beim GSV-8 wird von 24 auf 32 Bits sign-extended.

WriteZero (0x03)

0x02	W	WriteZero	GSV-6 / GSV-8
Neuen Tara-Wert setzen GSV-8: Voraussetzung: Benutzer-ID erforderlich			
5 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal GSV-6: [0-6] GSV-8: [0-8] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.
1	int32_t (4)	Tara-Wert	Neuen Tara-Wert schreiben. Anmerkung: GSV-6 nutzt nur die niederwertigeren 16Bits. GSV-8: 24 Bits.
0 Byte	0 Rückgaben		

ReadAoutOffset (0x04)

0x04	R	ReadAoutOffset	GSV-6 / GSV-8
Auslesen des Analogausgangsoffset (in Prozent)			
1 Byte	1 Parameter		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x04	R	ReadAoutOffset	GSV-6 / GSV-8
0	uint8_t (1)	Kanal	Abzufragender Kanal GSV-6: [1] GSV-8: [1-8]
4 Bytes	1 Rückgaben		
0	float (4)	Offset-Wert	Analogausgangsoffset im Bereich: -60.0 bis 60.0

WriteAoutOffset (0x05)

0x04	W	WriteAoutOffset	GSV-6 (Reset) / GSV-8
Setzen eines Analogausgangsoffset (in Prozent)			
5 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal GSV-6: [0-1] GSV-8: [0-8] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.
1	float (4)	Offset-Wert	Analogausgangsoffset im Bereich: -60.0 bis 60.0
0 Byte	0 Rückgaben		

ReadAoutScale (0x06)

0x06	R	ReadAoutScale	GSV-6
Auslesen des Scaling-Wertes			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Abzufragender Kanal[1-6]
4 Bytes	1 Rückgaben		
0	float (4)	Scale-Wert	Aktueller Scaling-Wert, der die ganze Meßverarbeitungskette betrifft, d.h. auch den analogen Ausgang

0x06	R	ReadAoutScale	GSV-8
Auslesen der Skalierung des analogen Ausgangs			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Abzufragender Kanal [1-8]
4 Bytes	1 Rückgaben		
0	float (4)	Scale-Wert	Aktuelle Skalierung, die nur auf den analogen Ausgang wirkt.

WriteAoutScale (0x07)

0x07	W	WriteAoutScale	GSV-6
Setzen des Scaling-Wertes			
5 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal [0-6] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.

0x07	W	WriteAoutScale	GSV-6
1	float (4)	Scale-Wert	Neuer Wert für die Eingangsskalierung, der die ganze Meßverarbeitungskette betrifft, d.h. auch den analogen Ausgang
0 Byte	0 Rückgaben		

0x07	W	WriteAoutScale	GSV-8
Setzen der Skalierung des analogen Ausgangs			
5 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal [0-8] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.
1	float (4)	Scale-Wert	Neue Skalierung, die nur auf den analogen Ausgang wirkt.
0 Byte	0 Rückgaben		

WriteAoutDirect (0x08)

0x08	W	WriteAoutDirect	GSV-8
Ausgang auf einen Wert setzen. Diese Befehlt funktioniert nur, wenn das Gerät mit dem Befehl SetAoutType (0x0E) in den Direkt-Modus versetzt wurde.			
3 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal GSV-8: [1-8]
1	uint16_t (2)	DAC-Code	Wert, der direkt in das DAC-Register geschrieben wird.
0 Byte	0 Rückgaben		

Load Config (0x09)

0x09	W / A	GetAll	GSV-6 / GSV-8
Die Betriebsparameter des Geräts aus dem angegebenen Slot lesen und einstellen. ***Achtung***: Beim GSV-6 sind die gelesenen Einstellungen **temporär** und werden **nicht** automatisch als aktuelle Einstellung gespeichert. Zum Sichern der Einstellungen ist beim GSV-6 ein SaveAll mit Datensatz 0 notwendig!			
1 Byte	1 Parameter		
0	uint8_t (1)	Daten Satz-Nr.	Einzustellender Datensatz GSV-6: [0-1] GSV-8: [0-7] [0]: Aktuelle (GSV-8: zuletzt gespeicherte) Einstellung [1]: Werkseinstellung (default) [2-7]: Benutzer-Speicherplätze.
0 Byte	0 Rückgaben		

Eine Liste der Parameter, die mit LoadConfig (GetAll) geladen werden, ist im Anhang F, S. 101. Mit Datensatz-Nr. =1 werden die dort genannten Defaultwerte gesetzt.

Store Config (0x0A)

0x0A	W	SaveAll	GSV-6 / GSV-8
Die Betriebsparameter des Geräts in den angegebenen Slot speichern.			
1 Byte	1 Parameter		
0	uint8_t (1)	Daten Satz-Nr.	Einzustellender Datensatz GSV-6: [0-1] GSV-8: [0-7] [0]: die aktuelle Einstellung [1]: die Werkseinstellung [2-7]: Benutzer-Speicherplätze. Die Datensätze sind nicht alle frei für den Benutzer einstellbar. So ist die Werkseinstellung [1] nur mit entsprechender Freigabe beschreibbar (GSV-8).
0 Byte	0 Rückgaben		

Eine Liste der Parameter, die mit StoreConfig gespeichert werden, ist im Anhang F, S. 101

SetZero (0x0C)

0x0C	W / A	SetZero	GSV-6 / GSV-8
Tara auf den aktuellen Eingangswert setzten, so dass der Ausgabewert Null wird.			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal GSV-6: [0-6/7] GSV-8: [0-8/10] Mit Kanal 0 werden ALLE Kanäle Null gesetzt.
0 Byte	0 Rückgaben		

GetAoutType (0x0D)

0x0D	R	GetAoutType	GSV-8 / GSV-6
Abfrage des Typs eines analogen Ausgangs. GSV-6: Vorhanden ab FW-ver 3.29			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Abzufragender Kanal GSV-8: [1-8] GSV-1: =1
2 Bytes	2 Rückgaben		

0x0D	R	GetAoutType	GSV-8 / GSV-6
0	uint8_t (1)	Type-Enum	Type-Enumerator 0: 0V – 10V 1: -10V – 10V 2: 0V – 5V 3: -5V – 5V 4: 4mA – 20mA 5: OFF 6: 0mA – 20mA 7: 0V - 2,5V (nur GSV-6 CPU/DEV) Andere Werte sind reserviert.
1	uint8_t (1)	Kanal-Modus	GSV-6: =0 GSV-8: <7:3> Reserviert <2> =1: Alternativer Input (Kanäle 7&8: Counter/Frequenzeingang) =0: Analogeingang <1> Ausgangskanal inaktiv==1, aktiv ==0 <0> Direktmodus aktiv - ==0: Folgt Messwert - ==1: Direkt Modus (siehe WriteAoutDirect (0x08))

SetAoutType (0x0E)

0x0E	W	SetAoutType	GSV-8 / GSV-6
Setzen des Typs und Modus eines analogen Ausgangs GSV-6: Vorhanden ab FW-ver 3.29			
3 Bytes	3 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal GSV-8: [0-8] GSV-6: 0-1 Mit Kanal 0 werden ALLE Kanäle mit dem gleichen Werten beschrieben. GSV-6: nur =1, sonst: Fehler
1	uint8_t (1)	Type-Enum	siehe GetAoutType (0x0D)
2	uint8_t (1)	Kanal-Modus	siehe GetAoutType (0x0D) (GSV-6: =0)
0 Byte	0 Rückgaben		

GetUnitNo (0x0F)

0x0F	R	GetUnitNo	GSV-6 / GSV-8
Abfrage der hinterlegten physikalischen Einheit			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Abzufragender Kanal GSV-6: [1-6] / [1-7] (FW-ver >=3.11) GSV-8: [1-10]
1 Byte	1 Rückgaben		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x0F	R	GetUnitNo	GSV-6 / GSV-8
0	uint8_t (1)	PhysUnit-Enum	Enumerator-Wert der physikalischen Einheit, siehe Physikalische Einheiten (Codes)oder 0xFE,FF: Freitext 1,0, s. GetUnitText (0x11)

SetUnitNo (0x10)

0x10	W	SetUnitNo	GSV-6 / GSV-8
Setzen der Kanalspezifischen physikalischen Einheit			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal GSV-6: [0-6] / [0-7] (FW-ver >=3.11) GSV-8: [0-10] Mit Kanal 0 werden ALLE Kanäle mit dem identischen Wert beschrieben.
0	uint8_t (1)	PhysUnit-Enum	Enumerator der physikalischen Einheit siehe Physikalische Einheiten (Codes) oder 0xFE,FF: Freitext 1, 0, s. 34
0 Byte	0 Rückgaben		

GetUnitText (0x11)

0x11	R (Seriell)	GetUnitText	GSV-6
Abfrage der Freitext-Einheit Achtung: Bei diesem Befehl gibt es eine Unterscheidung, die Schnittstellen-Spezifisch ist.			
1 Byte	1 Parameter		
0	uint8_t (1)	Slot	Text-Slot [0-1]
9 Bytes	2 Rückgaben		
0	uint8_t (1)	Indikator	Bei Seriell immer ==0
1	char[8]	Einheiten-Text	Der hinterlegte Freitext (NULL-terminiert oder genau 8 Zeichen lang)

0x11	R (CAN)	GetUnitText	GSV-6
Abfrage der Freitext-Einheit ***Achtung***: Es werden auf eine Anfrage **zwei** Antwort-Paket versendet. Und bei diesem Befehl gibt es eine Unterscheidung, die Schnittstellen-Spezifisch ist.			
1 Byte	1 Parameter		
0	uint8_t (1)	Slot	Text-Slot [0-1]
5 Bytes	2 Rückgaben		
0	uint8_t (1)	Indikator	==1 ‚Erste Hälfte' ==2 ‚Zweite Hälfte'
1	char[4]	Einheiten-Text	Der hinterlegte Freitext.

0x11	R	GetUnitText	GSV-8
Abfrage der Freitext-Einheit			

0x11	R	GetUnitText	GSV-8
1 Byte	1 Parameter		
0	uint8_t (1)	Slot	Text-Slot [0-1]
8 Bytes	1 Rückgabe		
1	char[8]	Einheiten-Text	Der hinterlegte Freitext (NULL-terminiert oder genau 8 Zeichen lang)

SetUnitText (0x12)

0x12	W (Seriell)	SetUnitText	GSV-6
Setzen einer Freitext-Einheit (maximal 8 Zeichen) Ähnlich wie bei der Abfrage ist auch beim Setzen ein Unterschied.			
10 Bytes	3 Parameter		
0	uint8_t (1)	Slot	Text-Slot [0-1]
1	uint8_t (1)	Indikator	Bei Seriell immer ==0
2	char[8]	Einheiten-Text	Der zu setzende Freitext (NULL-terminiert oder genau 8 Zeichen lang).
0 Byte	0 Rückgaben		

0x12	W (CAN)	SetUnitText	GSV-6
Setzen einer Freitext-Einheit (maximal 8 Zeichen) Es **muss** immer zuerst ein Befehl mit der Ersten Hälfte und dann mit der zweiten Hälfte versendet werden, damit der Text gespeichert wird! Wird nur die erste Hälfte versendet verweilt diese in einem temporären Speicher. Wird nur die zweite Hälfte versendet gibt es eine Fehlermeldung.			
6 Bytes	3 Parameter		
0	uint8_t (1)	Slot	Text-Slot [0-1]
1	uint8_t (1)	Indikator	==1 ‚Erste Hälfte' ==2 ‚Zweite Hälfte'
2	char[4]	Einheiten-Text	Der zu setzende Freitext.
0 Byte	0 Rückgaben		

0x12	W (Seriell)	SetUnitText	GSV-8
Setzen einer Freitext-Einheit (maximal 8 Zeichen)			
9 Bytes	3 Parameter		
1	uint8_t (1)	Slot	Text-Slot [0-1]
2	char[8]	Einheiten-Text	Der zu setzende Freitext (NULL-terminiert oder genau 8 Zeichen lang).
0 Byte	0 Rückgaben		

ReadUserScale (0x14)

0x14	R	ReadUserScale	GSV-6 / GSV-8
Auslesen des vom Benutzer einstellbaren Skalierungsfaktors			

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x14	R	ReadUserScale	GSV-6 / GSV-8
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Abzufragender Kanal GSV-6: [1-6] / [1-7] (FW-ver >=3.11) GSV-8: [1-10]
4 Bytes	1 Rückgaben		
0	float (4)	UserScale Faktor	Eingestellter Skalierungsfaktor des Float-Messwertes an der Schnittstelle

WriteUserScale (0x15)

0x14	W	WriteUserScale	GSV-6 / GSV-8
Setzen des vom Benutzer einstellbaren Skalierungsfaktors **Hinweis**: Wenn die 6-Achsen-Berechnung aktiv ist, haben diese Werte keine Auswirkung!			
5 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal GSV-6: [0-6] / [0-7] (FW-ver >=3.11) GSV-8: [0-10] Mit Kanal 0 werden ALLE Kanäle mit dem identischen Wert beschrieben.
1	float (4)	UserScale Faktor	Einzustellender Skalierungsfaktor des Float-Messwertes an der Schnittstelle
0 Byte	0 Rückgaben		

ReadCal (0x17)

0x17	R	MEreadCal	GSV-6
Auslesen des Kalibrierwerts. Der Kalibrierkanal ist die Kanal-Nummer zur „Basis 0“ und ein Offsetwert in die entsprechende Kalibrier-Datenreihe. ***Achtung***: Dieser Befehl erfordert eine MESYS-Freigabe!			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kalibrierkanal	0x00-0x05: Eingangsfaktor 0x08-0x0D: Eingangsoffset 0x10: Ausgangsfaktor 0x18: Ausgangsoffset 0x20-0x25: Scale-Werte (Faktor) 0x28-0x2D: Tara-Werte (Offset) Die anderen Werte sind reserviert!

0x17	R	MEreadCal	GSV-6
1	uint8_t (1)	SubIndex	Eingangsfaktor/-offset: für Vorverstärkung: - 0x00: Verstärkung 1x - 0x01: Verstärkung 2x - 0x02: Verstärkung 4x - 0x03: Verstärkung 8x Ausgangsfaktor/-offset für Ausgangstyp: - 0x00: 0V – 10V - 0x01: -10V – 10V - 0x02: 0V – 5V - 0x03: -5V – 5V - 0x04: 4mA – 20mA - 0x05: reserviert - 0x06: 0mA – 20 mA - 0x07: 0V-2.5V Scale- / Tara-Werte - ungenutzt Die anderen Werte sind reserviert!
4 Bytes	1 Rückgaben		
0	float (4) / int32_t (4)	Faktor Offset	

0x17	R	MEreadCal	GSV-8
Auslesen des Hersteller-Kalibrierwerts.			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	[1-8] Eingangs bzw. Ausgangskanal

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x17	R	MEreadCal	GSV-8
1	uint8_t (1)	SubIndex	- 0x00: Scalecal f. Brückeneingang Us= 8,75V - 0x01: Scalecal f. Brückeneingang Us= 5V - 0x02: Scalecal f. Brückeneingang Us= 2,5V - 0x03: ScaleCal f. SingleEnded Eingang +-10V - 0x04: ScaleCal f. PT1000-Eingang - 0x10: ScaleCal f. Analogausgang 10V - 0x11: OffsetCal f. Analogausgang 10V - 0x12: ScaleCal f. Analogausgang 5V - 0x13: OffsetCal f. Analogausgang 5V - 0x14: ScaleCal f. Analogausgang 4..20mA - 0x15: OffsetCal f. Analogausgang 4..20mA - 0x20: OffsetCal f. Brückeneingang Us= 8,75V - 0x21: OffsetCal f. Brückeneingang Us= 5V - 0x22: OffsetCal f. Brückeneingang Us= 2,5V - 0x23: OffsetCal f. SingleEnded Eingang +-10V - 0x24: OffsetCal f. PT1000-Eingang Andere Werte sind reserviert!
4 Bytes	1 Rückgaben		
0	float (4)	Faktor / Offset	

MEwriteCal (0x18)

0x18	W	MEwriteCal	GSV-6 (Reset)
Setzen eines Kalibrierwerts. Der Kalibrierkanal ist die Kanal-Nummer zur „Basis 0“ und ein Offsetwert in die entsprechende Kalibrierdatenreihe. ***Achtung***: Dieser Befehl erfordert eine MESYS-Freigabe!			
6 Bytes	3 Parameter		
0	uint8_t (1)	Kalibrierkanal	Siehe ReadCal (0x17)
1	uint8_t (1)	SubIndex	Siehe ReadCal (0x17)
2	float (4) / int32_t (4)	Faktor Offset	
0 Byte	0 Rückgaben		

0x18	W	MEwriteCal	GSV-8
Setzen eines Kalibrierwerts. ***Achtung***: Dieser Befehl erfordert eine MESYS-Freigabe!			
6 Bytes	3 Parameter		
0	uint8_t (1)	Kalibrierkanal	Zu setzender Eingangs bzw. Ausgangskanal [0-8] Mit Kanal 0 werden ALLE Kanäle mit dem identischen Wert beschrieben.
1	uint8_t (1)	SubIndex	Siehe ReadCal (0x17)
2	float (4)	Faktor / Offset	
0 Byte	0 Rückgaben		

MEsetID (0x19)

0x19	A	MEsetID	GSV-6 / GSV-8
Setzen eines Freischalt-Codes / Setzen des ID-Levels, den einige Schreibbefehle benötigen. Nur die aufgeführten Codes werden ohne Fehler akzeptiert. Wird ein fehlerhafter Code übergeben wird nach drei Fehlversuchen immer ein Fehler zurückgegeben. Die Freigabe ist bis zu einem Neustart aktiv.			
4 Bytes	1 Parameter		
0	uint32_t (4)	Freischalt-Code	ID_EXT_MESYS ID_EXT_USER_CONF (Benutzer-ID) ID_EXT_USER_CONF_ALL (reserviert)
0 Byte	0 Rückgaben		

Die Freischaltcodes sind wie folgt definiert:

```
ID_EXT_MESYS
Beim GSV-8 kann ID_EXT_USER_CONF mit dem Befehl SetPassword (0x58) geändert
werden
ID_EXT_USER_CONF      GSV-8: (0x42656C6E) = "Beln" (Default)
ID_EXT_USER_CONF      GSV-6: (0x55534331) = "USC1"
```

MEgetIDstate (0x1A)

0x1A	R	MEgetIDstate	GSV-6 / GSV-8
Abfragen des ID-States			
0 Byte	0 Parameter		
1 Byte	1 Rückgaben		
0	uint8_t (1)	Freischalt-Zustand	0x00: Normalbetrieb 0x08: Privilegierte Befehle nutzbar d.h. Benutzer-ID gegeben 0x10: ME-Befehle nutzbar

Read Sensor Model (0x1B)

0x1B	R	ReadSensorModel	GSV-8
Lesen des Modellnamens des Mehrachsensensors. Vorhanden ab Firmware-version 1.39			
0 Byte	0 Parameter		
1..15 Bytes	1 Rückgabe		
0	String(1..15)	Modell-Name	Null-terminierter String oder genau 15 Bytes lang (dann ohne Terminierung)

Write Sensor Model (0x1C)

0x1C	W	WriteSensorModel	GSV-8
Schreiben des Modellnamens des Mehrachsensensors. Vorhanden ab Firmware-version 1.39 Voraussetzung: Benutzer-ID erforderlich			
1..15 Bytes	1 Parameter		
0	String(1..15)	Modell-Name	Null-terminierter String oder genau 15 Bytes lang (dann ohne Terminierung)
0 Byte	0 Rückgaben		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Read Calibration Operator Name (0x1D)

0x1D	R	ReadCalOpName	GSV-8
Lesen Namens der Institution (mit Mitarbeiter-Kürzel) der letzten Geräte-Kalibrierung. Vorhanden ab Firmware-version 1.40			
0 Byte	0 Parameter		
8 Bytes	1 Rückgabe		
0	String(8)	Operator-Name	Genau 8 Bytes lang (dann ohne Terminierung) oder mit 0-Bytes auf 8 bytes Länge aufgefüllter String.

