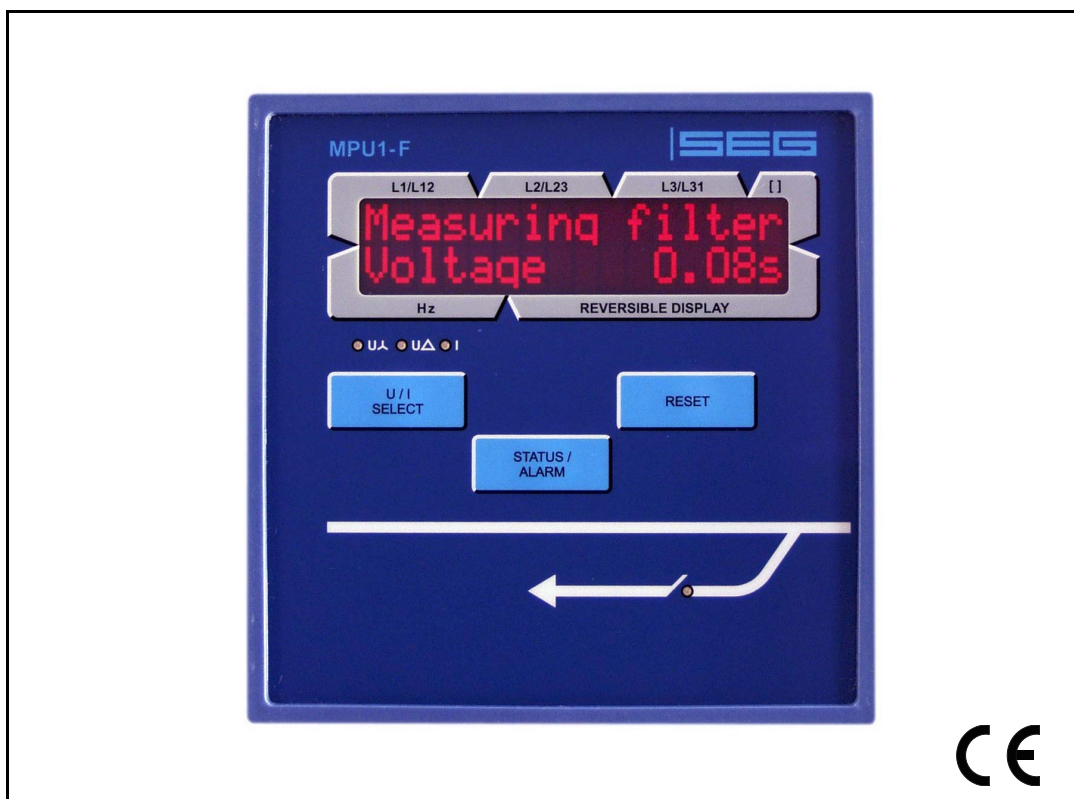


MPU1-F Multimeßwertumsetzer Protokollbeschreibung



1	Allgemeines	3
2	Rahmendaten zum CAN-Protokoll	4
2.1	Aufbau einer Visualisierungsmessung und des Sendetelegramms	4
2.2	Aufbau der Parametrierungsmessung	5
2.2.1	Begriffe.....	5
2.2.2	Leseanforderung des Masters	6
2.2.3	Schreibanforderung des Masters	7
2.2.4	Antwort des Slave (MPU1-F) auf eine Leseanforderung des Masters	9
2.2.5	Antwort des Slave (MPU1-F) auf eine Schreibanforderung des Masters	10
3	Sendetelegramm des MPU1-F	12
4	Parametrier-ID's des MPU1-F	13
4.1	Begriffe.....	13
4.1.1	Art	13
4.1.2	ID	13
4.1.3	Wertebereich	13
4.1.4	Berechnungsformel.....	13
4.2	Parametrierung über den CAN-Bus für das MPU1-F	13
4.2.1	Passwort für Codestufe Level 2	13
4.2.2	Spannungswandler sekundär	13
4.2.3	Spannungswandler primär.....	14
4.2.4	Stromwandler primär	14
4.2.5	Messung Netzart.....	14
4.2.6	Impuls Ausgang Pulsdauer	14
4.2.7	Impuls Ausgang Größe	14
4.2.8	Impuls Ausgang Logik	15
4.2.9	Impuls Ausgang Pulse pro kWh bzw. Pulse pro kvarh.....	15
4.2.10	Analog Ausgang 1 AUS/0..20mA/4..20mA/-20mA+20mA	15
4.2.11	Analog Ausgang Größe.....	16
4.2.12	Analog Ausgang unterer Wert.....	16
4.2.13	Analog Ausgang oberer Wert	16
4.2.14	Anzeige kWh aktiv JA/NEIN	17
4.2.15	Anzeige kvarh aktiv JA/NEIN	17
4.2.16	Reset kWh/kvarh	17
4.2.17	Highword Zähler pos. kWh stellen.....	17
4.2.18	Lowword Zähler pos. kWh stellen.....	17
4.2.19	Zähler pos. Wh stellen	18
4.2.20	Highword Zähler pos. kvarh stellen	18
4.2.21	Lowword Zähler pos. kvarh stellen	18
4.2.22	Zähler pos. varh stellen.....	18
4.2.23	Highword Zähler neg. kWh stellen	18
4.2.24	Lowword Zähler neg. kWh stellen.....	18
4.2.25	Zähler neg. Wh stellen	19
4.2.26	Highword Zähler neg. kvarh stellen.....	19
4.2.27	Lowword Zähler pos. kvarh stellen	19
4.2.28	Zähler pos. varh stellen.....	19
4.2.29	Zykluszeit Display	19
4.2.30	Messwertfilter Spannung	20
4.2.31	Messwertfilter Strom	20
4.2.32	Messwertfilter Leistung	20
4.2.33	Messwertfilter Frequenz.....	21