Write Calibration Operator Name (0x1E)

0x1E	W	WriteCalOpName	GSV-8
Schreiben des Modellnamens des Mehrachsensensors. Vorhanden ab Firmware-version 1.40 ***Achtung***: Dieser Befehl erfordert eine MESYS-Freigabe!			
8 Bytes	1 Parameter		
0	String(8)	Operator-Name	Genau 8 Bytes lang (dann ohne Terminierung) oder mit 0-Bytes auf 8 bytes Länge aufgefüllter String.
0 Byte	0 Rückgaben		

GetSerNo (0x1F)

0x1F	R	GetSerNo	GSV-6 / GSV-8
Abfrage der Seriennummer			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		
0	uint32_t (4)	Seriennummer	[1-99999999]. GSV-6 liefert 0xFFFFFFFF oder 0, wenn noch keine Seriennummer gesetzt ist, GSV-8 dann d01234567

MEsetSerNo (0x20)

0x20	W	MEsetSerNo	GSV-6 / GSV-8
Setzen der Seriennummer ***Achtung***: Dieser Befehl erfordert eine MESYS-Freigabe!			
4 Bytes	1 Parameter		
0	uint32_t (4)	Seriennummer	[1-99999999]
0 Byte	0 Rückgaben		

Read Calibration Date (0x21)

0x21	R	ReadCalDate	GSV-8
Lesen des Datums der letzten Geräte-Kalibrierung. Vorhanden ab Firmware-version 1.40			
0 Byte	0 Parameter		
4 Bytes	4 Rückgaben		

0x21	R	ReadCalDate	GSV-8
0	uint8_t(1)	Jahr	Jahr der letzten Kalibrierung seit 2000, d.h. dem Wert muss 2000 hinzuaddiert werden, um das tatsächliche Jahr zu erhalten.
1	uint8_t(1)	Monat	Monat der letzten Kalibrierung (1..12)
2	uint8_t(1)	Tag	Tag des Monats der letzten Kalibrierung (1..31)
3	uint8_t(1)	Stunde	Stunde des Tages der letzten Kalibrierung (0..23)

Write Calibration Date (0x22)

0x22	W	WriteCalDate	GSV-8
Schreiben des Datums der letzten Geräte-Kalibrierung. Vorhanden ab Firmware-version 1.40 ***Achtung***: Dieser Befehl erfordert eine MESYS-Freigabe!			
4 Bytes	4 Parameter		
0	uint8_t(1)	Jahr	Jahr der letzten Kalibrierung seit 2000, d.h. dem Wert muss 2000 hinzuaddiert werden, um das tatsächliche Jahr zu erhalten.
1	uint8_t(1)	Monat	Monat der letzten Kalibrierung (1..12)
2	uint8_t(1)	Tag	Tag des Monats der letzten Kalibrierung (1..31)
3	uint8_t(1)	Stunde	Stunde des Tages der letzten Kalibrierung (0..23)
0 Byte	0 Rückgaben		

StopTransmission (0x23)

0x23	A	StopTrasmission	GSV-6 / GSV-8
Permanente, selbstständige Messdatenübertragung für diese Schnittstelle stoppen. Dieser Zustand ist flüchtig, d.h. er bleibt nach dem Neustart nicht erhalten. Zum dauerhaften Stoppen der permanenten Messdatenübertragung bitte den Befehl SetTXmode verwenden.			
0 Byte	0 Parameter		
0 Byte	0 Rückgaben		

StartTransmission (0x24)

0x24	A	StartTrasmission	GSV-6 / GSV-8
Permanente, selbstständige Messdatenübertragung für diese Schnittstelle starten. Dieser Zustand ist flüchtig, d.h. er bleibt nach dem Neustart nicht erhalten. Zum dauerhaften Starten der permanenten Messdatenübertragung bitte den Befehl SetTXmode verwenden, falls er hiermit als gestoppt konfiguriert wurde.			
0 Byte	0 Parameter		
0 Byte	0 Rückgaben		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

ClearBufferAbortTX (0x25)

0x25	A	ClearBufferAbortTX	GSV-8 (USB)
Übertragungspuffer löschen (USB). Es werden im USB-Sendequeue der Messanwendung befindliche Messdatenframess aus diesem entfernt. Eine "pending transmission" auf USB-Ebene wird jedoch nicht abgebrochen, so dasss Messdatenframes, die zu diesem Zeitpunkt gerade gesendet werden, erhalten bleiben (es entsteht kein "Datenmüll").			
0 Byte	0 Parameter		
0 Byte	0 Rückgaben		

GetMode (0x26)

0x26	R	GetMode	GSV-6
Mode-Flags auslesen. Beim GSV-6 handelt es sich hierbei um die System-Flags.			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		
0	uint32_t (4)	Mode-Flags	GSV-6: <2:0> Anzahl der aktiven Kanäle. 0b001: 1 Kanal ... 0b111: 7 Kanäle <3> Save-Tara <4> User-Monitor <5> Offset aus TEDS laden (ab FW 3.19) <6> Scale-Negate <7> Spitzenwert-Ausgabe <8> Spitzenwert-Reset und Tara <9> TEDS Auslesen beim Boot <10> 6-Achsen Berechnung <11> ClickRclackR Menüausgabe 0-20mA, wenn Analogausgangstyp = Strom[4] Alle anderen Bits sind reserviert.

0x26	R	GetMode	GSV-8
Mode-Flags auslesen. Beim GSV-8 sind es reine Mode-Flags die vom User beschrieben werden können.			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		

4 Ab Firmware Version 3.29

0x26	R	GetMode	GSV-8
0	uint32_t (4)	Mode-Flags	GSV-8: <**0**> 6-Achsen Berechnung aktiv <**1**> Analogfilter automatisch setzen <**2**> Maximal- und Minimalwert ermitteln <**3**> Rauchunterdrückung aktiv <**7**> Alle Schreibfunktionen sperren <**15:8**>: Wenn am Eingangskanal BitNo-7 Sensoren mit TEDS angeschlossen sind, sollen deren Daten zur Verwendung des User-Scale-Wertes verwendet werden (wenn möglich) <**16**>: Wenn entsprechendes Bit in <15:8> gesetzt, soll auch die Einheit aus den TEDS-Daten eingestellt werden (wenn möglich) <**17**>: Wenn entsprechendes Bit in <15:8> gesetzt ist, soll auch die Eingangs-empfindlichkeit anhand der TEDS-Daten eingestellt werden (wenn möglich) <**18**>: Wenn entsprechendes Bit in <15:8> gesetzt ist, soll auch der Analogausgang anhand der TEDS-Daten skaliert werden (wenn möglich) <**19**>: Wenn entsprechendes Bit in <15:8> gesetzt ist, soll auch der Nullpunkt anhand der TEDS-Daten eingestellt werden (wenn möglich) <**22**>: Auto-Zero aktiv, s. Cmd Read AutoZero Setting (0x96) ff

SetMode (0x27)

0x27	W	SetMode	GSV-6 (Reset)
Mode-Flags setzen			
4 Bytes	1 Parameter		
0	uint32_t (4)	Mode-Flags	GSV-6: <5> Offset aus TEDS laden (ab FW 3.19) <6> Scale-Negate (ab FW-ver. 3.11) <7> Spitzenwert-Ausgabe (ab 3.11) <9> TEDs Auslesen beim Boot (ab 3.10) <10> 6-Achsen Berechnung <11> ClickRclackR Menüausgabe 0-20mA, wenn Analogausgang = Strom Alle anderen Bits werden ignoriert.
0 Byte	0 Rückgaben		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x27	W	SetMode	GSV-8
Mode-Flags setzen			
4 Bytes	1 Parameter		
0	uint32_t (4)	Mode-Flags	siehe GetMode (0x26) Bit 7 ist nicht veränderbar (read-only).
0 Byte	0 Rückgaben		

GetSoftwareConfiguration (0x2A)

0x2A	R	GetSoftwareConfiguration	GSV-8
Dieser Wert zeigt das Vorhandensein hardwareabhängiger Features an, die mit dementsprechenden Features der Gerätesoftware korrespondieren müssen. Deshalb wird dieser Wert (Hex-kodiert) auch in der Bootloader-Software angezeigt. Weicht er zwischen vorhandener Gerätesoftware und upzudatendem Neuprogramm ab, ist von einem Updatevorgang i.d.R. dringend abzuraten (es sei denn, er erfolgt im Zuge einer Hardwareaufrüstung in Rücksprache mit dem Hersteller).			
0 Byte	0 Parameter		
4 Bytes	1 Rückgabe		s. Tabelle

Tabelle: Bedeutung BUILD-Val

Bit	Wert	Name	Bedeutung (GSV-8)
0	1	HAS_ADC	AD-Umsetzer vorhanden
1	2	HAS_ETHERCAT	Ethercat-Feldbus vorhanden
2	4	HAS_LCD	LC-Display vorhanden
3	8	HAS_TEDS	TEDS-Sensor-Unterstützung
4	16	HAS_DIGI_IN_OUT	Digitale Ein- und Ausgänge vorhanden
5	32	HAS_ETH_TWOLEDS	Bei EtherCAT: Run/Err-LED getrennt
6	64	HAS_ANALOG_OUT	Analogausgänge vorhanden
7	128	HAS_SERIAL	Serielles Interface vorhanden (z.B. via Ethernet)
8	256	HAS_FREQ_OUT	Frequenzmodulierter Analogausgang vorhanden
9	512	HAS_AIN_MCU	Sensorüberwachung am Analogeingang
10	1024	HAS_SIXAXIS	Sechsachsensensoren werden unterstützt
11	2048	HAS_CANOPEN	CANopen feldbus vorhanden
12	4096	IS_HIGHSPEED	Ist 4-Kanal Sondergerät mit Fdata,max= 96000/s

0x2A	R	GetSoftwareConfiguration	GSV-6
Dieser Wert zeigt das Vorhandensein hardwareabhängiger Features an. Vorhanden ab FW-ver 3.29			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		s. Tabelle

Tabelle: Bedeutung der Ausstattungs-Flags

Bit	Wert (hex)	Bedeutung (GSV-6)
0..3		Anzahl der Schwellwertschalter-Ausgänge
4	0x10	Signal-Conditioner f. Analogausgang vorhanden, d.h. Ausgangstypen 0..4 und 6
5	0x20	Quadratur-Encoder vorhanden, d.h. Eingangskanal 7 f. Counter/Frequenz
6	0x40	Real-Time Clock (RTC) vorhanden
8	0x100	SD-Kartenslot vorhanden
8	0x100	SD-Kartenslot vorhanden
9	0x200	CAN-Bus Transceiver vorhanden
10	0x400	GSV-6 BT Gerät
11	0x800	Embedded Linux / Handgerät
16	0x10000	reserviert (=1)
17	0x20000	Temperatursensor vorhanden
18	0x40000	I2C Bus vorhanden (CPU-Software feature)
19	0x80000	CAN Bus vorhanden (CPU-Software feature)
20	0x100000	Monitor-Interface auf RS232 Port schaltbar (CPU-Software feature)
21	0x200000	GSV-Protokoll vorhanden (CPU-Software feature, stets =1)
22	0x400000	DA-Wandler unterstützt (CPU-Software feature)
23	0x800000	SPI Port vorhanden (CPU-Software feature)
27	0x8000000	Ist Low-Power Sondergerät

FirmwareVersion (0x2B)

0x2B	R	FirmwareVersion	GSV-6 / GSV-8
Firmware Version auslesen			
0 Byte	0 Parameter		
4 Bytes	2 Rückgaben		
0	uint16_t (2)	Hauptversion	VER_MAJOR
1	uint16_t (2)	Unterversion	VER_MINOR

Get Prohibit Set Zero (0x32)

0x32	R	GetProhibutSetZero	GSV-8
Das Nullsetzen mit Cmd. 0x0C sowie per Taste oder Feldbus wird abgewiesen, wenn das dem Eingangskanal entsprechende Flag gesetzt ist (Bit 0: K1 usw bis Bit9: Counter 2) Ab FW-ver. 1.54			
0 Byte	0 Parameter		
2 Bytes	1 Rückgabe		
0	uint16_t (2)	Zero Prohibit Flags	Bits 0-9: SetZero an Kanal 1-10 verboten

Write Prohibit Set Zero (0x33)

0x33	W	WriteProhibutSetZero	GSV-8
Das Nullsetzen mit Cmd. 0x0C sowie per Taste oder Feldbus kann hiermit dauerhaft unterbunden werden. Voraussetzung: Benutzer-ID gesetzt Ab FW-ver. 1.54			

2 Bytes	1 Parameter		
0	uint16_t (2)	Zero Prohibit Flags	Bits 0-9: SetZero an Kanal 1-10 verboten
0 Bytes	0 Rückgaben		

MEwriteInputRange (0x34)

0x34	**W**	**MEwriteInputRange**	**GSV-6 (Reset)**
Eingangsempfindlichkeit setzen. ***Achtung***: GSV-6 kann die Kanäle nicht unabhängig voneinander setzen! ***Achtung***: Bei Firmware-version < 3.11 erfordert dieser Befehl eine MESYS-Freigabe!			
6 Bytes	3 Parameter		
0	uint8_t (1)	Kanal	[0] Mit Kanal 0 werden ALLE Kanäle mit dem gleichen Wert beschrieben.
1	uint8_t (1)	SensIndex	ungenutzt (Null übergeben)
2	uint32_t (4)	Empfindlichkeit	HW-Empfindlichkeit in mV*100 Mögliche Werte sind: 800: 8mV/V 400: 4mV/V 200: 2mV/V 100: 1mV/V
0 Byte	0 Rückgaben		

0x34	**W**	**MEwriteInputRange**	**GSV-8**
Herstellereinstellung: Kommunizierten Wert für die Eingangsempfindlichkeit (=Messbereich) setzen. ***Achtung***: Dieser Befehl erfordert eine MESYS-Freigabe!			
6 Bytes	3 Parameter		
0	uint8_t (1)	Kanal	[0-8] Mit Kanal 0 werden ALLE Kanäle mit dem gleichen Wert beschrieben.
1	uint8_t (1)	SensIndex	[1-4] Eingangstyp: - 0x01: Range f. Brückeneingang Us= 8,75V - 0x02: Range f. Brückeneingang Us= 5V - 0x03: Range f. Brückeneingang Us= 2,5V - 0x04: Range f. SingleEnded Eingang +-10V
2	uint32_t	Empfindlichkeit	HW-Empfindlichkeit in mV*100 Standardversion: 2mV/V = 200, 3,5 ->350 7 -> 700. +-10V -> 1000000
0 Byte	0 Rückgaben		

SetInjectValOrOffset (0x35)

0x35	A	Set Inject Val or Offset	GSV-8
Eingangswerte-Simulation einstellen oder Nullsignaloffsetwerte laden			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	**0:** Zurücksetzen aller Inject Values oder Offsets zum Normalbetrieb **1:** Der halbe Nennaussteuerungswert (Rohwert: 0x003CF3CF) wird bei allen Kanälen als Pseudo-Messwert gesetzt, zugleich wird die Messanwendung vom AD-Umsetzer abgekoppelt. **2:** Die nach dem letzten Nullsetzen ermittelten Offsetwerte werden nicht verwendet; stattdessen werden die für exakt 0mV/V (Brückeneingang) bzw. 0V geltenden Offset-Hersteller-Kalibrierwerte des Messverstärkers geladen. Bei angeschlossenem Sensor unter Nulllast kann so dessen Grundverstimmung abgelesen werden (der AD-Umsetzer ist also weiterhin aktiv). **3:** Ähnlich wie 2., es werden allerdings die Offset-Kalibrierwerte des Sechsachsensensors geladen (sofern Sechsachsenkalibrierdaten vorhanden). Somit kann unter Nulllast des Sechsachsensensors die Abweichung seiner Grundverstimmung von der des Zeitpunktes seiner Kalibrierung - und somit ein Indikator für dessen Abnutzung - abgelesen werden.
0 Byte	0 Rückgaben		

GetHardwareVersion (0x36)

0x36	R	GetHardwareVersion	GSV-6 / GSV-8
Hardware Option auslesen (GSV-6: Liefert hier z.Z. immer 0!)			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		
0	uint32_t (4)	Optionen (GSV-8)	Bits<31:16>: Hardware-Versionierung des Hauptboards. "Versioniert" werden Hardware-Releases, die einer angepassten, bzw. geänderten Geräte-Software bedürfen. Bits<15:8>: Unteres Byte der Hardwareversion. Vorgesehen f. Geräte-Software relevante Versionierung kommender Tochterboardversionen. Bits<7:4>: reserviert Bits<3:0>: Versionsnummer des Co-Prozessors.

GetCustomSerNo (0x37)

0x37	R (Seriell)	GetCustomSerNo	GSV-6
Abfrage der zweiten Seriennummer Achtung: Bei diesem Befehl gibt es eine Unterscheidung, die Schnittstellen-Spezifisch ist.			
0 Byte	0 Parameter		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x37	R (Seriell)	GetCustomSerNo	GSV-6
9 Bytes	2 Rückgaben		
0	uint8_t (1)	Indikator	Bei Seriell immer ==0
1	char[8]	Ser.Nr.-Text	Der hinterlegte Ser.Nr. als ASCII-Text (NULL-terminiert oder genau 8 Zeichen lang)

0x37	R (CAN)	GetCustomSerNo	GSV-6
Abfrage der zweiten Seriennummer ***Achtung***: Es werden auf eine Anfrage **zwei** Antwort-Paket versendet. Und bei diesem Befehl gibt es eine Unterscheidung, die Schnittstellen-Spezifisch ist.			
0 Byte	0 Parameter		
5 Bytes	2 Rückgaben		
0	uint8_t (1)	Indikator	=1 ‚Erste Hälfte' =2 ‚Zweite Hälfte'
1	char[4]	Ser.Nr.-Text	Der hinterlegte Ser.Nr. als ASCII-Text (NULL-terminiert oder genau 8 Zeichen lang)

Die Befehle 0x37-0x39 sind erst ab Firmware-Version 3.33 vorhanden. Wenn keine zweite Ser.No gespeichert ist, wird ein Leerstring ausgegeben (alle Textbytes =0). Wenn eine gespeichert ist, so wird sie auch mit dem Befehl GetSerNo 0x1F ausgegeben, sofern alle Bytes der Custom-Nr. als ASCII-Zahlen gespeichert wurden (Big-Endian). Mit GetManufacturerSerNo 0x39 wird stets die erste Ser.Nr gelesen. Typische Fälle der Nutzung sind „Smart-Sensors" mit eingebautem GSV-6, bei denen die Custom Ser.Nr. der aufgedruckten Seriennummer des Produktes entspricht.

SetCustomSerNo (0x38)

0x38	W (Seriell)	SetCustomSerNo	GSV-6
Setzen der zweiten Seriennummer (maximal 8 Zeichen). Erfordert MESYS Freigabe. Ähnlich wie bei der Abfrage ist auch beim Setzen ein Unterschied.			
9 Bytes	2 Parameter		
0	uint8_t (1)	Indikator	Bei Seriell immer ==0
1	char[8]	Ser.Nr.-Text	Die zu setzende Ser.Nr. als ASCII-Text (NULL-terminiert oder genau 8 Zeichen lang)
0 Byte	0 Rückgaben		

0x38	W (CAN)	SetCustomSerNo	GSV-6
Setzen der zweiten Seriennummer (maximal 8 Zeichen) Es **muss** immer zuerst ein Befehl mit der Ersten Hälfte und dann mit der zweiten Hälfte versendet werden, damit der Text gespeichert wird! Wird nur die erste Hälfte versendet verweilt diese in einem temporären Speicher. Wird nur die zweite Hälfte versendet gibt es eine Fehlermeldung.			
5 Bytes	2 Parameter		
0	uint8_t (1)	Indikator	==1 ‚Erste Hälfte' ==2 ‚Zweite Hälfte'
1	char[4]	Ser.Nr.-Text	Die zu setzende Ser.Nr. als ASCII-Text (NULL-terminiert oder genau 8 Zeichen lang)
0 Byte	0 Rückgaben		

Dieser Befehl kann nur vom Hersteller ausgeführt werden und er ist erst ab Firmware-Version 3.33 vorhanden.

GetManufacturerSerNo (0x39)[5]

0x39	R	GetManuf.SerNo	GSV-6
Abfrage der ersten Seriennummer. Der Befehl dient zur Unterscheidung, falls eine CustomSerNo gesetzt ist. Im allgemeinen genügt der Befehl GetSerNo (0x1F) zum Lesen der Produkt-Seriennummer.			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		
0	uint32_t (4)	Seriennummer	[1-99999999]. Wenn noch keine Seriennummer gesetzt ist, liefert der Befehl 0xFFFFFFFF oder 0

GetRawValue (0x3A)

0x3A	R	GetRawValue	GSV-6 / GSV-8
Rohwert eines Eingangs einlesen			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	[1-7] / [1-10]
4 Bytes	1 Rückgaben		
0	int32_t (4)	Rohwert	Rohwert

GetValue (0x3B)

0x3B	R	GetValue	GSV-6 / GSV-8
Absenden eines Messwert-Frames anforden. ***Achtung***: Dieser Befehl liefert **keinen** Antwort-Frame, daher auch keinen Fehler-Code! Stattdessen wird bei abgeschaltete ständiger Messdatenübertragung ein Messwert-Frame versendet.			
0 Byte	0 Parameter		
0 Byte	0 Rückgaben		

ClearMaxValue (0x3C)

0x3C	A	ClearMaxValue	GSV-6 / GSV-8
Extremwertspeicher zurücksetzen, d.h. Maximal- und Minimalwert(e)			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Zu setzender Kanal GSV-6: [0-6] / [0-7] GSV-8: [0-8] / [0-10] Mit Kanal 0 werden ALLE Kanäle zurückgesetzt.
0 Byte	0 Rückgaben		

5 Der Befehl ist erst ab Firmware-Version 3.33 vorhanden.

GetLastProtocolError (0x42)

0x42	R	GetLastError	GSV-6 / GSV-8
GSV-6: Letzten Protokollfehler auslesen. Bei einem (synchronen) Protokollfehler handelt es sich um den in einem Antwort-Frame versendeten Fehler-Code. Bei einem asynchronen Protokollfehler handelt es sich um den Fehler-Code, der bei der autonomen Messwert-Frame Versendung aufgetreten ist. GSV-8: Bei einem asynchronen Protokollfehler handelt es sich entweder um einen Fehler, der beim Empfang des Anfrageframes aufgetreten ist und der das Ignorieren dieses Anfrageframes zur Folge hatte, da fundamentale Protokollanforderungen verletzt wurden. Ein so stark fehlerhafter Anfrageframe wird also (ausnahmsweise) nicht bestätigt. Als Fehlercode tritt dann ERR_FRAME_ERROR auf. Oder es handelt sich um einen Messwertverlust im Sendepuffer des USB-Sendequeues, wodurch ältere, noch nicht gesendete Messwertframes durch neuere überschrieben wurden. Dies kann auftreten, wenn bei hoher Datenrate (>= 16000/s) die USB-Kommunikation zu stark ausgelastet ist, z.B.: durch häufige Befehlsanfragen; oder auch bei Verwendung mehrerer USB-Geräte an demselben USB-Hub, an dem der GSV-8 angeschlossen ist. In diesem Fall tritt als Fehlercode ERR_RET_TXBUF auf.			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	0: (Synchroner) Protokollfehler 1: Asynchronen Protokollfehler Alle anderen Werte sind reserviert
4 Bytes	1 Rückgaben		
0	uint32_t	Fehler-Code	siehe Tabelle S.17

GetLastValueError (0x43)

0x43	R	GetLastValueError	GSV-6
Letzten Messwertfehler auslesen. Siehe Tabelle oben, S. 19			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	0: Fehlerbits Andere Werte ohne Funktion
6 Bytes	1 Rückgaben		
0	uint8_t[6]	Fehler	Index 0: Fehler [0] Fehlerbits (Maximalwertüberschreitung 6-Achsen) Fehler [1] Fehlerbits (Eingangswertsättigung) Fehler [2..5] =0 s.S.21

0x43	R	GetLastValueError	GSV-8
Letzte(n) Messwertfehler auslesen. Siehe Tabelle oben, S. 19			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	0: Fehlerzähler 1-40: Fehler-Struktur

0x43	R	GetLastValueError	GSV-8
6 Bytes	1 Rückgaben		
0	uint8_t [6] ErrorStruct	Fehler	Index 0: Anzahl der seit dem letzten Einschalten aufgetretenen Fehler und Anzahl der nichtflüchtig gespeicherten Fehler auslesen. Antwortparameter: Von den 6 gesendeten Bytes sind nur die beiden zuerst gesendeten relevant, d.h. die beiden "höherwertigen" bei pseudo-numerischer Interpretation; die letzten bzw. unteren 4 Bytes sind konstant 0. Das zuallererst gesendete ist die Anzahl der seit dem letzten Einschalten aufgetretenen Fehler. Das darauffolgende ist die Anzahl der nichtflüchtig im EEPROM gespeicherten Fehler. Index 1: ErrorStruct seit Power On: Fehler, der zuletzt seit dem letzten Einschalten aufgetreten ist und noch nicht nichtflüchtig gespeichert wurde. Indizes 2 bis 83: ErrorStruct des nichtflüchtig im EEPROM gespeicherten Fehlers. Höhere Indizes gehören zu neueren Fehlern, niedrigere zu älteren. Definition des ErrorStruct in C-Notation: {unsigned short ErrFlags:16; unsigned long ErrTime:29; unsigned ErrType:3; } ErrorStruct;

EraseErrorMemory (0x44)

0x44	W / A	EraseErrorMemory	GSV-8
Löschen des nichtflüchtigen Fehlerspeichers. Benutzer-ID erforderlich.			
0 Byte	0 Parameter		
0 Byte	0 Rückgaben		

GetSensorPlugged (0x45)

0x45	R	GetSensorPlugged	GSV-8
Ermitteln, ob am Analogeingang ein Sensor angeschlossen ist (nicht nutzbar bei SingleEnded Input).			
0 Byte	0 Parameter		
4 Bytes	1 Rückgabe		
0	uint32_t	Flagwert	Bit 0: Kanal 1 … Bit 7: Kanal 8 Bit=1: Brückensensor angeschlossen, Bit=0: Kein Brückensensor angeschlossen Bits<15:8>: TEDS-Baustein angeschlossen[6] Kanal 8:1. Bits<23:16>: TEDS schreibbar Kanal 8:1. Bits<31:24>: Reserviert f. weitere Sensoreigenschaften

ReadFTSensorCal (0x47)

0x47	R	ReadFTSensorCal	GSV-6 / GSV-8
Ein Kalibrierwert des **F**orce/**T**orque-Sechsachsensensors wie z.B. Kalibriermatrixelement, Seriennummer, Normierungsfaktor der Kalibriermatrix, Sensormaximalwert und Eingangsempfindlichkeit wird zurückgegeben. **Hinweise**: 1. Es werden mehrere Sechsachsenkalibrierarrays unterstützt (beim GSV-8 bis zu 5). Um Werte(e) aus einem bestimmten Array auszulesen, sollte dieses vorher mit PrepReadFTsensor ausgewählt worden sein. 2. Die 2-dimensionale Kalibriermatrix wird zeilenweise im Array hinterlegt (a11, a12, a13, … a21, …), so dass index={0..5} den Zeilenvektor a11..a15 adressiert, index={6..11} den Zeilenvektor a21..a25 usw.			
2 Bytes	2 Parameter		
0	uint8_t (1)	type	0: Seriennummer (index muss =0 sein) 1: Normierungsfaktor (index muss =0 sein) 2: Matrixwerte (index: [0-35]) 3: Geom. Offsets x,y,z (index: [0-2]) 4: Maximalwerte (index: [0-5]) 5: Eingangsempfindlichkeit (index muss =0 sein) 6: Grundverstimmungswerte bei Nulllast (GSV-8) 7: Sensortyp (GSV-8, FWver >= 1.39): Typ=0: 6-Achsen mit Standard Matrix, nur 1. Ordnung Typ=1: 6-Achsen mit Matrix 1. und 2. Ordnung Typ=2: 3/4-Achsensensor (reserviert) 8: Werte der Matrix B 2. Ordnung (index: [0-35], GSV-8, FWver >= 1.39) 9: Indizes der ersten Eingangsfaktoren (GSV-8, FWver >= 1.39) 10: Indizes der zweiten Eingangsfaktoren (GSV-8, FWver >= 1.39) 11: Enum der Einheit, siehe Cmd. 0x0F (GSV-8, FWver >= 1.54), Index [0-5] S.89 12: Kalibrierdatum: index 0..2: Jahr, Monat, Tag (GSV-8, FWver >= 1.54)
1	uint8_t (1)	index	Index des durch *type* adressierten Arrays
4 Bytes	1 Rückgaben		
0	uint32_t (4) / float (4)	Seriennummer / Einträge	Bis auf die Seriennummer sind alle Einträge float-Werte.

6 Unabhängig davon, ob dieser gültige Daten enthält. Letzteres kann mit Befehl Get TEDS active ermittelt werden (sofern in der Mode-variablen<15:8> das entsprechnede Bit gesetzt ist.

WriteFTSensorCal (0x48)

0x48	W (RAM)	WriteFTSensorCal	GSV-6 / GSV-8
Einen Kalibrierwert des **F**orce/**T**orque-Sechsachsensensors wie z.B. Kalibriermatrixelement, Seriennummer, Normierungsfaktor der Kalibriermatrix, Sensormaximalwert und Eingangsempfindlichkeit schreiben. **Hinweise**: 1. Die 2-dimensionale Kalibriermatrix wird zeilenweise im Array hinterlegt (a11, a12, a13, … a21, …), so dass index={0..5} den Zeilenvektor a11..a15 adressiert, index={6..11} den Zeilenvektor a21..a25 usw. 2. Zum persistenten Speichern der übertragenen Werte muss die Funktion *StoreSensorCal* (0x7F) aufgerufen werden, nachdem alle in sich konsistenten Werte geschrieben wurden. Die Werte für Matrix B sind nur bei Sensortyp=1 relevant. 3. GSV-8: Voraussetzung: Benutzer-ID erforderlich			
6 Bytes	3 Parameter		
0	uint8_t (1)	type	0: Seriennummer (index muss =0 sein) 1: Normierungsfaktor (index muss =0 sein) 2: Matrix A Werte (index: [0-35]) 1. Ord. 3: Geom. Offsets x,y,z (index: [0-2]) 4: Maximalwerte (index: [0-5]) 5: Eingangsempfindlichkeit (index muss =0 sein) 6: Grundverstimmungswerte bei Nulllast (index: [0-5], nur GSV-8) 7: Sensortyp (GSV-8, ab FW-ver 1.39) 8: Werte der Matrix B (index: [0-35], GSV-8, ab FW-ver 1.39) 9: Indizes der ersten Eingangsfaktoren (GSV-8, ab FW-ver 1.39) 10: Indizes der zweiten Eingangsfaktoren (GSV-8, ab FW-ver 1.39) 11: Enum der Einheit, siehe Cmd. 0x0F (GSV-8, FWver >= 1.54) S.89 12: Kalibrierdatum: index 0..2: Jahr, Monat, Tag (GSV-8, FWver >= 1.54)
1	uint8_t (1)	index	Index des durch *type* adressierten Arrays
2	uint32_t (4) / float (4)	Seriennummer / Einträge	Bis auf die Seriennummer sind alle Einträge float-Werte.
0 Byte	0 Rückgaben		

GetTXmapping (0x49)

0x49	R	GetTXMapping	GSV-8 /GSV-6
Zuordnung und physische Bedeutung der Werte im Messwertframe auslesen. Bemerkung: GSV-8: Die Indizes 1..[Anzahl gemappter Objekte] sind erst ab Firmware-version 1.45 implementiert. An Index 0 wird mit älteren Versionen konstant 0 zurückgegeben. GSV-6: Implementiert wie beschrieben ab FW-Ver. 3.11			
1 Byte	1 Parameter		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x49	R	GetTXMapping	GSV-8 /GSV-6
0	uint8_t (1)	Index	Nummer des Mappingeintrags, d.h. der Position des Objektes im Messwertframe [1..<Anzahl gemappter Objekte>] oder =0: Anzahl der Eingangskanäle im Messdatenframe bestimmen.
2 Bytes	1 Rückgaben		
0	uint16_t (2)	Anzahl / Mapping-eintrag	Wenn =0: Kein Mapping definiert. >0: Wenn Index=0: Anzahl gemappter Objekte. Sonst: Mappingeintrag wie folgt: Bits<15:8>, d.h. erstes Datenbyte der Antwort: Physikalischer Typ. z.Zt. vordefiniert (reserviert): 0x00: Kein physikalischer Typ definiert 0x01: Kraft in X Richtung (hauptsächlich mit 6-Achsensensoren) 0x02: Kraft in Y Richtung (hauptsächlich mit 6-Achsensensoren) 0x03: Kraft in Z Richtung (hauptsächlich mit 6-Achsensensoren) 0x04: Drehmoment in X Richtung (haupts. mit 6-Achsensen.) 0x05: Drehmoment in Y Richtung (haupts. mit 6-Achsensen.) 0x06: Drehmoment in Z Richtung (haupts. mit 6-Achsensen.) 0x10: Rohwert bei 6-Achsensor (vor Berechnung) 0x20: Temperatur 0x26: Digitaleingänge (als ganze Zahl (Bits) zu interpretieren) 0x28: Quadratur-Enkoder Pulse 0x38: Quadratur-Enkoder Geschwindigleit Bits<7:4>: Aktueller Wert / Maximal- / Minimalwert: NORMAL=0x00: Das Objekt ist ein aktueller Wert MAX_VAL=0x01: Das Objekt ist ein Maximalwert MIN_VAL=0x02: Das Objekt ist ein Minimalwert Bits<3:0>: Kanalnummer des Eingangs: 1..8: Analoger Sensoreingang (GSV-8) 1..6: Analoger Sensoreingang (GSV-6) 7: Quadraturenkoder-Pulseingang (GSV-6 BT) 9: Quadraturenkoder-Pulseingang Nr. 1 (GSV-8) 10: Quadraturenkoder-Pulseingang Nr. 2 (GSV-8) 11: Digitaleingänge (GSV-8, zulünftige Funktion)

SetTXmapping (0x4A)

0x4A	W	SetTXMapping	GSV-8 / GSV-6
Messwertframe-Zuordnung setzen GSV-6: Implementiert mit Index=0 ab FW-Ver. 3.11 GSV-8: Implementiert mit Index=0 ab FW-Ver. 1.45			
3 Bytes	2 Parameter		
0	uint8_t (1)	Index	Index =0: Mit dem Wert in *Anzahl / Mappingeintrag* wird die Anzahl gemappter Eingangskanäle im Messdatenframe festgelegt. Index >0 [1-16]: Mappingeintrag setzen (reserviert)

0x4A	W	SetTXMapping	GSV-8 / GSV-6
1	uint16_t (2)	Anzahl / Mappingeintrag	Anzahl gemappter Objekte im Messdatenframe bzw. Mappingeintrag, wie in GetTXmapping definiert. (reserviert)
0 Byte	0 Rückgaben		

Wertebereiche an Index 0:

GSV-6: Nur Werte 1,2,3,6 (kein Counter vorhanden)[7]
GSV-6BT: Mit Counter: Nur Werte 2,3,4,7. Ohne Counter: s. GSV-6
GSV-8 (ab FW.ver 1.45): Ohne Counter: USB: 2..8, UART: 1..8
GSV-8 (ab FW.ver 1.45): Mit 1 Counterkanal: 2..9
GSV-8 (ab FW.ver 1.45): Mit 2 Counterkanälen: 3..10

GetDigiFiltType (0x4B)

0x4B	R	GetDigiFiltType	GSV-6 / GSV-8
Typ des Digitalfilters ermitteln			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer GSV-6: [1-6] GSV-8: [1-8]
1	uint8_t (1)	subTyp	GSV-6: Ungenutzt, =0 setzen GSV-8: 0x00: IIR-Filter 0x80 FIR-Filter (**nur GSV-8**)
1 Byte	1 Rückgaben		
0	uint8_t (1)	Filtertyp-Enum	<7>: FIR-Filter-Indikator: =1 FIR Filter, =0 IIR <6:4> Filtertyp: -0: Tiefpass -1: Hochpass -2: Bandpass -3: Bandstop -4: Kammfilter (GSV-8) <3:0> Filterlänge i.a. für IIR konstant =4 und für FIR einer der Werte: 4,6,8,10,12,14

SetDigiFiltType (0x4C)

0x4C	W (RAM)	SetDigiFiltType	GSV-6 / GSV-8
Typ des Digitalfilters setzen ***Achtung***: 1. Zum persistenten Speichern der übertragenen Filter-Werte muss die Funktion StoreDigiFilt (0x7E) aufgerufen werden! Dies sollte nach der Übertragung aller Filterwerte (Koeffizienten, Typ, Grenzen) für alle zu setzenden Filter einmalig erfolgen. 2. Das FIR Filter ist nur mit GSV-8 verwendbar			
2 Bytes	2 Parameter		

7 Ab Firmware-Version 3.33 kann jede Kanalanzahl von 1-6 gesetzt werden; mit Counter 2-7.

0x4C	W (RAM)	SetDigiFiltType	GSV-6 / GSV-8
0	uint8_t (1)	Kanal	Kanalnummer GSV-6: [0-6] GSV-8: [0-8] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.
1	uint8_t (1)	Filtertyp-Enum	<7>: FIR-Filter-Indikator: =1 FIR Filter, =0 IIR <6:4> Filtertyp: -0: Tiefpass -1: Hochpass -2: Bandpass -3: Bandstop -4: Kammfilter (GSV-8) <3:0> Filterlänge i.a. für IIR konstant =4 und für FIR einer der Werte: 4,6,8,10,12,14
0 Byte	0 Rückgaben		

ReadDigiFiltCutOff (0x4D)

0x4D	R	ReadDigiFiltCutOff	GSV-6 / GSV-8
Digitalfilter-Grenzfrequenz auslesen. Hiermit werden die informativen Werte, die mittels WriteDigiFiltCutOff (0x4E) gesetzt wurden, zurückgegeben.			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer GSV-6: [1-6] GSV-8: [1-8]
1	uint8_t (1)	Index	0: Fgu (untere Grenzfrequenz) 1: Fgo (obere Grenzfrequenz)
4 Bytes	1 Rückgaben		
0	float (4)	Grenzfrequenz	

WriteDigiFiltCutOff (0x4E)

0x4E	W (RAM)	WriteDigiFiltCutOff	GSV-6 / GSV-8
Einen der beiden Digitalfilter-Grenzfrequenzen (-3dB) setzen. Hierbei handelt es sich um einen rein informativen Wert, der keine Auswirkung auf das Digitalfilter besitzt. Bei Tief- und Hochpass gilt nur Fgu. ***Achtung***: Zum persistenten Speichern der übertragenen Filter-Werte die Funktion StoreDigiFilt (0x7E) aufgerufen werden! Dies sollte nach der Übertragung aller Filterwerte (Koeffizienten, Type, Grenzen) für alle zu setzenden Digitalfilter einmalig erfolgen.			
6 Bytes	3 Parameter		

0x4E	W (RAM)	WriteDigiFiltCutOff	GSV-6 / GSV-8
0	uint8_t (1)	Kanal	Kanalnummer GSV-6: [0-6] GSV-8: [0-8] Mit Kanal 0 werden ALLE Kanäle mit dem identischen Wert beschrieben.
1	uint8_t (1)	Index	0: Fgu (untere Grenzfrequenz) 1: Fgo (obere Grenzfrequenz)
2	float (4)	Grenzfrequenz	
0 Byte	0 Rückgaben		

ReadDigiFiltCoeff (0x4F)

0x4F	R	ReadDigiFiltCoeff	GSV-6
Digitalfilter-Koeffizienten auslesen			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer [1-6]
1	uint8_t (1)	Index	Nur mit IIR: 0x00...0x04: Koeffizienten A (Eingang) 0x10...0x13: Koeffizienten B (Ausgang, Rückkopplung)
4 Bytes	1 Rückgaben		
0	float (4)	Koeffizient	

0x4F	R	ReadDigiFiltCoeff	GSV-8
Filter-Koeffizienten auslesen			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer [1-8]
1	uint8_t (1)	Index	Bei IIR: -0x00...0x04: Koeffizienten A (Eingang) -0x10...0x13: Koeffizienten B (Ausgang, Rückkopplung) Bei FIR: -0x80...0x87: Koeffizienten (nur bis einschließlich Mitte der symmetrischen Koeffizientenfolge)
4 Bytes	1 Rückgaben		
0	S7.24 (4)	Koeffizient	

WriteDigiFiltCoeff (0x50)

0x50	W (RAM)	WriteDigiFiltCoeff	GSV-6
Filter- Koeffizienten setzen ***Achtung***: Zum persistenten Speichern der übertragenen Filter-Werte muss die Funktion *StoreDigiFilt* (0x7E) aufgerufen werden! Dies sollte nach der Übertragung aller Filterwerte (Koeffizienten, Type, Grenzen) für alle zu setzenden Filter einmalig erfolgen.			
6 Bytes	3 Parameter		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x50	W (RAM)	WriteDigiFiltCoeff	GSV-6
0	uint8_t (1)	Kanal	Kanalnummer [0-6] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.
1	uint8_t (1)	Index	Nur mit IIR: - 0x00...0x04: Koeffizienten A (Eingang) - 0x10...0x13: Koeffizienten B (Ausgang, Rückkopplung)
2	float (4)	Koeffizient	
0 Byte	0 Rückgaben		

0x50	W	WriteDigiFiltCoeff	GSV-8
Filter- Koeffizienten setzen[8] ***Achtung***: Zum persistenten Speichern der übertragenen Filter-Werte muss die Funktion StoreDigiFilt (0x7E) aufgerufen werden! Dies sollte nach der Übertragung aller Filterwerte (Koeffizienten, Type, Grenzen) für alle zu setzenden Filter einmalig erfolgen.			
6 Bytes	3 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer [0-8] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.
1	uint8_t (1)	Index	Bei IIR: -0x00...0x04: Koeffizienten A (Eingang) -0x10...0x13: Koeffizienten B (Ausgang, Rückkopplung) Bei FIR: -0x80...0x87: Koeffizienten (nur bis einschließlich Mitte der symmetrischen Koeffizientenfolge)
2	S7.24 (4)	Koeffizient	
0 Byte	0 Rückgaben		

GetDfiltOnOff (0x51)

0x51	R	GetDfiltOnOff	GSV-6
Abfrage der aktiven Filter (Bit Maske) Gesetzte Bits entsprechen einem aktiven Filter.			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		
0	uint32_t (4)	Kanal Flags	GSV-6: <5:0> Filter auf dem entsprechenden Kanal aktiv <31:6> reserviert

8 Das Gerät kann die Koeffizienten nicht selbst ermitteln und dies ist auch nicht Gegenstand dieses Dokuments. Die Windows DLL MEGSV86xx.dll bietet eine Funktion zum Berechnen und Setzen der Filterkoeffizienten.

0x51	R	GetDfiltOnOff	GSV-8
Abfrage der aktiven Filter (Bit Maske) Gesetzte Bits entsprechen einem aktiven Filter.			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		
0	uint32_t (4)	Kanal Flags	GSV-8: <7:0> IIR-Filter auf dem entsprechenden Kanal (Bit0: K1) aktiv <15:8> FIR-Filter auf dem entsprechenden Kanal (Bit8: K1) aktiv <31:16> reserviert

SetDfiltOnOff (0x52)

0x52	W	SetDfiltOnOff	GSV-6
Setzen der aktiven Filter (Bit Maske) Gesetzte Bits entsprechen einem aktiven Filter.			
4 Bytes	1 Parameter		
0	uint32_t (4)	Kanal Flags	GSV-6: - <5:0> Filter auf dem entsprechenden Kanal aktiv - <31:6> reserviert
0 Byte	0 Rückgaben		

0x52	W	SetDfiltOnOff	GSV-8
Setzen der aktiven Filter (Bit Maske) Gesetzte Bits entsprechen einem aktiven Filter.			
4 Bytes	1 Parameter		
0	uint32_t (4)	Kanal Flags	GSV-8: - <7:0> IIR-Filter auf dem entsprechenden Kanal (Bit0: K1) aktivieren - <15:8> FIR-Filter auf dem entsprechenden Kanal (Bit8: K1) aktivieren - <31:16> reserviert
0 Byte	0 Rückgaben		

ReadMaxMinVal (0x53)

0x53	R (Seriell)	ReadMaxMinVal	GSV-6
Maximal- und Minimalwert eines Kanals auslesen			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer [1-6] / [1-7]
9 Bytes	3 Rückgaben		
0	uint8_t (1)	Indikator	=0
1	float (4)	Maximalwert	

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x53	R (Seriell)	ReadMaxMinVal	GSV-6
5	float (4)	Minimalwert	

0x53	R (CAN)	ReadMaxMinVal	GSV-6
Maximal- und Minimalwert eines Kanals auslesen **Achtung**: Es werden zwei Antworten versendet! Erst der Maximalwert, dann der Minimalwert.			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer: [1-6] / [1-7]
5 Bytes	2 Rückgaben		
0	uint8_t (1)	Indikator	==1: Maximalwert ==2: Minimalwert
1	float (4)	Extremwert	

0x53	R (Seriell)	ReadMaxMinVal	GSV-8
Maximal- und Minimalwert eines Kanals auslesen			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer [1-8] / [1-10]
8 Bytes	2 Rückgaben		
0	float (4)	Maximalwert	
1	float (4)	Minimalwert	

GetFTSensorCalArrNo (0x54)

0x54	R	GetFTSensorCalArrNo	GSV-6 / GSV-8
Informationen über die Kalibrierstrukte der Mehrachsensensoren auslesen			
0 Byte	0 Parameter		
3 Bytes	3 Rückgaben		
0	uint8_t (1)	Maximum	Maximale Anzahl der Matrizen, die im Gerät zur Verfügung stehen.
1	uint8_t (1)	NumAktiv	Anzahl der gespeicherten Strukte N
2	uint8_t (1)	Strukt-Nummer	Nummer des aktiven Struktes [0..N-1]

SetFTSensorCalArrNo (0x55)

0x55	W	SetFTSensorCalArrNo	GSV-6 (Reset) / GSV-8
Setzen des aktiven Kalibrierstruktes für Mehrachsensensor			
1 Byte	1 Parameter		
0	uint8_t (1)	Strukt-Nummer	[0..N-1]
0 Byte	0 Rückgaben		

ReadDeviceHours (0x56)

0x56	R	ReadDeviceHours	GSV-8 / GSV-6BT
Einer der beiden Betriebsstundenzähler kann hiermit ausgelesen werden. Der „absolute“ Betriebsstundenzähler ist monoton und kann nicht zurückgesetzt werden. Der „einstellbare“ Betriebsstundenzähler ist monoton, solange er nicht über das Kommando WriteDeviceHours (0x57) auf einen neuen Wert gesetzt wird. GSV-6: implementiert ab FW-Ver. 3.11, sofern Hardware-Voraussetzung erfüllt (sonst: Fehlerrückgabe)			
1 Byte	1 Parameter		
0	uint8_t (1)	Zähler-Index	0: absolut 1: setzbar Andere Werte sind reserviert.
4 Bytes	1 Rückgaben		
0	float (4)	Zähler	Betriebsstunden des Geräts.

WriteDeviceHours (0x57)

0x57	W	WriteDeviceHours	GSV-8
Den veränderlichen Betriebsstundenzähler setzen Voraussetzung: Benutzer-ID erforderlich GSV-6: implementiert ab FW-Ver. 3.11, sofern Hardware-Voraussetzung erfüllt (sonst: Fehlerrückgabe)			
1 Byte	1 Parameter		
0	float (4)	Wert	Wert für den veränderlichen Betriebsstundenzähler in Stunden.
0 Byte	0 Rückgaben		

SetPassword (0x58)

0x58	W	SetPassword	GSV-8
Das Benutzer-Passwort verändern Hinweis: Voraussetzung: Vorangehendes Setzen Benutzer-ID erforderlich			
4 Bytes	1 Parameter		
0	Char[4]	Wert	Wert des neuen Passwortes. Muss 4 ASCII-Zeichen lang sein.
0 Byte	0 Rückgaben		

GetDIOdirection (0x59)

0x59	R	GetDIOdirection	GSV-8
Die Datenrichtung (1=Input, 0=Output) der Gruppe der digitalen I/O Anschlüsse ermitteln. I/Os 1-4 gehören zur Gruppe 1, 5-8 zu 2, 9-12 zu 3 und 13-16 zur Gruppe 4.			
1 Byte	1 Parameter		
0	uint8_t	Gruppen-No	Nummer der Gruppe [1-4]. =0: Richtungen aller 4 Gruppen bestimmen
1 Byte	1 Rückgaben		=0: Ausgang, =1: Eingang. Bei Gruppen-No=0: In Bit 0: Gruppenrichtung No 1… Bit 3: Richtung Gruppe 4

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

SetDIOdirection (0x5A)

0x5A	W	SetDIOdirection	GSV-8
Die Datenrichtung (1=Input, 0=Output) der Gruppe der digitalen I/O Anschlüsse setzen. I/Os 1-4 gehören zur Gruppe 1, 5-8 zu 2, 9-12 zu 3 und 13-16 zur Gruppe 4.			
2 Bytes	2 Parameter		
0	uint8_t	Gruppen-No	Nummer der Gruppe [1-4]. =0: Richtungen aller 4 Gruppen setzen
1	uint8_t	Datenrichtung	=0: Ausgang, =1: Eingang. Bei Gruppen-No=0: In Bit 0: Gruppenrichtung No 1… Bit 3: Richtung Gruppe 4
0 Byte	0 Rückgaben		

Die digitalen I/O-Leitungen sind in Gruppen zu je 4 Leitungen organisiert. I/O-Leitungen 1-4 gehören zu Gruppe 1, 5-8 zu Gruppe 2, 9-12 zu Gruppe 3, 13-16 zu Gruppe 4. Innerhalb einer Gruppe haben alle 4 Leitungen die gleiche Richtung. Auch dies schränkt die Konfigurationsvielfalt der I/O-Typen ein.

GetDIOtype (0x5B)

0x5B	R	GetDIOtype	GSV-8 / GSV-6
Die Funktion des digitalen I/O Anschlusses ermitteln. GSV-6: ab FWver 3.27			
1 Byte	1 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses GSV-8: [1-16]. GSV-6: [1-5].
4 Bytes	2 Rückgaben		
0	Uint_24	DIOtyp	Typ des DIO Anschlusses (s. Tabelle im Anhang B, S. 90)
1	uint8_t	Zugeordneter Kanal	Zugeordneter Eingangskanal GSV-8: [1-8] / GSV-6: [1-6] (nur relevant bei Schwellwertschalter und Einzel-Tara)

SetDIOtype (0x5C)

0x5C	W	SetDIOtype	GSV-8
Die Funktion des digitalen I/O Anschlusses setzen. Bemerkung: Evenmtuell erst die Datenrichtung mit *SetDIOdirection* gesetzt werden. Bei den im Anhang B tabellierten Typen ist die Richtung vermerkt.			
5 Bytes	3 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses [1-16]. Wert=0 bedeutet: Alle auf denselben Typ setzen
1	Uint_24	DIOtyp	Typ des DIO Anschlusses (s. Tabelle im Anhang B)
2	uint8_t	Zugeordneter Kanal	Zugeordneter Eingangskanal [1-8] (nur relevant bei Schwellwertschalter und Einzel-Tara)
0 Byte	0 Rückgaben		

0x5C	W	SetDIOtype	GSV-6
Die Funktion des digitalen I/O Anschlusses setzen. Ab FWver 3.27 vorhanden. Bemerkung: Ein- und Ausgangsfunktionen sind den Hardwaranschlüssen (Pins) wie folgt zugeordnet: DIO-Nr. 1-3: SW1,SW2,SW3[9]: Nur Ausgangsfunktionen DIO-Nr. 4: TRIGGER: Nur Eingangsfunktionen DIO-Nr. 5: TARA: Nur Eingangsfunktionen			

0x5C	W	SetDIOtype	GSV-6
5 Bytes	3 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses [1-5].
1	Uint_24	DIOtyp	Typ des DIO Anschlusses (s. Tabelle im Anhang B)
2	uint8_t	Zugeordneter Kanal	Zugeordneter Eingangskanal [1-6] (nur relevant bei Schwellwertschalter und Einzel-Tara)
0 Byte	0 Rückgaben		

GetDIOlevel (0x5D)

0x5D	R	GetDIOlevel	GSV-8 / GSV-6
Den Pegel des digitalen I/O Anschlusses ermitteln. GSV-6: ab FWver 3.29			
1 Byte	1 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses [1-16] / GSV-6: [1-5]. Wenn =0: Pegel aller DIO Anschlüsse bestimmen
2 Bytes	1 Rückgaben		
0	uint16_t	DIOlevel	Pegel des DIO Anschlusses: 0: Low, 1: High. Wenn DIOno=0: GSV-8: Bit 0: Anschluss No. 1 bzw. 1.1. ... Bit15: GPIO No 16 bzw. 4.4. GSV-6: Bit 0: No 1 (SW1)... Bit 3: No 4 (TRIG), Bit 4: DIO No 5 (TARA)

SetDIOlevel (0x5E)

0x5E	W	SetDIOlevel	GSV-8
Den Pegel des digitalen GPIO Anschlusses setzen (wenn Datenrichtung = Output und Typ= GP-out)			
3 Bytes	2 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses 1..16. =0: Pegel aller DIO Anschlüsse setzen
1	uint16_t	DIOlevel	Pegel des DIO Anschlusses: 0: Low, 1: High. Wenn DIOno=0: Bit 0: DIO No. 1.. Bit15: DIO No 16
0 Bytes	0 Rückgaben		

0x5E	W	SetDIOlevel	GSV-6
Den Pegel des digitalen Output Anschlusses setzen. Ab FWver 3.29 vorhanden. Die Anzahl vorhandenen Ausgänge kann mit Cmd. 0x2A ermittelt werden. Beim GSV-6BT sind es 2, sonst in der Regel 3.			
3 Bytes	2 Parameter		
0	uint8_t	DIO No	Nummer des DO Anschlusses [1-3/2]. =0: Pegel aller DO Anschlüsse setzen
1	uint16_t	DIOlevel	Pegel des DO Anschlusses: 0: Low, 1: High. Wenn DIO No=0: Bit 0: DO No. 1 (SW1).. Bit2: DO No 3 (SW3)
0 Bytes	0 Rückgaben		

9 SW3 ist bei einigen Modellen, z.B. GSV-6BT nicht vorhanden. Das kann mit Cmd. 0x2A ermittelt werden.

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

ReadDIOthreshold (0x5F)

0x5F	R	ReadDIOthreshold	GSV-8 / GSV-6
Den Schwellwert des digitalen Schwellwertausgangs ermitteln. GSV-6: ab FWver 3.29			
2 Bytes	2 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses. GSV-8: [1-16]. GSV-6: [1-3]
1	uint8_t	Upper / lower	=0: Unterer Schwellwert, =1: Oberer Schwellwert
4 Bytes	1 Rückgabe		
0	Float	DIOthreshold	Schwellwert des digitalen Schwellwertausgangs

WriteDIOthreshold (0x60)

0x60	W	WriteDIOthreshold	GSV-8 / GSV-6
Den Schwellwert des digitalen Schwellwertausgangs schreiben. GSV-6: ab FWver 3.29			
6 Bytes	3 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses. GSV-8: [1-16]. GSV-6: [1-3] =0: Alle auf denselben Wert setzen.
1	uint8_t	Upper / lower	=0: Unterer Schwellwert, =1: Oberer Schwellwert
2	Float	DIOthreshold	Schwellwert des digitalen Schwellwertausgangs
0 Byte	0 Rückgaben		

GetDIOinitialLevel (0x61)

0x61	R	GetDIOinitLevel	GSV-8 / GSV-6
Den Default- bzw Initialisierungspegel des Digitalausgangs bestimmen. GSV-6: ab FWver 3.29			
1 Byte	1 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses GSV-8: [1-16]. GSV-6: [1-3] =0: Initpegel aller DIO Anschlüsse bestimmen
2 Bytes	1 Rückgabe		
0	uint16_t	DIO Init-Level	Initpegel des DIO Anschlusses: 0: Low, 1: High. Wenn DIOno=0: GSV-8: Bit 0: Anschluss No. 1 bzw. 1.1. ... Bit15: GPIO No 16 bzw. 4.4. GSV-6: Bit 0: DIO Nr. 1 (SW1).. Bit 2: Nr. 3 (SW3)

SetDIOinitialLevel (0x62)

0x62	W	SetDIOinitLevel	GSV-8 / GSV-6
Den Default- bzw Initialisierungspegel des Digitalausgangs schreiben. GSV-6: ab FWver 3.29			
3 Bytes	2 Parameter		
0	uint8_t	DIO No	Nummer des DIO Anschlusses. GSV-8: [1-16]. GSV-6: [1-3] =0: Initpegel aller DIO Anschlüsse setzen
1	uint16_t	DIO Init-Level	Initpegel des DIO Anschlusses: 0: Low, 1: High. Wenn DIOno=0: GSV-8: Bit 0: Anschluss No. 1 bzw. 1.1. ... Bit15: GPIO No 16 bzw. 4.4. GSV-6: Bit 0: DIO Nr. 1 (SW1).. Bit 2 Nr.3 (SW3)
0 Byte	0 Rückgaben		

Der DIO-Default-Pegel wird bei digitalen Ausgängen dann gesetzt, wenn (noch) keine der

definierten Bedingungen für High- oder Low-Zustand vorliegen; zB bei General-Purpose-Output nach dem Einschalten.

ReadDataRateRange (0x63)

0x63	R	ReadDataRateRange	GSV-8
Die maximal oder minimal einstellbare Datenrate bestimmen.			
1 Byte	1 Parameter		
0	uint8_t	Index	=0: Maximale Datenrate bestimmen =1: Minimale Datenrate bestimmen
4 Bytes	1 Rückgabe		
0	Float	Max/Min Drate	Maximale / Minimale Datenrate

ReadTEDSdataEntry (0x64)

0x64	R	ReadTEDSdataEntry	GSV-8
TEDS-Einträge lesen			
4 Bytes	4 Parameter		
0	uint8_t	Kanalnummer	1..8
1	uint8_t	TEDS Template-ID.	0= Basic-TEDS, 33, 35, 25
2	uint8_t	Property-ID	s. Anhang D
3	uint8_t	Array-Index	[Reserviert, z.Zt. =0]. Verwendet, wenn a) gleiche Prop-ID mehrfach vorh., b) wenn gleiches Template mehrfach vorh.
6 Bytes	3 Rückgaben		
0	uint8_t	Next Index	Nächster Index der Liste
1	uint8_t	ValTyp/Error	Datentyp des Wertes (Bytes 2..5) u. Errors, die nicht per Error-Frame zurückgegeben werden
2	uint32_t oder Float	Data	Daten. Typ=Float, wenn ValTyp_Err =1. Sonst Uint32

Die Property-ID indiziert alle möglichen Einträge, dh 1. Properties (s. IEEE1541.4 Annex D, S. 96), 2. relevante Selectors, 3. Sonderwerte; s. Anhang D. Die Reihenfolge der verketteten Liste entspricht (nach ID 0) der im Template.

Wenn Next ID =0: Letzter Eintrag.

ValTyp/Error:
Datentyp/Bedeutung des Wertes in Bytes 2..5 oder Fehlercode, der nicht im Antwortframe kodiert ist.

Value	Name	Meaning
0	ANSW_IS_UINT	Ganzzahliger unsigned-Wert (uint32_t)
1	ANSW_IS_FLT	Wert im Float-Format, bzw darin umgerechnet (float32)
2	IS_PACKED_CHR5	CHR5-Typ, gepackt (siehe IEEE1541.4)
4	IS_DATE_DAYS	Datum in Tagen seit 1.1.1998
0x80	ENTRY_HAS_ERROR	Bit7=1: Vergleichswert f. Error-Erkennung: Flags>=0x80: Datenwert nicht auswerten
0xFD	ENTRY_INVALID	Eintrag ungueltig, zB NaN bei Float
0xFE	ENTRY_NOT_SET	Eintrag existiert, ist aber als "don't care" geflaggt, dh alle Bits=1
0xFF	ENTRY_NOT_EXIST	Entry-Anforderung bzw Property-ID existiert nicht im Template. Auch bei Index 0; dann nicht als error anzusehen

ReadTEDSrawArray (0x65)

0x65	R	ReadTEDSrawArray	GSV-8 / GSV-6
TEDS-Rohdaten lesen **Hinweis**: Der Befehl ist mit dem GSV-6 erst ab Firmware-Version 3.10 verfügbar			
3 Bytes	2 Parameter		
0	uint8_t	Kanalnummer	1..8. Bei GSV-6: Irrelevant, nur an Kanal 1 vorhanden.
1	uint16_t	Byteadresse	Byteadresse des TEDS-Bereiches im 1-wire EEPROM ohne Checksumme, 4-Byte aligned
4 Bytes	4 Rückgaben		
0	uint8_t	Data @ Address	Databyte 1
1	uint8_t	Data @ Address +1	Databyte 2
2	uint8_t	Data @ Address +2	Databyte 3
3	uint8_t	Data @ Address +3	Databyte 4

Beim GSV-8 existieren seit FW-Ver 1.53 mit Co-Prozessor Version >=2 folgende Sonderfunktionen mit speziellen Adresswerten:

Adresse (Hex)	Funktion	Anwendbare 1-wire ICs
8000	Lese ROM Datenarray, erste 4 Bytes: Databyte 1: Family code, Dann erste 3 Bytes der Ser.No.	alle
8004	Lese ROM Datenarray, letzte 4 Bytes: Letzte 3 Bytes der Ser.No., Checksumme. Voraussetzung: Vorher an 0x8000 gelesen.	alle
FFF0	Lese Temperatur, zwei Bytes. Databyte 1: Lowbyte, 2: High. Ist Temperatur in °C x16.	MAX31820, MAX31826
FFFF	"Binär" Lesen einleiten. Wenn diese Adresse zuerst gegeben wird (Data dann =0), erfolgen alle folgenden Leseaufrufe mit Rohdaten. Hiermit können dann beliebige, auch Nicht-TEDS-konforme EEPROM Daten aus dem 1-wire Speicherbaustein gelesen werden. Andernfalls wird (wie bisher), die im TEDS-Standard definierte Checksumme intern geprüft und aus den Lesedaten herausgenommen.	24B33, MAX31826, DS2430

WriteTEDSbytes (0x66)

0x66	W (RAM)	WriteTEDSbytes	GSV-8
TEDS-Rohdaten schreiben			
6 Bytes	6 Parameter		
0	uint8_t	Kanalnummer	1..8
1	uint8_t	Byteadresse	Virtuelle Byteadresse, die nach jedem Schreibvorgang mit StoreTEDSdata wieder mit 0 beginnt. Muss /4 teilbar und kleiner 64 sein.
2	uint8_t	Data @ Address	Databyte 1
3	uint8_t	Data @ Address +1	Databyte 2
4	uint8_t	Data @ Address +2	Databyte 3
5	uint8_t	Data @ Address +3	Databyte 4
0 Bytes	0 Rückgaben		

Die Daten werden nur in das RAM des Gerätes kopiert, d.h. noch nicht in das 1-wire EEPROM geschrieben. Bitfelder auf Bit 0 an Adresse 0 sind aligned, d.h. höchstwertigste Bytes/Bits ggf. don't care. Ohne Checksumme (wird in Cmd 0x67 berechnet und gesetzt). Zum Übertragen in das 1-wire EEPROm Kommando StoreTEDSdata verwenden.

StoreTEDSdata (0x67)

0x67	**W**	**StoreTEDSdata**	**GSV-8**
TEDS-Rohdaten aus Geräte-RAM in das 1-wire-EEPROM übertragen			
5 Bytes	3 Parameter		
0	uint8_t	Kanalnummer	1..8
1	uint16_t	Bitadresse	Bitadresse im 1-wire EEPROM, ab dem die Daten geschrieben werden, ohne Checksumme
2	int16_t	Bitsize	Größe des zu schreibenden Datenfeldes in Bits. Bitsize >0: TEDS Daten, d.h. Checksumme wird berechnet und ergänzt. Bitsize<0, d.h. Größe= Bitsize *-1: Non-TEDS Rohdaten, d.h. Ohne Checksumme "as it is" (ab FWver. 1.53)
0 Bytes	0 Rückgaben		

Get TEDS active (0x68)

0x68	**R**	**Get TEDS active**	**GSV-8 / GSV-6**
Ermitteln, ob ein gültige Daten aus einem Sensor mit TEDS geladen und verwendet werden **Hinweis**: Der Befehl ist mit dem GSV-6 erst ab Firmware-Version 3.10 verfügbar (nur Bit 0).			
0 Bytes	0 Parameter		
4 Bytes	1 Rückgabe		Bits<31:8>: reserviert Bits<7:0>: Wenn Bit=1: Am Eingangskanal (BitNo+1) ist ein TEDS-Baustein mit gültigen und bekannten Daten angeschlossen, die zur Berechnung des UserScale-Wertes verwendet wurden (d.h. dann ist auch entsprechendes Bit in <15:8> des Mode-Wertes gesetzt).

Read Counter/Freq Mode (0x69)

0x69	**R**	**ReadCountFreqMode**	**GSV-6 BT / GSV-8**
Einstellungen der QEI-Endoder Messung / Frequenzmessung lesen **Hinweis**: Der Befehl ist beim GSV-6 erst ab Firmware-Version 3.11 verfügbar, beim GSV-8 erst ab 1.45 GSV-8: In Bits<7:4> des Index-Parameters steht die Counter-Nummer: 0 für QEI 1, 1 für QEI 2			
1 Byte	1 Parameter		
0	uint8_t	Index	0..2 (GSV-6) / 0x00..0x12 (GSV-8)
4 Bytes	1 Rückgabe		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x69	R	ReadCountFreqMode	GSV-6 BT / GSV-8
0	uint32_t	ModusFlags/GateTime /Wert	**Index 0: Modus-Flags:** Bit 0: =1 Modul vorhanden, =0: Nicht vorhanden. Hinweis: Wenn nicht vorhanden, ist der ganze Wert =0 und die Funktionalität nicht nutzbar Bit 1: =1: Zählermessung aktiviert Bit 2: =1: Frequenz/Drehzahl Messung aktiv. Diese beiden Funktionen sind beim GSV-6 nicht zugleich nutzbar. Beim GSV-8 sind sie nur für QEI 1 zugleich nutzbar. Bits <4:3>: QEI mode: 00 FreeRun, non-QEI mode 01 QEI x1 10 QEI x2 11 QEI x4 Bit 5: =1: Home/Index Eingang verwenden Bit 6: =0: Zählerwert-Sättigung aus =1: Zählerwert-Sättigung an Bit 7: =1: Letzter Zählerwert wird nichtflüchtig gespeichert Bit 8: =1: Periodenmessung für Frequenzmodus Bit 9: =1: Automatische Auswahl Periodenmessung oder Zählung für Frequenzmodus (nur GSV-8) Bit 10: =0: Pullup-Widerstände (10kΩ) **an** =1: Pullup-Widerstände (10kΩ) **aus** (nur GSV-8) Bit 11: =1: Index Eingang invertiert (nur GSV-8) Bit 12: =1: Eingangsfilter verwenden (nur GSV-8) **Index 1: Torzeit** Counter Modus: Anzahl der Datenperioden, bis Zählerwert für Frequenzmessung ausgewertet wird. Periodenmessung: Anzahl der Perioden mit gleichbleibendem Wert, nach deren Ablauf der Frequenzwert =0 gesetzt wird. **Index 2:** Startwert f. Home/Index-Eingang

Write Counter/Freq Mode (0x6A)

0x6A	W	WrCountFreqMode	GSV-6 BT / GSV-8
Einstellungen der QEI-Endoder Messung / Frequenzmessung setzen **Hinweis**: Der Befehl ist mit dem GSV-6 erst ab Firmware-Version 3.11 verfügbar, sofern die Funktion durch die Hardwareausführung unterstützt wird; siehe ReadCountFreqMode, Index 0, Bit 0. Beim GSV-8 erst ab Firmware-Version 1.45 verfügbar. GSV-8: In Bits<7:4> des Index-Parameters steht die Counter-Nummer			
5 Bytes	2 Parameter		
0	uint8_t	Index	0..2

0x6A	W	WrCountFreqMode	GSV-6 BT / GSV-8
1	uint32_t	ModusFlags/GateTime/ Wert	**Index 0: Modus-Flags:** < Siehe ReadCounterFreqMode> **Hinweis**: Bei Änderung in Bits 1 oder 2 ändert sich u.U. die Anzahl der Objekte im Messdatenframe. Diese kann anschliessend mit GetTXmapping (0x49) abgefragt werden. **Index 1: Torzeit** Anzahl der Datenperioden, bis Zählerwert für Frequenzmessung ausgewertet wird **Index 2:** Startwert f. Home/Indexeingang
0 Byte	0 Rückgaben		

Read Clock Time (0x6B)

0x6B	R (seriell)	ReadClockTime	GSV-6BT
Datum und Uhrzeit der RTC oder Alarmintervall des RTC-Alarms lesen. **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardware eine RTC enthält (Andernfalls wird eine Fehlermeldung gesendet).			
1 Byte	1 Parameter		
0	uint8_t	Index	[0-3] 0: Aktuelle Uhrzeit. 1: Alarmintervall 1 (Tag, Stunde, Minute) 2: Alarmintervall 2 (reserviert) 3: UTC offset[10]: Eintrag "Tag": Vorzeichen: 0=positiv, 0xFF= negativ. Stunde, Minute: Verschiebung zu UTC
7 Bytes	7 Rückgaben		
0	uint8_t	Indikator	Stets =0
1	uint8_t	Jahr	Kodiert als Binärwert, der einen Summanden zum Jahr 2000 angibt. In 2017 ist der Wert z.B. =17. Die Alarmzeit (Index=1) hat kein Jahr; der Wert ist dann =0 und zu ignorieren.
2	uint8_t	Monat	Monat von 1 (Januar) bis 12. Nur Index 0
3	uint8_t	Tag	Tag des Monats von 1 bis 31 bzw UTC-of. VZ
4	uint8_t	Stunde	Stunde des Tages von 0 bis 23
5	uint8_t	Minute	Minute von 0 bis 59
6	uint8_t	Sekunde	Sekunde von 0 bis 59. Nur Index 0

0x6B	R (CAN)	ReadClockTime	GSV-6
Datum und Uhrzeit der RTC oder Alarmintervall des RTC-Alarms lesen. ***Achtung***: Es werden auf eine Anfrage **zwei** Antwort-Paket versendet. **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardware eine RTC enthält (Andernfalls wird eine Fehlermeldung gesendet).			
1 Byte	1 Parameter		
0	uint8_t	Index	0..1 0: Aktuelle Uhrzeit. 1: Alarmintervall (Tag, Stunde, Minute)
4 Bytes	4 Rückgaben		
0	uint8_t (1)	Indikator	=1 ‚Erste Hälfte': Jahr/Monat/Tag =2 ‚Zweite Hälfte': Std./Min./Sek.
1	uint8_t	Jahr/Stunde	siehe seriell

10 Vorhanden ab Firmware Version 3.27

0x6B	R (CAN)	ReadClockTime	GSV-6
2	uint8_t	Monat/Minute	siehe seriell
3	uint8_t	Tag/Sekunde	siehe seriell

Write Clock Time (0x6C)

0x6C	W (seriell)	WriteClockTime	GSV-6BT
Datum und Uhrzeit der RTC oder Alarmintervall des RTC-Alarms setzen. **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardware eine RTC enthält (Andernfalls wird eine Fehlermeldung gesendet).			
8 Bytes	8 Parameter		
0	uint8_t	Index	[0-3] 0: Aktuelle Uhrzeit. 1: Alarmintervall 1 (Tag, Stunde, Minute) 2: Alarmintervall 2 (reserviert) 3: UTC offset[11]: Eintrag "Tag": Vorzeichen: 0=positiv, 0xFF= negativ. Stunde, Minute: Verschiebung zu UTC. Alle anderen Einträge unbenutzt.
1	uint8_t	Indikator	Muss =0 sein
2	uint8_t	Jahr	Kodiert als Binärwert, der einen Summanden zum Jahr 2000 angibt. In 2017 ist der Wert z.B. =17. Die Alarmzeit (Index=1) hat kein Jahr; der Wert ist dann =0 und zu ignorieren.
3	uint8_t	Monat	Monat von 1 (Januar) bis 12. Nur Index 0
4	uint8_t	Tag	Tag des Monats von 1 bis 31 bzw UTC-of. VZ
5	uint8_t	Stunde	Stunde des Tages von 0 bis 23
6	uint8_t	Minute	Minute von 0 bis 59
7	uint8_t	Sekunde	Sekunde von 0 bis 59. Nur Index 0
0 Byte	0 Rückgaben		

0x6C	W (CAN)	WriteClockTime	GSV-6
Datum und Uhrzeit der RTC oder Alarmintervall des RTC-Alarms setzen Es **muss** immer zuerst ein Befehl mit der Ersten Hälfte und dann mit der zweiten Hälfte versendet werden, damit der Wert gespeichert wird! **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardware eine RTC enthält (Andernfalls wird eine Fehlermeldung gesndet).			
5 Bytes	5 Parameter		
0	uint8_t	Index	0..1 0: Aktuelle Uhrzeit. 1: Alarmintervall (Tag, Stunde, Minute)
1	uint8_t (1)	Indikator	==1 ‚Erste Hälfte': Jahr/Monat/Tag ==2 ‚Zweite Hälfte': Std./Min./Sek.
2	uint8_t	Jahr/Stunde	siehe seriell
3	uint8_t	Monat/Minute	siehe seriell
4	uint8_t	Tag/Sekunde	siehe seriell

11 Vorhanden ab Firmware Version 3.27

Read Logger Settings (0x6D)

0x6D	R	ReadLoggerSettings	GSV-6BT
Konfiguration der Messwertaufzeichnung nach Datei auf SD-Karte lesen **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardwarevoraussetzungen (RTC und SD-Karte vorhanden) erfüllt sind.			
1 Byte	1 Parameter		
0	uint8_t	Index	0..4 0: Basic Flags, z.T. Read-only 1: Alarm Mode 2: B<3:0>: DirectoryCreationMode (Verzeichniserstellung), B<7:4>: File finalization mode (Dateibeendigung), B<12:8>: Content formatting flags 3: File length lines (Zeilenanzahl in Datei) 4: Dezimationsdivisor Datenfrequenz zu Logging Rate 5: Multiplikationsfaktor Datenfrequenz zu Logging Rate[12]
4 Bytes	1 Rückgabe		

12 Ab Firmware-Version 3.35

0	uint32_t	Wert	**Index 0:** Basic Flags: Bit<0>: =1: Geeignete SD-Karte gesteckt (read-only) Bit<1>: =1: Datei aktuell zur Aufnahme geöffnet (read-only) Bit<2>: =1: Aufnahme-aktiv-Zustand (nichtflüchtig gespeichert) Bit<4>: =1: permanente Datenaufzeichnung =0: Einzelwertaufzeichnung Bit<5>: =1: Aufzeichnung per Triggerbedingung (z.Zt. nur Digitaleingang) Bit<6>: =1: Aktuell werden Messdaten aufgezeichnet (read-only) **Index 1:** Bit<0>: =1: Alarm Mode: RTC aktiviert Alarm-leitung 1x pro Minute, sobald Sekundenwert =0 **Index 2:** Bit<0>: Directory creation mode: Legt fest, wann bei Aufzeichnungsbeginn ein neues Verzeichnis angelegt wird: =0: Neues Verzeichnis jeden Tag =1: Neues Verzeichnis jeden Monat (RdOnly, stets=1) Bit <1>: =1: Bei Einzelwertaufzeichnung wird jeden Tag eine neue Datei erstellt. =0: es wird stets an die bestehende Datei angehängt Bit<2>: Bei Aufzeichnung per Triggerbedingung: =1: An bestehende Datei anhängen. =0: Bei Zutreffen der Triggerbedingung wird eine neue Datei erstellt Bit<8>: =1: Print Date to each line Bit<9>: =1: Print Time to each line (RdOnly, stets=1) Bit<11>: =1: Print Header. Default=1. Ohne Header (=0) nicht empfohlen, da so für eine spätere Konvertierung der Datei in andere Formate Informationen fehlen. **Index 3:** File length lines: Maximale Zeilenanzahl pro Datei **Index 4:** Dezimationsdivisor N: Ist dieser >1, wird bei permanenter Datenaufzeichnung nur jeder N-te Messwert geloggt (W: 1..65535) **Index 5:** Multiplikationsfaktor N: Ist dieser >1, ist bei permanenter Datenaufzeichnung die Logging-Rate N-mal höher als die Kommunikations-Datenrate[12]

Write Logger Settings (0x6E)

0x6E	W	WriteLoggerSettings	GSV-6BT
Messwertaufzeichnung in Datei auf SD-Karte konfigurieren **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardwarevoraussetzungen (RTC und SD-Karte vorhanden) erfüllt sind.			
5 Bytes	2 Parameter		

0x6E	W	WriteLoggerSettings	GSV-6BT
0	uint8_t	Index	0..4 0: Basic Flags, z.T. Read-only 1: Alarm Mode 2: B<3:0>: DirectoryCreationMode (Verzeichniserstellung), B<7:4>: File finalization mode (Dateibeendigung), B<12:8>: Content formatting flags 3: File length lines (Zeilenanzahl in Datei) 4: Dezimationsfaktor Datenfrequenz zu Logging Rate
1	uint32_t	Wert	Siehe ReadLoggerSettings S. 70
0 Bytes	0 Rückgabe		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Control Logger (0x6F)

0x6F	A	ControlLogger	GSV-6BT
Messwertaufzeichnung in Datei auf SD-Karte Starten/Beenden **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardwarevoraussetzungen (RTC und SD-Karte vorhanden) erfüllt sind.			
1 Byte	1 Parameter		
0	uint8_t	Action-Enum	=**0**: Aufnahme beenden =1: Einzelwert aufnehmen, d.h. eine Zeile mit Messwerten der konfigurierten Eingangskanäle =**2**: Permanente Aufnahme mit eingestellter Datenrate (/Dezimationsfaktor) starten =**3**: Neue Datei zur Aufnahme öffnen. Hierdurch kann z.B. ein anschließender Befehl zur Einzelwertaufnahme (oder per Leitungs-Trigger) schneller ausgeführt werden. Wenn vorher Werte an die bestehende Datei angehängt wurden, wird hiermit eine neue Datei angelegt und geöffnet. =**4**: Dateisystem neu initialisieren (falls vorher deinitialisiert oder Karte ausgeworfen)[13] =**5**: Dateisystem deinitialisieren & SD-Hardware freigeben, zB für Verwendung durch anderen Master 8) =**0x40** RTC-Alarm nur zurücksetzen =**0x41** RTC-Alarm zurücksetzen und Alarm-Intervall neu starten
0 Byte	0 Rückgaben		

QueryFileSystem (0x70)

0x70	R	QueryFileSys	GSV-6BT
Datei- u. Verzeichnisnamen auf der SD-Karte ermitteln **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardwarevoraussetzungen (RTC und SD-Karte vorhanden) erfüllt sind.			
2 Byte	2 Parameter		
0	uint8_t	Steuerwert	=0: Start; =1: nächster Eintrag =0xFE: letzter Verzeichnisname des Dateloggers =0xFF: letzter Dateiname des Dateloggers
1	uint8_t	Verzeichnisebene	=0: Root-Verzeichnis. Hiermit beginnen; nach Öffnen eines Verzeichnisses darin: =1, usw. Hinweis: Die Länge des internen Pfad-Strings ab Root darf 255 Zeichen nicht überschreiten!
14 Bytes	2 Rückgaben		
0	uint8_t	Flags	Bit 0: =1: IsDirectory =0: IsFile Bit 1: =1: Has Known Name Pattern Bit 2: =1: Has LFN (long file name) (reserviert) Bit 7: =1: No Entry exists anymore in current directory
13	String	Name	Name innerhalb des aktuellen Verzeichnisses: Dateiname 8.3, Null-terminiert oder Verzeichnisname 8 Chars, Null-terminiert

13 Vorhanden ab Firmware Version 3.27

OpenFileDir (0x71)

0x71	A	OpenFileDir	GSV-6BT
Datei oder Verzeichnis auf SD-Karte zum Lesen öffnen **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardwarevoraussetzungen (RTC und SD-Karte vorhanden) erfüllt sind.			
14 Bytes	2 Parameter		
0	uint8_t	Steuerwert	=0: Datei öffnen, =1: Verzeichnis öffnen =2: Zuletzt gespeicherte Datei öffnen (wenn vorhanden). Name wird dabei ignoriert
13	String	Name	Dateiname 8.3, Null-terminiert Verzeichnisname 8 Chars, Null-terminiert
0 Bytes	0 Rückgaben		

GetFileInfo (0x72)

0x72	R	GetFileInfo	GSV-6BT
Informationen zu geöffneter oder zuletzt per Query ermittelter Datei oder Verzeichnis lesen, und zwar Flags, Größe und zuletzt-geändert-Zeitstempel **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardwarevoraussetzungen (RTC und SD-Karte vorhanden) erfüllt sind.			
0 Bytes	0 Parameter		
12 Bytes	9 Rückgaben		
0	uint8_t	Level	Verzeichnistiefe, z.B. =0: im Root-Dir
1	uint8_t	Flags	Bit0: =1: isDirectory, =0: isFile Bit1: =1: Read only Bit2: =1: Hidden Bit3: =1: System Bit6: =1: Archive
2	Uint32 (4)	Size	Dateigröße in Bytes. Bei Verzeichnis: =0
3	uint8_t	Jahr	Kodiert als Binärwert, der einen Summanden zum Jahr 2000 angibt. In 2017 ist der Wert z.B. =17. Hinweis: Der Datumsbereich von 1980 bis 1999 wird als Jahr=0 angegeben.
4	uint8_t	Monat	Monat von 1 (Januar) bis 12
5	uint8_t	Tag	Tag des Monats von 1 bis 31
6	uint8_t	Stunde	Stunde des Tages von 0 bis 23
7	uint8_t	Minute	Minute von 0 bis 59
8	uint8_t	Sekunde	Sekunde von 0 bis 58

Read File (0x73)

0x73	R	ReadFile	GSV-6BT
Datei auf SD-Karte Lesen. Voraussetzung: Datei zum Lesen geöffnet. **Hinweis**: Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardwarevoraussetzungen (RTC und SD-Karte vorhanden) erfüllt sind. - Nach Lesen muss die Datei mit ControlLogger (Parameter=0) geschlossen werden!			
0 Bytes	0 Parameter		
0..15 Bytes	0 oder 1 Rückgabe		

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x73	R	ReadFile	GSV-6BT
0..15	uint8_t[0..15]	Gelesene Daten	Es werden bei aufeinanderfolgenden Aufrufen bis zu 15 Bytes an Daten gelesen. I.d.R. ist die Datei größer als 15 Bytes, d.h. es werden bei jedem Aufruf die jeweils nächsten 15 Bytes nacheinander geliefert; beim letzten weniger als 15, d.h. 0..14.

Read File Extended (0x74)

0x74	R	ReadFileExtended	GSV-6BT
Datei auf SD-Karte Lesen. Voraussetzung: Datei zum Lesen geöffnet, permanente Messdatenübertragung gestoppt. Der Dateidownload kann mit diesem Kommando ca. 6-fach schneller realisiert werden als mit ReadFile. **Hinweise**: 1. Der Befehl ist erst ab Firmware-Version 3.14 verfügbar, wenn die Hardwarevoraussetzungen (RTC und SD-Karte vorhanden) erfüllt sind. 2. Nach Lesen muss die Datei mit ControlLogger (Parameter=0) geschlossen werden! 3. Wenn die permanente Messdatenübertragung aktiv ist oder während Lesen einer Datei andere Kommandos ausgeführt werden (was u.U. abgewiesen wird), werden nur 15 Bytes gelesen.			
2 Bytes	1 Parameter		
0	uint16_t	Buffer size	Maximale Anzahl der zu lesenden Bytes. Wertebereich: 1..270.
0..15 Bytes	0 oder 1 Rückgabe		
0..270	uint8_t (1) [0..269]	Gelesene Daten	Bis zu <BufferSize> oder 270 Datenbytes (der kleinere Wert gilt) werden bei aufeinanderfolgenden Kommandoaufrufen gelesen. Wenn eine Datei größer als 270 bzw <BufferSize> Bytes ist, gibt jeder Aufruf die nächsten Bytefolge zurück und der letzte Aufruf weniger als 270 bzw <BufferSize> Bytes, z.B. 0 bis 269 Bytes. Siehe Kap. Seriell-Antwort-Frame für eine Beschreibung der langen Antwort und Dekodierung der tatsächlich gelesenen Byteanzahl.

Read Value String (0x75)

0x75	R	ReadValueString	GSV-6
Messwert als Textstring lesen. **Hinweise**: - Der Befehl ist erst ab Firmware-Version 3.13 verfügbar - Beim Versuch, einen nicht konfigurierten Kanal zu lesen, wird ein leerer String zurückgegeben - Wenn Kanal=0, muss die permanente Messdatenübertragung aus sein. Bit 1 der Formatierungsflags muss dann =0 sein.			
2 Bytes	2 Parameter		
0	uint8_t	Kanal	Kanal-Nr. 0..9. 7: Zähler/Counter. 8: Temperatur 9: Betriebsspannung GSV-6CPU (ab FW-ver. 3.20) 0: Alle konfigurierten Messwertkanäle, d.h. bis zu 7 Kanäle lesen (ab FW-ver. 3.20)
1	uint8_t	Formatierungsflags	Bit 0: =1: Einheit mit ausgeben. Bit 1: =1: Feste Gesamtbreite der Antwort= 15 Bytes. =0: Antwortlänge variabel von 1 bis 15 bzw. 1 bis 127 Bytes

0x75	R	ReadValueString	GSV-6
1..15 Bytes bzw. 1..127 Bytes	1 Rückgabe		
1..15 / 1..127	uint8_t[1..15] / uint8_t[1..127]	Messwerttext	Wenn der angeforderte Eingangskanal nicht konfiguriert (dh nicht vorhanden) ist, wird ein Leerstring ausgegeben (1. Zeichen=0x00) Andernfalls beginnt der Messwert stets mit dem Vorzeichen '+' oder '-', dann folgen die Vorkommastellen (mindestens eine), dann der Dezimalpunkt '.', dann die Nachkommastellen. Es werden insgesamt 5 Dezimalstellen ausgegeben, wenn der gültige Maximalwert <= 99999 ist, andernfalls die nötigen Vorkommastellen ohne Nachkommastellen. Der gültige Maximalwert ist der UserScale-Wert oder (bei aktiver Sechsachsensensormessung) der entsprechende Maximalwert der 6-Achsensensorkonfiguration. Wenn eine Einheit ausgegeben werden soll, ist diese mit einem Leerzeichen ' ' vom Messwert getrennt. Wenn der String 15 Bytes lang sein soll, wird die Einheit ggf. verkürzt. Der String ist 0-terminiert **oder** genau 15 Zeichen lang (kann bei langer Einheit auftreten) und enthält nie Zeilenvorschubzeichen. Wenn der String mehrere Messwerte enthält (Kanal=0), sind diese mit dem Tabulatorzeichen getrennt. Wenn Kanal=0 ist oder eine benutzerdefinierte (d.h. ggf. bis zu 8 Bytes lange) Einheit mitausgegeben werden soll, wird versucht, einen langen Antwortframe zu versenden, siehe Kap. Seriell-Antwort-Frame. Dieser kann bis zu 127 Bytes lang sein.

ME Prepare Special Mode (0x77)

0x77	A	Reset Device	GSV-6
Spezielle Aktion / Zustand vorbereiten, z.B. Bootloader Der Befehl ist erst ab Firmware-Version 3.27 verfügbar. Vorr.: Hersteller-ID gesetzt			
4 Bytes	1 Parameter	Signatur	Festgelegte Sonderwerte, die die Funktion auslösen, bzw vorbereiten.
0 Bytes	0 Rückgabe		

Reset Device (0x78)

0x78	A	Reset Device	GSV-6
Den GSV-6 Neustarten. **Hinweis**: Der GSV-6 sendet auf diesen Befehl ausnahmsweise **keinen Antwortframe!** Der Befehl ist erst ab Firmware-Version 3.10 verfügbar			
0 Bytes	0 Parameter		
0 Bytes	0 Rückgabe		

Release Interface (0x7A)

0x7A	A	ReleaseInterface	GSV-8
Dieses verwendete Kommunikationsinterface freigeben. Sollte am Ende einer Kommunikationssession aufgerufen werden, um anderen Interfaces ggf. das Schreibrecht zu geben.			
0 Byte	0 Parameter		
0 Byte	0 Rückgaben		

ReadInterfaceSetting (0x7B)

0x7B	R	ReadInterfaceSetting	GSV-6 / GSV-8
Einstellungen der Kommunikationsinterfaces bestimmen			
1 Byte	1 Parameter		
0	uint8_t	Index / Interface-Nr.	0.. Anzahl vohandener Interfaces-1: Basic settings. Darüber: Extended settings
6 Bytes	3 Rückgaben		
0	uint8_t	next index	Nächster Index in der verketteten Liste. =0: Letzter Eintrag.
1	uint8_t	Phys.Typ / Dat.typ	Basic settings: Physikalischer Schnittstellentyp (Enum) Extended settings: Bit 7: =1: Eintrag schreibbar. =0: Nicht schreibbar. Bits<6:0>: Typ des Dateneintrags (Enum)
2	uint32_t	Data	Basic: Bits<31:24>: Interface-Applikationstyp (Enum) Bits<23:0>: Flags Extended: Bits<31:0>: Dateneintrag

Die Indizes sind in 2 Bereiche unterteilt. Der Indexbereich 1 reicht von 0 bis (Anzahl vorhandener Schnittstellen -1). Hier können "Basic settings" zu jeder der vorhandenen Schnittstellen ermittelt werden (s. Anhang C, S. 94). Die Anzahl vorhandener Schnittstellen kann mit dem Befehl GetInterface (0x01) ermittelt werden.

Die Rückgabe "Next index" zeigt auf den Beginn des Indexbereiches der "Extended settings". Bei diesen wird im Rückgabeparameter 1 ein Enum zurückgegeben, der bestimmt, um was für Daten es sich handelt (s. Anhang C, S. 94).

ReadInterfaceSetting und WriteInterfaceSetting sind beim GSV-6 erst ab FW-ver 3.33 vorhanden.

WriteInterfaceSetting (0x7C)

0x7C	W	WriteInterfaceSetting	GSV-6 / GSV-8
Einstellungen der Kommunikationsinterfaces ändern.			
5 Bytes	2 Parameter		
0	uint8_t	Index / Interface-Nr.	0.. Anzahl vohandener Interfaces-1: Basic settings. Darüber: Extended settings

0x7C	W	WriteInterfaceSetting	GSV-6 / GSV-8
1	uint32_t	Data	Basic: Bits<31:24>: Interface-Applikationstyp (Enum) Bits<23:0>: Flags Extended: Bits<31:0>: Dateneintrag
0 Byte	0 Rückgaben		

PrepReadFTsensor (0x7D)

0x7D	A	ReadDataRateRange	GSV-6 / GSV-8
FT-Kalibrierarray zum Lesen auswählen. Nachfolgende Aufrufe von ReadFTsensorCal lesen aus diesem Array. Die Ausführung hat keine Auswirkung auf die aktuelle Sechachsensensorberechnung.			
1 Byte	1 Parameter		
0	uint8_t	Index	Array-Index [0-4]
0 Byte	0 Rückgaben		

StoreDigiFilt (0x7E)

0x7E	W	StoreDigiFilt	GSV-8
Speichern der Digitalfilterparameter aller Kanäle. Die beiden Parameter werden beim GSV-8 ignoriert.			
2 Bytes	2 Parameter		
0	uint8_t (1)		
1	uint8_t (1)		
0 Byte	0 Rückgaben		

0x7E	W	StoreDigiFilt	GSV-6
Speichern der Digitalen-Filter im Flash von einem (Kanal-Beginn==Kanal-Ende) oder mehrerer Kanäle (Kanal-Beginn!=Kanal-Ende).			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal-Beginn	Start-Index (0-5)
1	uint8_t (1)	Kanal-Ende	Stop-Index (0-5)
0 Byte	0 Rückgaben		

StoreFTSensorCal (0x7F)

0x7F	W	StoreSensorCal	GSV-6 / GSV-8
Speichern einer 6-Achsen Kalibriermatrix, die im Matrix-Ram liegt, auf Index im Flash. Voraussetzung: Benutzer-ID gesetzt			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	GSV-6: Array-Index [0-7] GSV-8: Array-Index [0-4 / 0-10] oder Sonderwert 0xFF. Es können bestehende Datenarrays überschrieben oder neue hinten angehängt werden, „Speicherlöcher“ sind nicht erlaubt. Mit dem Sonderwert 0xFF kann das letzte Array gelöscht werden.

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0x7F	W	StoreSensorCal	GSV-6 / GSV-8
0 Byte	0 Rückgaben		

GetTXMode (0x80)

0x80	R	GetTXMode	GSV-6
Aktuelle Messwert-Frame Einstellungen abfragen			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	0: TX-Mode 1: Messwert-Datenformat 2: Maximale Kanalanzahl Alle anderen Werte sind reserviert.
2 Bytes	1 Rückgaben		
0	uint16_t (2)	Rückgabe	**Index 0**: TX-Mode Flags <0> =1: TX temporär AUS <1> =1: TX permanent AUS <2> =1: Maximalwerte im Messwertframe übertragen[14] (read-only) <4:3> reserviert <5> =1: Messwertframe enthält CRC16 Checksumme[15] <6> =1: Schreiben via UART blockiert (read only)[9] <7> reserviert <8> =1: TX-Synchonisation: Slave (read only)[9] <9> =1: TX-Synchonisation: Master (read only)[9] <15:10> reserviert **Index 1**: Messwert-Datenformat - 1: int16 - 3: float - andere Werte reserviert **Index 2**: (read-only) - Maximalanzahl der Kanäle

0x80	R	GetTXMode	GSV-8
Aktuelle Messwert-Frame Einstellungen abfragen			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	0: TX-Mode 1: Messwert-Datenformat 2: Kanalnummern-Bereich Alle anderen Werte sind reserviert.
2 Bytes	1 Rückgaben		

14 Ab Firmware-Version 3.30
15 Ab Firmware-Version 3.35

0x80	R	GetTXMode	GSV-8
0	uint16_t (2)	Rückgabe	Index **0**: TX-Mode Flags <0> =1: Messwert-Übertragung temporär AUS (read only) <1> =1: Messwert-Übertragung permanent AUS <2> =1: Maximalwerte im Messwertframe übertragen <3> =1: Minimalwerte im Messwertframe übertragen <4> reserviert, intern verwendet <5> =1: Messwertframe enthält CRC16 Checksumme[16] <6> =1: Schreiben via UART blockiert (read only) <7> =1: Schreiben via USB blockiert (read only) <8> =1: TX-Synchonisation: Slave (read only) <9> =1: TX-Synchonisation: Master (read only) <15:10> reserviert Index **1**: Messwert-Datenformat 1: int16 2: S24 3: float - andere Werte reserviert Index **2:** Vorhandene Eingangskanäle: (read-only) Bits<7:0>: Kleinste Kanalnummer (=2) Bits<15:8>: Höchste Kanalnummer (=8 / =10)

SetTXMode (0x81)

0x81	W	SetTXMode	GSV-6 / GSV-8
Setzen der Messwert-Frame Einstellungen.			
3 Bytes	2 Parameter		
0	uint8_t (1)	Index	0: TX_mode 1: Messwert-Datenformat Alle anderen Werte reserviert.
1	uint16_t (2)	Wert	siehe GetTXMode (0x80)
0 Byte	0 Rückgaben		

Bemerkung: Änderung am CRC16-Flag (Bit 5) wird erst nach einem Neustart wirksam.

Read Debounce Time (0x86)

0x86	R	ReadDebounceTime	GSV-6
Zeit in mSek lesen, die ein Digitaleingang bzw. die Taste gedrückt sein muss, bis Funktion gültig. Gilt nicht im Menü und bei schnellen Trigger-Eingängen, z.B. Sync-Slave.			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	[1-3]: 1: Entprellzeit TARA Input 2: Entprellzeit SCALE Taste 3: Entprellzeit TRIGGER Input
2 Bytes	1 Rückgaben		
0	uint16_t (2)	Entprellzeit	Entprellzeit in ms

16 Vorhanden ab Firmware Version 1.56.

Write Debounce Time (0x87)

0x87	W	WriteDebounceTime	GSV-6 (Reset)
Zeit in mSek schreiben, die ein Digitaleingang bzw. die Taste gedrückt sein muss, bis Funktion gültig. Gilt nicht im Menü und bei schnellen Trigger-Eingängen, z.B. Sync-Slave. Voraussetzung: Benutzer-ID erforderlich			
3 Bytes	2 Parameter		
0	uint8_t (1)	Index	[1-3]: 1: Entprellzeit TARA Input 2: Entprellzeit SCALE Taste 3: Entprellzeit TRIGGER Input
1	uint16_t (2)	Entprellzeit	Entprellzeit in ms [1-65535]
0 Byte	0 Rückgaben		

ReadDataRate (0x8A)

0x8A	R	ReadDataRate	GSV-6 / GSV-8
Datenrate, mit der Messwertframes ausgegeben werden, auslesen.			
0 Byte	0 Parameter		
4 Bytes	1 Rückgaben		
0	float (4)	Datenrate	Messwertframes pro Sekunde in Hz

WriteDataRate (0x8B)

0x8B	W	WriteDataRate	GSV-6 (Reset) / GSV-8
Datenrate, mit der Messwertframes ausgegeben werden sollen, setzen			
4 Bytes	1 Parameter		
0	float (4)	Datenrate	Datenrate in Hz
0 Byte	0 Rückgaben		

GetCANSetting (0x8C)

0x8C	R	GetCANSetting	GSV-6 / GSV-8
CAN Einstellung abfragen Hinweis für GSV-8: Wenn der Aufruf dieser Funktion ERR_CMD_NOTKNOWN ergibt, wird CAN/CANopen nicht unterstützt.			
1 Byte	1 Parameter		
0	uint8_t (1)	Index	0: CAN_IN Befehl Can-ID (nur GSV-6) 1: CAN_OUT Antwort Can-ID (nur GSV-6) 2: CAN_CV Messdaten-Frame Can-ID (nur GSV-6) 3: CAN_CAST Multicast Can-ID (nur GSV-6) 4: CAN_BAUD Baudrate (in Bits/s) 5: CAN_FLAGS 6: CANopen Node-ID (nur GSV-8) Alle anderen Werte sind reserviert
4 Bytes	1 Rückgaben		
0	uint32_t (4)	Can-ID Baudrate Flags	Can-ID: 0x00000000-0x000007FF: Std-ID 0x80000000-0x9FFFFFFF: Ext-ID (nur GSV-6) Baudraten (GSV-6): 1000000, 500000, 250000, 125000, 100000, 50000,

0x8C	R	GetCANSetting	GSV-6 / GSV-8
			25000, 12500 Baudraten (GSV-8): 1000000, 500000, 250000, 125000, 50000 Flags: <31:2> Reserviert (=0) Bit 0: CAN on / off:[17] =**1**: CAN **aus**geschaltet =0: CAN eingeschaltet Bit 1: CAN-Applikationsprotokoll (read-only) =1: CANopen aktiv =0: Proprietätes CAN (wie hier beschrieben)

SetCANSetting (0x8D)

0x8D	W	SetCANSetting	GSV-6 (Reset) / GSV-8 (ggf. Reset)
CAN Einstellungen ändern. Hinweis: Beim GSV-8 können Baudrate und CAN- bzw. NodeIDs nur bei ausgeschaltetem CAN-Interface eingestellt werden.			
5 Bytes	2 Parameter		
0	uint8_t (1)	Index	siehe GetCANSetting (0x8C)
1	uint32_t (4)	Can-ID, Baudrate, Flags	siehe GetCANSetting (0x8C)
0 Byte	0 Rückgaben		

ReadAnalogueFilter (0x90)

0x90	R	ReadAnalogueFilter	GSV-8
Analog-Filter Frequenz auslesen			
0 Byte	0 Parameter		
2 Bytes	1 Rückgaben		
0	uint16_t (2)	Filter-Frequenz	Grenzfrequenz in Hz

WriteAnalogueFilter (0x91)

0x91	W	WriteAnalogueFilter	GSV-8
Grenzfrequenz des analogen Vorfilters setzen.			
4 Bytes	1 Parameter		
0	float (4)	Grenzfrequenz in Hz	Es können 3 verschiedene Frequenzen gesetzt werden: - 28 Hz - 885 Hz – 11,4 kHz Bei Übergabe anderer Werte wird die nächste vorhandene Frequenz gesetzt

17 Bei GSV-6 erst ab Firmware-version 3.25

0x91	W	WriteAnalogueFilter	GSV-8
0 Byte	0 Rückgaben		

SwitchBlocking (0x92)

0x92	W	SwitchBlocking	GSV-6 / GSV-8
Schreibschutz setzen oder löschen Voraussetzung: Benutzer-ID erforderlich			
4 Bytes	1 Parameter		
0	uint32_t (4)	Schutz-Code	BLOCKING_UNLOCK_THIS 0x554c4b74 = "ULKt" (reserviert bei GSV-8) BLOCKING_UNLOCK_ALL 0x554c4b61 = "ULKa" BLOCKING_LOCK_THIS 0x426c6b54 = "BlkT" (reserviert bei GSV-8) BLOCKING_LOCK_ALL 0x426c6b41 = "BlkA"
0 Byte	0 Rückgaben		

GetCommandAvailable (0x93)

0x93	R	GetCommandAvailable	GSV-6 / GSV-8
Überprüft, ob alle Kommandos im command-ID-bereich HighCmd bis einschließlich LowCmd implementiert sind, und liefert den entsprechenden Fehler-Code. ***Achtung***: LowCmd muss kleiner oder gleich HighCmd sein!			
2 Bytes	2 Parameter		
0	uint8_t (1)	HighCmd	Obere Kommando Nummer (inklusive)
1	uint8_t (1)	LowCmd	Untere Kommando Nummer (inklusive)
0 Byte	0 Rückgaben (Information im ErrorByte des Antwortframes)		Information im Error-Byte: ERR_OK: Alle Befehle vorhanden. ERR_CMD_NOTIMPL: Ein oder mehrere Befehle nicht vorhanden.

Read NoiseCut Threshold (0x94)

0x94	R	ReadNoiseCutThreshold	GSV-8
Abfragen der Schwelle für die Rauschunterdrückung. Unterhalb dieser Schwelle und oberhalb der negierten Schwelle (d.h. „um 0 herum“) werden Messwerte =0 gesetzt.			
1 Byte	1 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer [1-8]
4 Bytes	1 Rückgabe		
0	float (4)	Schwelle	

Write NoiseCut Threshold (0x95)

0x95	W	WriteNoiseCutThreshold	GSV-8
Setzen der Schwelle für die Rauschunterdrückung Unterhalb dieser Schwelle und oberhalb der negierten Schwelle (d.h. „um 0 herum“) werden Messwerte =0 gesetzt.			
5 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer [0-8] 0: Alle auf denselben Wert setzen.
1	float (4)	Schwelle	
0 Byte	0 Rückgaben		

Read AutoZero Setting (0x96)

0x96	R	ReadAutoZeroSetting	GSV-8
Abfragen der Zählperiode oder der Schwelle für die automatische Nullsetzfunktion Vorhanden ab FWver 1.39			
2 Bytes	2 Parameter		
0	uint8_t (1)	Typ	=0: Auto-Zero Zählperiode in Sekunden (Kanal wird ignoriert) =1: Auto-Zero Schwellwert
1	uint8_t (1)	Kanal	Kanalnummer [1-8] (wenn Typ=1)
4 Bytes	1 Rückgabe		
0	float (4)	Periode / Schwelle	

Write AutoZero Setting (0x97)

0x97	W	WriteAutoZeroSetting	GSV-8
Setzen der Periode oder der Schwelle für die automatische Nullsetzfunktion. Wenn der Messwert für <Periode> Sekunden unterhalb seines Schwellwertes bleibt, werden bestimmte Kanäle (einstellbar mit SetAutoZeroProperty) Null gesetzt, wenn Bit 22 der Mode-Variablen gesetzt ist, s.GetMode 41 Vorhanden ab FWver 1.39			
6 Bytes	3 Parameter		
0	uint8_t (1)	Typ	=0: Auto-Zero Zählperiode in Sekunden (Kanal wird ignoriert) =1: Auto-Zero Schwellwert
1	uint8_t (1)	Kanal	Kanalnummer [1-8] (wenn Typ=1)
2	float (4)	Wert	Periode / Schwelle
0 Byte	0 Rückgaben		

Get AutoZero Property (0x98)

0x98	R	GetAutoZeroProperty	GSV-8
Abfragen von Flags, die das Verhalten der Auto-Zero Funktion bestimmen Vorhanden ab FWver 1.39			
2 Bytes	2 Parameter		

0x98	R	GetAutoZeroProperty	GSV-8
0	uint8_t (1)	Typ	**=0: ApplyAZ**: Flags, die festlegen, welche Kanäle Nullgesetzt werden sollen, bzw (mit Mode=1), an welchen Kanälen die Nullsetzfunktion aktiv ist. **=1: UseThreshold**: Mit Mode=0: Flags, die festlegen, welche Kanäle mit ihrer Schwelle verglichen werden, deren Ergebnis dann verUNDet bestimmt, ob die Nullsetzfunktion ausgeführt wird. **=2: Mode**: Bit 0: =0: Kanäle gruppiert, d.h. wenn alle Schwellwert- und Zeitbedingungen zutreffen (s. UseThreshold), werden die mit ApplyAZ demensprechend konfigurierten Kanäle Nullgesetzt. Bit 0 =1: Auto-Zero wird auf alle mit ApplyAZ aktivierten Kanäle unabhängig voneinander angewendet (UseThreshold wird ignoriert).
1	uint8_t (1)	Kanal	Reserviert, z.Zt. ignoriert
2 Bytes	1 Rückgabe		
0	uint16_t (2)	Flags	Typ =0 und =1: Kanalflags: Bit 0: Bedingung/Aktion wird auf Kanal 1 angewendet, usw bis Bit 7: wird auf Kanal 8 angewendet.

Set AutoZero Property (0x99)

0x99	W	WriteNoiseCutThreshold	GSV-8
Setzen von Flags, die das Verhalten der Auto-Zero Funktion bestimmen Vorhanden ab FWver 1.39			
4 Bytes	3 Parameter		
0	uint8_t (1)	Typ	siehe GetAutoZeroProperty
1	uint8_t (1)	Kanal	Reserviert, z.Zt. ignoriert
2	uint16_t (2)	Flags	Typ =0 und =1: Kanalflags: Bit 0: Bedingung/Aktion wird auf Kanal 1 angewand, usw bis Bit 7: wird auf Kanal 8 angewand.
0 Byte	0 Rückgaben		

ReadUserOffset (0x9A)

0x9A	R	ReadUserOffset	GSV-6 / GSV-8
Benutzerdefinierten Offset auslesen			
1 Byte	1 Parameter		

0x9A	R	ReadUserOffset	GSV-6 / GSV-8
0	uint8_t (1)	Kanal	Kanalnummer GSV-6: [1-6] / [1-7] (FW-ver >=3.11) GSV-8: [1-10]
4 Bytes	1 Rückgaben		
0	float (4)	Offset	Offset des Kanals

WriteUserOffset (0x9B)

0x9B	W	SetUserOffset	GSV-6 / GSV-8
Benutzerdefinierten Offset setzen **Hinweis für GSV-6:** Wenn die 6-Achsen-Berechnung aktiv ist, haben diese Werte keine Auswirkung!			
5 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer GSV-6: [0-6] / [0-7] (FW-ver >=3.11) GSV-8: [0-10] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.
1	float (4)	Offset	
0 Byte	0 Rückgaben		

GetInputType (0xA2)

0xA2	R	GetInputType	GSV-6
Aktuellen Eingangstyp abfragen Bei GSV-6 wird hiermit der elektrische Gesamt-Messbereich des Analogeingangs bestimmt.			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer GSV-6: [1-6] / [1-7] (FW-ver >=3.11)
1	uint8_t (1)	Res.	=0 (ungenutzt)
4 Bytes	2 Rückgaben		
1	uint8_t (1)	Type	=0: Analoger Eingang. =6: Counter Eingang
1	U24 (3)	Sens	Eingangsmessbereich in mV/V*100

0xA2	R	GetInputType	GSV-8
Aktuellen Eingangstyp abfragen Bei GSV-8 werden hiermit die möglichen Messbereiche der analogen Eingangsbeschaltung abgefragt, die zu dem angeschlossenen Sensor zusammenpassen sollte. Es können sowohl alle vorhandenen Messbereiche abgefragt werden, als auch der aktuell gesetzte Messbereich und Eingangstyp.			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer [1-8]

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

0xA2	R	GetInputType	GSV-8
1	uint8_t (1)	SensIndex	[0xFF..0x05] Der Sensindex ist wie folgt definiert: 0xFF Aktuell eingestellten Eingangstyp und dessen Messbereich lesen. Wenn <0xFF, wird der abzufragende Type-Enum übergeben: 0x00 Eingangsmessbereich für Eingangstyp = Brückensensor mit Speisespannung 8,75V lesen 0x01 Eingangsmessbereich für Eingangstyp = Brückensensor mit Speisespannung 5V lesen 0x02 Eingangsmessbereich für Eingangstyp = Brückensensor mit Speisespannung 2,5V lesen 0x03 Eingangsmessbereich für Eingangstyp = Single-ended Input lesen 0x04 Eingangsmessbereich für Eingangstyp = Temperatursensor PT1000 lesen 0x05 Eingangsmessbereich für Eingangstyp = Temperatursensor K-Type lesen, absolut. 0x06: Eingangsmessbereich für Eingangstyp = Temperatursensor K-Type lesen, relativ. 0x07: Eingangsmessbereich f. Counter/Frequenz lesen
5 Bytes	2 Rückgaben		
0	uint8_t (1)	Type	Aktueller Eingangs-Vorbeschaltungstyp Eingangstyp wie oben unter SensIndex definiert (Bereich: 0..7)
1	Uint32_t (4)	Sens	Eingangsmessbereich in mV*100 (bzw. mV/V*100). Counter/ Frequenz: Maximum*100. Temperatur: °C*100

SetInputType (0xA3)

0xA3	W	SetInputType	GSV-8
Ändern des Eingangstyps Hinweis: Beim Ändern des Eingangstyps werden die Kalibrier-Nullpunkte (Hersteller) und die Default-UserScale Werte geladen, die vorherigen Parameter für Nullpunkt und UserScale werden dabei überschrieben.			
2 Bytes	2 Parameter		
0	uint8_t (1)	Kanal	Kanalnummer GSV-8: [0-8] Mit Kanal 0 werden ALLE Kanäle mit dem Identischen Wert beschrieben.

0xA3	W	SetInputType	GSV-8
1	uint8_t	Eingangstyp	Der Eingangstyp ist wie folgt definiert: 0x00: Eingangstyp = Brückensensor mit Speisespannung 8,75V 0x01: Eingangstyp = Brückensensor mit Speisespannung 5V 0x02: Eingangstyp = Brückensensor mit Speisespannung 2,5V 0x03: Eingangstyp = Single-ended Input 0x04: Eingangstyp = Temperatursensor PT1000 0x05: Eingangstyp = Temperatursensor K-Type absolut (erfordert PT1000 Sensor an Kanal 8), unterstützt ab FWver 1.39 0x06: Eingangstyp = Temperatursensor K-Type relativ (kein PT1000 Sensor erforderlich), unterstützt ab FWver 1.39
0 Byte	0 Rückgaben		

Anhang

A: Physikalische Einheiten (Codes)

In der folgenden Tabelle sind die Standard ME-Codes für die Einheiten aufgeführt:

Einheit	Code
mV/V	0
kg	1
g	2
N	3
cN	4
V	5
µm/m	6
(keine)	7
t	8
kN	9
lb	10
oz	11
kp	12
lbf	13
pdl	14
mm	15
m	16
cNm	17
Nm	18
°C	19
°F	20
K	21
oztr	22
dwt	23
kNm	24
%	25
‰	26
W	27
kW	28
rpm	29
bar	30
Pa	31
hPa	32
MPa	33
N/mm²	34
°	35
Hz	36
m/s	37
km/h	38
m³/h	39
mA	40
A	41
m/s²	42

Einheit	Code
flbs	43
ftlb	44
J	45
kWh	46
UnitText 2	254 (-2)
UnitText 1	255 (-1)

Tabelle: Einheiten-Tabelle

B: Typen der digitalen Ein- und Ausgänge

Es können folgende Funktionen konfiguriert werden:

Nr	Funktion	Daten-richtung	**Wert** Gerätebefehl Get/SetDIOtype = Wert DLL-Funktion GSV86get/setDIOtype	Kurzbeschreibung
0	Keine	n.a.	0x000000	Anschluss nicht vorhanden (nur GSV-6). **Read-only**
1	General-Purpose Input	Eingang	0x000004	Allgemeiner Eingang. Logikpegel kann mit GetDIOlevel / GSV86getDIOlevel abgefragt werden. Ist Defaultwert.
2	Nullsetzen Einzelkanal	Eingang	0x000010	Aktiver Input-Pegel setzt einen analogen Eingangskanal Null.
3	Nullsetzen alle Kanäle	Eingang	0x000020	Aktiver Input-Pegel setzt alle analogen Eingangskanäle Null.
4	Rücksetzen der Maximal- und Minimalwert-ermittlung	Eingang	0x000040	Aktiver Input-Pegel setzt alle Maximal- und Minimalwerte zurück.
5	Rücksetzen der Digitalausgänge zu Default-Pegel	Eingang	0x000050	Aktiver Input-Pegel setzt alle als Ausgang konfigurierten DIOs auf den Default-Level
6	Trigger Send actual value	Eingang	0x000080	Auslösen des Sendens eines Messwertframes mit aktuellen Messwerten über die USB-Schnittstelle an inaktiv-zu-aktiv Flanke des digitalen Eingangs. **Nur GSV-8**
7	Trigger Send maximum value	Eingang	0x000100	Bei inaktiv-zu-aktiv Flanke am digitalen Eingang wird die Maximalwertermittlung (alle Eingangskanäle) begonnen und an aktiv-zu-inaktiv Flanke wird ein Frame

Nr	Funktion	Daten-richtung	**Wert** Gerätebefehl Get/SetDIOtype = Wert DLL-Funktion GSV86get/setDIOtype	Kurzbeschreibung
				mit diesen Maximalwerten an die USB-Schnittstelle gesendet. **Nur GSV-8**
8	Trigger Send minimum value	Eingang	0x000200	Bei inaktiv-zu-aktiv Flanke am digitalen Eingang wird die Minimalwertermittlung (alle Eingangskanäle) begonnen und an aktiv-zu-inaktiv Flanke wird ein Frame mit diesen Minimalwerten an die USB-Schnittstelle gesendet. **Nur GSV-8**
9	Trigger Send mean value	Eingang	0x000400	Bei inaktiv-zu-aktiv Flanke am digitalen Eingang wird eine dezimierende Mittelwertbildung (alle Eingangskanäle) begonnen und an aktiv-zu-inaktiv Flanke wird ein Frame mit diesen Mittelwerten an die USB-Schnittstelle gesendet. **Nur GSV-8**
10	Trigger Send actual value	Eingang	0x000800	Während der Input-Pegel aktiv ist, werden Messwertframes mit aktuellen Messwerten über die USB-Schnittstelle gesendet, mit der eingestellten Datenrate. **Nur GSV-8**
11	Sync Slave	Eingang	0x000002	An Low-to-High Flanke wird der zuletzt ermittelte Messwertframe übertragen. Nur zur Verwendung mit einem zweiten GSV-8, der als Sync-Master konfiguriert ist. Dessen Sync-Output (s. Nr. 19) muss mit dem Sync-Eingang des oder der Slave(s) verdrahtet sein. Dieser Typ kann nicht verwendet werden, wenn ein Index-Eingang eines QEI-Encoders aktiviert ist.
12	QEI-Encoder	Eingang	0x000008	Eingang für Quadratur-Zähler / Frequenzmessung. **Read-Only**. Zum Ändern muss der Befehl Write Counter/Freq Mode an Index 0 verwendet werden. **Nur GSV-8**

Nr	Funktion	Daten-richtung	**Wert** Gerätebefehl Get/SetDIOtype = Wert DLL-Funktion GSV86get/setDIOtype	Kurzbeschreibung
13	File-Logging	Eingang	0x000001	Trigger-Eingang zur Messdatenaufzeichnung auf Datei. **Nur GSV-6** mit Logger-Funktion.
14	General-Purpose Output	Ausgang	0x001000	Allgemeiner Ausgang. Aktueller Logikpegel kann mit SetDIOlevel / GSV86setDIOlevel festgelegt werden.
15	Threshold output aktueller Messwert	Ausgang	0x010000	Schwellwertausgang: Ausgang wird aktiviert, wenn der zugeordnete Messwert größer als der obere Schwellwert ist und deaktiviert, wenn er kleiner als der untere Schwellwert ist.
16	Threshold output Maximalwert	Ausgang	0x014000	Schwellwertausgang: Ausgang wird aktiviert, wenn der zugeordnete Maximalwert größer als der obere Schwellwert ist und deaktiviert, wenn er kleiner als der untere Schwellwert ist.
17	Threshold output Minimalwert	Ausgang	0x018000	Schwellwertausgang: Ausgang wird aktiviert, wenn der zugeordnete Minimalwert **kleiner** als der **untere** Schwellwert ist und deaktiviert, wenn er **größer** als der **obere** Schwellwert ist.
18	Fensterkompa-ratorausgang aktueller Messwert	Ausgang	0x012000	Fensterkomparator: Ausgang wird aktiviert, wenn der zugeordnete Messwert kleiner als der obere Schwellwert und größer als der untere Schwellwert ist; sonst deaktiviert.
19	Fensterkompa-ratorausgang Maximalwert	Ausgang	0x016000	Fensterkomparator: Ausgang wird aktiviert, wenn der zugeordnete Maximalwert kleiner als der obere Schwellwert und größer als der untere Schwellwert ist; sonst deaktiviert.
20	Fensterkompa-ratorausgang Minimalwert	Ausgang	0x01A000	Fensterkomparator: Ausgang wird aktiviert, wenn der zugeordnete Minimalwert kleiner als der obere Schwellwert und größer als der untere

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Nr	Funktion	Daten-richtung	**Wert** Gerätebefehl Get/SetDIOtype = Wert DLL-Funktion GSV86get/setDIOtype	Kurzbeschreibung
				Schwellwert ist; sonst deaktiviert.
21	Sync-Master	Ausgang	0x020000	Der Sync-Master erzeugt sychron zu seiner Messwertframe-übertragung ein Synchronisations-signal, an dem die Slaves angeschlossen sind (s. Nr. 11). Bei Messwertübetragung ist der Pegelwechsel Low-zu-High, nach der halben Datenperiode High-zu-Low.

Die DIOs besitzen Pullup-Widerstände, die bei offenem Eingang High-Pegel erzeugen. Bei Eingangstrigger-Funktionen, die mit einem Ein/Aus-Schalter bedient werden sollen, ist daher der Schalter/Taster zwischen der DIO-Klemme und GNDD anzuschließen. Damit die Funktion bei geschlossenem Schalter ausgeführt wird, muss der Anschluss per Software funktional invertiert werden. Dazu ist bei Verwendung des Geräteinterfaces oder der DLL der in o.g. Spalte "Wert" genannte **mit 0x800000 zu verodern**.
Auch die Schwellwertausgänge können so invertiert werden.

In o.g. Tabelle bedeutet:

Pegel	Nicht-Invertiert	Invertiert
Aktiv	Logisch 1 = High = 5V	Logisch 0 = Low = 0V
Inaktiv	Logisch 0 = Low = 0V	Logisch 1 = High = 5V

Nur bei Verwendung der GeneralPurpose-Funktionen und der Master-Slave-Sync Funktionen (Nr. 1,11, 14 und 21 in o.g. Tabelle) hat die Invertierung keine Wirkung; die Funktionen GSV86get/setDIOlevel und Get/SetDIOlevel lesen bzw. schreiben den Pegel stets direkt, d.h. nicht-invertiert.
Bei Verwendung des Schwellwertschalters oder Fensterkomparators besteht zusätzlich die Möglichkeit, die Schwellwerte im negativen Bereich zu "spiegeln". Der Schwellwertschalter ist dann aktiviert, wenn gilt: Messwert > Obere Schwelle *oder* Messwert < -Obere Schwelle. Um diese Funktion zu aktivieren, ist der in o.g. Spalte "Wert" genannte **mit 0x400000 zu verodern**.[18] Beide Schwellwerte sollten dabei als positive Zahlen >0 gesetzt werden.

Weitere Hinweise sind in der allgemeinen Bedienungsanleitung zu finden, Kapitel "Digitale Ein- u. Ausgänge".

18 Diese Funktion ist bei GSV-6 ab Firmware-Ver. 3.29 vorhanden und beim GSV-8 ab 1.54

C: Flags und Enumerationen für Read/Write Interface Settings (Cmd 0x7B / 0x7C)

Basic Settings: Physikalischer Schnittstellentyp

Enum-No.	Bedeutung
1	RS232 Schnittstelle (asynchron, 8N1) mit V24 Pegel
2	UART Schnittstelle (asynchron, 8N1) mit 3,3V Pegel (Low=0V, High=3,3V)
3	USB Schnittstelle
4	CAN Feldbus-Schnittstelle
5	100BaseTX ("Ethernet")
6	RS422 Schnittstelle (5-Draht, asynchron, 8N1) mit differentiellem Pegel
7	SPI Schnittstelle Master, ggf. nur für Messwerte

Basic Settings: Typ des Applikationslayers

Enum-No.	Bedeutung
1	GSV8/6 Protokoll, wie in diesem Dokument beschrieben
2	CANopen
3	EtherCAT CoE
4	Textuelles "Monitor" Protokoll (nur GSV-6)
5	Skalierte SINT16 Messwerte (nur GSV-6)

Dieser Wert wird in Bits<31:24> des Datenwertes in den Basic Settings kommuniziert

Basic Settings: Flags im Flagwert

Bit-Nr.	Bedeutung
0	=1: Interface ist aktiviert und empfangsbereit
1	=1: Interface ist aktiviert und sendebereit
2	=1: Interface hat Schreibrecht
3	=1: Integer-Messdatenwert ist im Binary offset-Format (nur bei App.Enum =1 gültig) =0: Integer-Messdatenwert ist im SignedInt Format (nur bei App.Enum =1 gültig)
4	=1: Permanente Messdatenübertragung an (flüchtiger Zustand im RAM)
5	=1: Permanente Messdatenübertragung an (nicht-flüchtiger Zustand im EEPROM)
16	=1: Interface zu- und abschaltbar
17	=1: Interface verwendet Geräte- oder Dienstadressen (Feldbus)
18	=1: Adresse(n) ist/sind änderbar

Dieser Flagwert wird in Bits<23:0> des Datenwertes in den Basic settings kommuniziert

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Extended Settings: Typ des Dateninhalts (Enum)

Enum-No.	Bedeutung
1	Maske zum Setzen des BasicSettingsFlagwertes: Bit=1: Gleiche BitNo darf =1 gesetzt werden
2	Maske zum Löschen des BasicSettingsFlagwertes: Bit=1: Gleiche BitNo darf =0 gesetzt werden
3	Nr. der Schnittstelle(n), mit denen dieses wechslseitig auschließend (mutually exclusive) vorhanden ist. Dabei kann in jedem der 4 Bytes eine Schnittstellennummer >0 stehen, auffüllend beginnend mit dem LSbyte. Wert =0x00000000 bedeutet: Mit keiner Schnitttselle wechslseitig auschließend.
4	Aktive Baudrate in Bits/s. Wert =0 bedeutet: Keine Baudrate anzuwenden bzw nicht veränderbar.
5	Vorhandene Baudrate in Bits/s
6	Anzahl der aktiven Dienst-IDs bzw. (Geräte-) Adressen
7	CAN-ID des Kommandodienstes Host->Device
8	CAN-ID des Kommandodienstes Befehlsantworten Device-> Host
9	CAN-ID der Messwertframes Device-> Host
10	CAN-ID Multicast Host->Devices
11	CANopen NodeID
16	Gerätezustand, bes. bei Feldbus. s. nächste Tabelle
17	Bits<31:24>: CANopen Transmission-Type. Bits<15:0>: CANopen Event-Timer
18	Bits<31:16>: CANopen Inhibit-Time. Bits<15:0>: CANopen Heartbeat Timer
19	Anzahl der kommunizierten Kanäle. Muss <= der Kanal-Grundeinstellung sein.
20	Skalierungsfaktor bei Integer-Messwert (Float-Wert -> Int)

Dieser Wert wird in Bits<6:0> des zweiten Rückgabeparameters bei Extended-Settings Anfragen des Befehls ReadInterfaceSetting (0x7B) kommuniziert.

Kodierung des Gerätezustands

Wert	Bedeutung
0	Interface ist abgeschaltet.
2	Interface an, Zustand = "Init" / "Stopped"
4	Interface an, Zustand = "Pre-Operational"
8	Interface an, Zustand = "Safe-Operational"
12	Interface an, Zustand = "Operational"

Dieser Code wird im Datenwert bei Typ-Enum =16 in den Extended Settings kommuniziert.

D: IDs für ReadTEDSdataEntry

ID	Property name and Description
0	Placeholder in Dictionary, points to first entry index (Sonderwert: nur NextID auszuwerten)
1	TEMPLATE ID
2	Separator
3	Select Case—Physical Measurand
4	Select Case—Full-Scale Electrical Value Precision
5..9	reserviert f. weitere Select-cases u. Sonder-IDs
10	Sensitivity and mapping properties General %Sens Sensitivity of transducer
11	%Sens@Ref Sensitivity of transducer at reference conditions
12	%Reffreq Reference frequency (f ref)
13	%RefTemp Reference temperature (T ref)
14	%Sign Phase inversion (0° or 180°)
15	%Direction Direction, or axis, of sensitivity (x, y, or z)
16	%MapMeth Mapping Method of physical to electrical units
17	%MinPhysVal Minimum value of physical measurement/control range
18	%MaxPhysVal Maximum value of physical measurement/control range
19	%MinElecVal Minimum value of electrical signal range
20	%MaxElecVal Maximum value of electrical signal range
21	Strain/Bridge %GageType Topology and rosette orientation of gage
22	%BridgeType Type of bridge (quarter, half, or full)
23	%GageFactor Sensitivity of strain gage
24	%GageTransSens Transverse sensitivity of strain gage
25	%GageOffset Zero offset of gage circuit after installation
26	%PoissonCoef Poisson Coefficient of strain gage
27	%YoungsMod Youngs Modulus of material to which gage is attached
28	%GageArea Area of gage element
29	RTD and thermistor %RTDCoef_R0 RTD or thermistor resistance at 0 °C
30	%RTDCoef_A Coefficient A of Callendar Van-Dusen equation for RTDs
31	%RTDCoef_B Coefficient B of Callendar Van-Dusen equation for RTDs
32	%RTDCoef_C Coefficient C of Callendar Van-Dusen equation for RTDs

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

33 %SteinhartA Coefficient A of Steinhart-Hart equation for thermistors

34 %SteinhartB Coefficient B of Steinhart-Hart equation for thermistors

35 %SteinhartC Coefficient C of Steinhart-Hart equation for thermistors

36 %SelfHeating Coefficient of self-heating, intended for thermistors

37 TC %TCType Thermocouple calibration type (J, K, T, etc.)

38 %CJSource Cold-junction compensation method

39 Electrical signal properties General %ElecSigType Type of electrical signal (enumerated)

40 %RespTime Response Time

41 %ACDCCoupling Coupling of electrical signal (AC or DC)

42 %SensorImped Electrical impedance of sensor (of each element in case of bridge)

43 %DiscSigType Discrete signal type

44 %DiscSigAmpl Discrete signal voltage amplitude

45 %PulseMeasType Pulse signal measurement type (frequency, period, count, etc.)

46 %Gain Gain of preamplifier

47 %Filter Indicates selectable filter

48 %TempCoef Temperature coefficient

49 Sensitivity and mapping properties Mic/Preamp %Prepolarized Prepolarized (yes or no)

50 %RefPol Polarization voltage

51 %Rin Input resistance of amplifier

52 %Rout Output resistance of amplifier

53 %Cin Input capacitance of amplifier

54 %Cmic Microphone capacitance

55 %Cstray Microphone stray capacitance

56 %Rleakage Microphone leakage resistance

57 %MicType Microphone type

58 %MicSize Microphone size

59 %Resp_Type Frequency response type

60 %RefPress Reference pressure

61 %Equi_Vol Equivalent microphone volume

62 %Gate Gate present

63 Excitation and power %ExciteAmplNom Excitation or power-supply level, nominal

64 %ExciteAmplMin Excitation or power-supply level, minimum

65 %ExciteAmplMax Excitation or power-supply level, maximum

66 %ExciteType Type of excitation or power (DC, AC, or bipolar DC)

67 %ExciteCurrentDraw Maximum current required to power/excite transducer

68 %ExciteFreqNom Excitation signal frequency, nominal

69 %ExciteFreqMin Excitation signal frequency, minimum

70 %ExciteFreqMax Excitation signal frequency, maximum

71 %LoopSupplyMin Supply for current loop transducers, minimum

72 %LoopSupplyMax Supply for current loop transducers, maximum

73 Calibration properties Mg. %CalDate The date of the last calibration

74 %CalInitials Calibration initials

75 %CalPeriod Amount of time recommended between calibrations

76 Sensitivity and mapping properties Calibration table and curves %CalTable_Domain Indicates calibration table domain as electrical or physical

77 %CalPoint_DomainValue Domain calibration value

78 %CalPoint_RangeValue Range calibration deviation

79 %CalCurve_Domain Indicates calibration curve domain as electrical or physical

80 %CalCurve_PieceStart Start of calibration curve segment

81 %CalCurve_Power Power of domain value

82 %CalCurve_Coef Coefficient of polynomial

83 Transfer function %TF_SZ Single zero

84 %TF_SP Single pole (low-pass filter of first order)

85 %TF_KZr Complex zero

86 %TF_KZq Quality factor parameter Qz of a complex zero

87 %TF_KPr Complex pole at Fpres (F mounted resonance)

88 %TF_KPq Quality factor Qp for the complex pole (mounted quality factor)

89 %TF_HP_S Single zero at 0 and a single pole (high-pass filter)

90 %TF_SL Constant relative slope

91 %TF_SZm Single zero dependent on previous property

92 %TF_SPm Single pole dependent on previous property

93 %PhaseCorrection Phase correction at the reference condition

94 %TF_Table_Freq Frequency point value for tabular transfer function

95 %TF_Table_Ampl Amplitude point value for tabular transfer function

96 Miscellaneous properties Attached transducer %Attached_MfgrID Manufacturer ID of transducer attached to amplifier

97 %Attached_ModelNum Model number of transducer attached to amplifier

98 %Attached_VersionLetter Version letter of transducer attached to amplifier

99 %Attached_VersionNum Version number of transducer attached to amplifier

100 %Attached_SerialNum Serial number of transducer attached to amplifier

101 %System_MfgrID Manufacturer ID of system

102 %System_ModelNum Model number of system

103 %System_VersionLetter Version letter of system

104 %System_VersionNum Version number of system

105 %System_SerialNum Serial number of system

106 Sensitivity and mapping properties Miscellaneous %Stiffness Stiffness of transducer

107 %Mass_below Mass below gage

108 %Weight Weight of transducer

109 %TestGain Test gain

110 %Passive Indicates support of passive mode

111 %PollFreq The frequency with which the host shall update the FR

112 %MeasID Measurand ID

113 %Ccable Capacitance of cable

114 %CableLen Length of cable

115 %Appended_TEDS Indicates if an Appended TEDS exists

116 %Appended_TEDS_location Indicates location of the Appended TEDS if it exists

117 %EMBTPL Indicates that the next portion of the TEDS is in Embedded Template format

118 %DefaultFR Defines the default setting of the FR. Cannot coexist with subproperty Default.

119 %XML XML format

120 %MDEF Prefix for manufacturer-defined parameters

121 %TDL_CHKSUM User template validation checksum

122 %user Freeform TEDS format

123 Grouping properties %PhysicalParameterType Describes the parameter type

124 %MemberIndex Indicate the order in which XdcrChannels are grouped within a PublicXdcr

E: Inbetriebnahme des Gerätes GSV-6 (Beispiel)

Voraussetzungen zur Inbetriebnahme sind von GSV-6CPU sind:

a) Versorgungsspannung 3,7V …5.0V an Pin 14VCC_IN

b) GND 0V an Pin 15;

c) Pin13 „Supply Warnung“ mit Pin14 “VCC_IN” verbunden

d) UART2_RX und UART2_TX (3,3V TTL Pegel) ggfs mit Pegelumsetzer beschalten

Die folgende Sequenz protokolliert das Verhalten ab dem Einschalten mit Default-Werten. Die Verwendung der Kommandos ist hier beispielhaft vorgeschlagen.

Das Gerät sendet zunächst die Messwerte aller 6 Kanäle mit der konfigurierten Datenfrequenz[19], sofern der Messdatenframe für 6 Kanäle konfiguriert wurde. Die vom Gerät gesendeten Daten sind rot dargestellt. Die Kommandos, die zum Gerät gesendet werden, sind blau dargestellt.

```
Nach dem Einschalten werden beim GSV-6 z.B. 6 Kanäle übertragen mit 10Hz:
AA 15 B0 3A 49 9B 2C BF 86 66 66 BF 5C D4 2D BF 4E E3 26 B9 A8 01 50 BF 86 66 66 85
AA 15 B0 BC 40 27 E6 BF 86 66 66 BE DC 40 93 BE 50 BA A2 BC 8C B4 4C BF 86 66 66 85
AA 15 B0 BC EA 28 3A BF 86 66 66 3E 1A 85 C4 3F 1B 51 A7 BD 23 8A E0 BF 86 66 66 85
AA 15 B0 BD 30 24 93 BF 86 66 66 3F 23 BF 6C 3F 86 66 66 BD 72 4B 7E BF 86 66 66 85
AA 15 B0 BD 58 4E 7D BF 86 66 66 3F 75 9F 22 3F 86 66 66 BD 93 44 59 BF 86 66 66 85
AA 15 B0 BD 6E 5B 76 BF 86 66 66 3F 86 66 66 3F 86 66 66 BD A1 4F A9 BF 86 66 66 85
AA 15 B0 BD 78 11 EF BF 86 66 66 3F 86 66 66 3F 86 66 66 BD A6 F4 81 BF 86 66 66 85
```

Stop Transmission

```
Durch die Anwendung des Kommandos StopTransmission wird die Datenübertragung angehalten:
AA 90 23 85
AA 50 00 85
```

GetValue

```
Durch die Anwendung des Kommandos GetValue wird ein Messdatenframe angefordert
AA 90 3B 85
AA 15 B0 BD FA 09 F3 BF 86 66 66 3F 86 66 66 3F 86 66 66 BE 1E E2 0A BF 86 66 66 85
```

19 Dies tut es nicht, wenn die ständige Messdatenübertragung deaktiviert wurde durch Cmd. SetTXmode

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

F: Defaulteinstellungen

Im folgender Tabelle sind die Herstellereinstellungen genannt, die nach Ausführen des Kommandos Load Config (0x09) mit Parameter =1 gelten. Bei Sondervarianten können diese abweichen.

Parameter	Defaultwert GSV-8	Defaultwert GSV-6	Rd,Wr Cmd (hex)
Eingangsempfindlichkeit	3,5 mV/V	2 mV/V	GSV-8: A2, A3 GSV-6: 6, 7
Messbereich / Typ	Brückeneingang 3,5 mV/V	Brückeneingang 4 mV/V	A2, A3
Messdatenrate	10 Frames/s	10 Frames/s	8A, 8B
Offset (Nullpunkt)	Kalibrierwert für 0 mV/V	Kalibrierwert für 0 mV/V	2, 3
User-Scale	3,5	2	14, 15
User-Offset	0	0	9A, 9B
Einheit	mV/V (Enum: 0)	mV/V (Enum: 0)	0F, 10
Einheit-Sondertext	"" [keine=Leertext]	"unknown0", "unknown1"	11, 12
Perm. Messdaten-übertragung?	Ja	Ab v.3.11: Unverändert Werksseinstellung: Ja	80, 81
Bezugswert f. SCALE Key Skalierfunktion	---	Unverändert Werksseinstellung: 100%	---
Sync. Master/Slave?	Nein	Nein	5B, 5C
TX Max / Min?	Nein, d.h. aktueller Wert	Nein, d.h. aktueller Wert	GSV-8: 80, 81 GSV-6: 26, 27
Digitalfilter	Aus	Aus	51, 52
6-Achsensensor aktiv	Nein	Nein	26, 27
6-Achsensensor: Aktive Datensatz-Nr.	0	0	54, 55
Analogfilter anhand Fg automatisch setzen	Ja	---	26, 27 (nur GSV-8)
Noisecut aktiv	Nein	---	26, 27 (nur GSV-8)
Noisecut Schwelle	5% v. FS, d.h. 0,175 mV/V	---	94, 95 (nur GSV-8)
Maxwert ist Betrags-Maximalwert?	Nein	---	26, 27 (nur GSV-8)
Schreibschutz an?	Nein	Nein	92
TEDS verwenden	Ja	Ja	26, 27
Einheit anhand TEDS	Ja	Ja (nicht konfigurierbar)	26, 27 (nur GSV-8)
InTyp anhand TEDS	Nein	Ja (nicht konfigurierbar)	26, 27 (nur GSV-8)
AoutScale anhand TEDS	Ja	Ja (nicht konfigurierbar)	26, 27 (nur GSV-8)
Nullpukt anhand TEDS	Nein	Nein[20]	26, 27
Menüausgabe 0-20mA,	---	Unverändert	26, 27

20 Ausnahme: Nur bei Firmware-version 3.20 ist "Nullpunkt anhand TEDS" per Default AN

Parameter		Defaultwert GSV-8	Defaultwert GSV-6	Rd,Wr Cmd (hex)
wenn Analogausgang Strom			Werksseinstellung: AUS	
Autozero aktiv		Nein	---	26, 27 (nur GSV-8)
Autozero Schwelle		5% v. FS, d.h. 0,175 mV/V	---	96, 97 (nur GSV-8)
Autozero Periode		5 Sek.	---	96, 97 (nur GSV-8)
Analogausgang: Typ		±10 V	±10 V	0D, 0E
Analogausgang: Offset		0	0	4, 5
Analogausgang: Scaling		1	2 (wirkt auch auf Digitalausgang)[21]	6, 7
Digital I/O: Typ		GP-Input	No. 1-3: Schwellwert m. Hysterese 4: TRIG: SD-File Log. Trigger 5: TARA: TaraAll: Alles Nullsetzen	5B, 5C
Digital I/O: Flags		Nicht-invertiert, unipolar	Nicht-invertiert, unipolar	5B, 5C
Digital I/O: Defaultpegel		0 (Low)	0 (Low)	61, 62
Schwellwert, oberer		105% v. FS, d.h. 3,675 mV/V, d.h. inaktiv	SW1: 90% v.FS, d.h. 1,8 mV/V SW2: 50% v.FS, d.h. 1 mV/V SW3: 75% v.FS, d.h. 1,5 mV/V	5F, 60
Schwellwert, unterer		-105% v. FS, d.h. -3,675 mV/V, d.h. inaktiv	SW1: 88% v.FS, d.h. 1,76 mV/V SW2: 48% v.FS, d.h. 0,96 mV/V SW3: 73% v.FS, d.h. 1,46 mV/V	5F, 60
Digitalfilter verwendet:		Nein	Nein	51, 52
Zähler/Frequenzmesser-Modus AN/AUS		Unverändert Werksseinstellung: AUS	Unverändert Werksseinstellung: AUS	69, 6A
Zähler	Modeflags	0b0000.11001uu1	0b0000.11001uu1 (u= unverändert)	69, 6A (0)
	Torzeit	0	0	69, 6A (1)
	Startwert	0	0	69, 6A (2)
Tasten / Digitaleingang Entprellzeit		DIO1-16: 40 ms Sampling-Periode Tasten: 120-159 ms (beides nicht veränderbar)	TARA: 1 Sek. SCALE: 1 Sek. TRIG: 10 ms	86, 87 (nur GSV-6)
Logging auf SD-Karte		---	AUS	6D, 6E (nur GSV-6 mit SD-Slot)
Logger	Modeflags	---	0b000000u	6D, 6E (0)
	Alarm-Mode	---	0b0	6D, 6E (1)
	Dir./Form.Flags	---	0b1011.00000001	6D, 6E (2)
	Messwertzeilen pro Datei	---	30000	6D, 6E (3)
	Dezimation	---	1	6D, 6E (4)
	Multiplikator	---	1	6D, 6E (5)
Prohibit Set Zero		0x0000 (=alle erlaubt)	---	32, 33
CRC-16 in Messframes		Ohne CRC-16	Ohne CRC-16	01, 80, 81 (0)

21 Bei GSV-1L/K: Kalibrierter Wert, d.h. kann leicht von 2 abweichen

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

Folgende Parameter verändern sich duch Aufruf von Load Config **nicht**:
Kommunikations-Einstellungen: Bitrate, CAN-IDs, Kanalanzahl im Messdatenframe, Messdatentyp. Ferner Digitalfilterkoeffizienten, Sechsachsen-Sensordaten, Benutzer-Passwort sowie Flags, die das Vorhandensein von Hardware Komponenten anzeigen.

Parameter		Defaultwert GSV-8	Defaultwert GSV-6	Rd,Wr Cmd (hex)
CAN-IDs	Cmd-Request	-	0x100	8C, 8D (0)
	Cmd-Response	-	0x101	8C, 8D (1)
	Messwerte	-	0x101	8C, 8D (2)
	CANopen NodeID	0x40 (GSV-8 CANopen)	-	8C, 8D (6)
CAN Bitrate		500 kBits/s (GSV-8 CANopen)	1 MBits/s	8C, 8D (4)
UART-Bitrate		115200	230400	7B, 7C
Kanalanzahl im Frame		8	6[22]	49, 4A (0)
Messdatentyp		Float	Float	80, 81 (1)
Benutzer-Passwort		"Beln"	"USC1"	58 (nur GSV-8, Wr)

22 Bei GSV-6: ab Firmware Version 3.11. Vorher wurde auf EIN Kanal mit Datentyp= FLOAT gesetzt. Je nach Modellausführung ist u.U. nur der erste Kanal kalibriert. Bei Analogmodellen: Nur Kanal 1 (Kanalanzahl=1).

G: Funktionen in C zur Prüfsummenberechnung

```
const uint16_t u16CrcTable [256] = {
        0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
        0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
        0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
        0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
        0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
        0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
        0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
        0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
        0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
        0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
        0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
        0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
        0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
        0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
        0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
        0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
        0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
        0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
        0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
        0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
        0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
        0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
        0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
        0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
        0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
        0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
        0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
        0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
        0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
        0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
        0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
        0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040};

/*****************************************************************************
* Function Name  : u16MakeCRC16
* Description    : Berechnet CRC16 Checksumme für Modbus
* Input          : *pu8data : Pointer auf die zu berechnenden Daten
*                : u16length : Länge der Daten in Bytes
* Output         : None
* Return         : CRC16
*****************************************************************************/

uint16_t u16MakeCRC16(const uint8_t *pu8data, uint16_t u16length)
{
        uint16_t count;
        uint8_t u8erg1;
        uint16_t u16erg2;
        union {
           uint16_t u16Buffer;
           uint8_t u8Buffer[2];
        } uCrc;
        uCrc.u16Buffer = 0xFFFF;

        for (count = 0; count < u16length; ++count)
        {
                u8erg1 =        pu8data[count] ^ uCrc.u8Buffer[0] ;
                u16erg2 = u16CrcTable[u8erg1];
                uCrc.u16Buffer = uCrc.u8Buffer[1] ^ u16erg2;
        }
        return uCrc.u16Buffer;
}
```


ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de

```
const uint8_t u8CRC8Table[256]= {
    0x00, 0x07, 0x0E, 0x09, 0x1C, 0x1B, 0x12, 0x15, 0x38, 0x3F, 0x36, 0x31, 0x24, 0x23, 0x2A, 0x2D,
    0x70, 0x77, 0x7E, 0x79, 0x6C, 0x6B, 0x62, 0x65, 0x48, 0x4F, 0x46, 0x41, 0x54, 0x53, 0x5A, 0x5D,
    0xE0, 0xE7, 0xEE, 0xE9, 0xFC, 0xFB, 0xF2, 0xF5, 0xD8, 0xDF, 0xD6, 0xD1, 0xC4, 0xC3, 0xCA, 0xCD,
    0x90, 0x97, 0x9E, 0x99, 0x8C, 0x8B, 0x82, 0x85, 0xA8, 0xAF, 0xA6, 0xA1, 0xB4, 0xB3, 0xBA, 0xBD,
    0xC7, 0xC0, 0xC9, 0xCE, 0xDB, 0xDC, 0xD5, 0xD2, 0xFF, 0xF8, 0xF1, 0xF6, 0xE3, 0xE4, 0xED, 0xEA,
    0xB7, 0xB0, 0xB9, 0xBE, 0xAB, 0xAC, 0xA5, 0xA2, 0x8F, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9D, 0x9A,
    0x27, 0x20, 0x29, 0x2E, 0x3B, 0x3C, 0x35, 0x32, 0x1F, 0x18, 0x11, 0x16, 0x03, 0x04, 0x0D, 0x0A,
    0x57, 0x50, 0x59, 0x5E, 0x4B, 0x4C, 0x45, 0x42, 0x6F, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7D, 0x7A,
    0x89, 0x8E, 0x87, 0x80, 0x95, 0x92, 0x9B, 0x9C, 0xB1, 0xB6, 0xBF, 0xB8, 0xAD, 0xAA, 0xA3, 0xA4,
    0xF9, 0xFE, 0xF7, 0xF0, 0xE5, 0xE2, 0xEB, 0xEC, 0xC1, 0xC6, 0xCF, 0xC8, 0xDD, 0xDA, 0xD3, 0xD4,
    0x69, 0x6E, 0x67, 0x60, 0x75, 0x72, 0x7B, 0x7C, 0x51, 0x56, 0x5F, 0x58, 0x4D, 0x4A, 0x43, 0x44,
    0x19, 0x1E, 0x17, 0x10, 0x05, 0x02, 0x0B, 0x0C, 0x21, 0x26, 0x2F, 0x28, 0x3D, 0x3A, 0x33, 0x34,
    0x4E, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5C, 0x5B, 0x76, 0x71, 0x78, 0x7F, 0x6A, 0x6D, 0x64, 0x63,
    0x3E, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2C, 0x2B, 0x06, 0x01, 0x08, 0x0F, 0x1A, 0x1D, 0x14, 0x13,
    0xAE, 0xA9, 0xA0, 0xA7, 0xB2, 0xB5, 0xBC, 0xBB, 0x96, 0x91, 0x98, 0x9F, 0x8A, 0x8D, 0x84, 0x83,
    0xDE, 0xD9, 0xD0, 0xD7, 0xC2, 0xC5, 0xCC, 0xCB, 0xE6, 0xE1, 0xE8, 0xEF, 0xFA, 0xFD, 0xF4, 0xF3 };

/*****************************************************************************
* Function Name  : makeCRC8
* Description    : Berechnet CRC8 Checksumme
* Input          : * u8Tele : Pointer auf die zu berechnenden Daten
*                : len : Länge der Daten in Bytes
* Output         : None
* Return         : CRC8
*****************************************************************************/

uint8_t makeCRC8(const uint8_t* u8Tele, uint16_t len)
{
        uint8_t u8Result = 0;
        uint16_t i;
        for (i = 0; i<len; i++)
        {
                u8Result = u8CRC8Table[u8Result ^ u8Tele[i]];
        }
        return u8Result;
}
```

Made in Germany

Änderungen vorbehalten.
Alle Angaben beschreiben unsere Produkte in allgemeiner Form.
Sie stellen keine Eigenschaftszusicherung im Sinne des §459 Abs. 2, BGB, dar
und begründen keine Haftung.

ME-Meßsysteme GmbH
Eduard-Maurer Str. 9
16761 Hennigsdorf
Tel.: +49 3302 89824 60
Fax: +49 3302 89824 69
Mail: info@me-systeme.de
Web: www.me-systeme.de