1 Allgemeines

Dieses Dokument beschreibt das CAN-Bus-Protokoll, das zur Steuerung, Visualisierung und Parametrierung genutzt wird. Die Beschreibung erfolgt analog zu den Softwareoptionen des Gerätes und gliedert sich in der Beschreibung des Parametrierteils wie die Geräteversionen selbst in einen High-, Medium- und Lowteil. Mit Hilfe der Protokollbeschreibung ist es möglich, beliebige Applikationen auf den Bus aufzusetzen, die zur Parametrierung, Steuerung oder Visualisierung genutzt werden können.



ACHTUNG !

Die Offenlegung dieses Protokolles geschieht auf Kundenwunsch. Es darf nur auf den angegebenen Adressen innerhalb der vorgegebenen Wertebereiche parametrieren werden, da ansonsten Fehlfunktionen im Gerät auftreten können. Für Schäden irgendwelcher Art, die durch eine solche Fehlparametrierung entstehen, wird keinerlei Haftung übernommen.

2 Rahmendaten zum CAN-Protokoll

125 kBaud nach CIA
11 Bit Identifier
8 Byte Messagelänge
ca. 100ms Abstand zwischen zwei Messages

2.1 Aufbau einer Visualisierungsmessung und des Sendetelegramms

Eine Visualisierungsmessung wird vom **MPU1-F** an eine andere Einheit z.B. SPS oder PC mit CAN-Modul gesendet. Die Visualisierungsmessungen beinhalten Messdaten, Fehlerzustände und Betriebszustände des **MPU1-F**. Die CAN-ID, auf der das **MPU1-F** sendet berechnet sich wie folgt:

$$\text{CAN-ID} = \text{H'320} + \text{Gerätenummer}$$

(Die Gerätenummer ist ein am **MPU1-F** einstellbarer Parameter, der unmittelbar die CAN-ID, auf der das Gerät seine Visualisierungsbotschaften sendet, beeinflusst.)

Eine Visualisierungsmessung, die auf der CAN-ID eines bestimmten **MPU1-F** empfangen wird besteht aus 8 Byte und ist wie folgt aufgebaut:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
H'DD	Muxnummer	Datenwort 1 High-Byte	Datenwort 1 Low Byte	Datenwort 2 High-Byte	Datenwort 2 Low Byte	Datenwort 3 High-Byte	Datenwort 3 Low Byte

Bei einer Visualisierungsbotschaft steht im Byte 0 immer der hexadezimale Wert DD, dieser kennzeichnet die Sendung als Visualisierungsbotschaft. Da das Sendetelegramm des MPU mehr als drei Datenworte beinhaltet, wird auf Byte 1 zusätzlich eine Muxnummer beginnend bei 0 gesendet, so ist es theoretisch möglich über eine CAN-ID ($256 * 3 = 768$) Datenworte zu senden. Das gesamte Telegramm baut sich dann folgendermaßen auf:

Zeile 1: Muxnummer 0, Datenwort 1
Zeile 2: Muxnummer 0, Datenwort 2
Zeile 3: Muxnummer 0, Datenwort 3
Zeile 4: Muxnummer 1, Datenwort 1
Zeile 5: Muxnummer 1, Datenwort 2
Zeile 6: Muxnummer 1, Datenwort 3
.
.
Zeile (n-2): Muxnummer (n-3), Datenwort 1
Zeile (n-1): Muxnummer (n-2), Datenwort 2
Zeile (n-0): Muxnummer (n-1), Datenwort 3

n hängt von der Gesamtlänge des geräteeigenen Telegramms ab und kann nicht größer als H'FF sein. Hier ist nur das Prinzip der Visualisierung beschrieben, das reale Sendetelegramm des **MPU1-F** befindet sich in Kapitel 4.

2.2 Aufbau der Parametriermessages

Hier wird nur das Prinzip der Parametrierung über den CAN-Bus beschrieben. Die realen Parametrier-ID's (Adressen), die Wertebereiche und Berechnungsformeln befinden sich in Kapitel 5.

2.2.1 Begriffe

a.) CAN-ID

Wird im Folgenden von CAN-ID gesprochen, sind damit die Adressen der Empfangs- und Sendeboxen der CAN-Hardware gemeint. **Diese sind nicht mit den Parametrier-ID's zu verwechseln.**

b.) Parametrier-ID

Wird im Folgenden von Parametrier-ID gesprochen, oder findet sich innerhalb einer Message der Begriff ID oder Parametrier-ID, so sind damit die Adressen der einzelnen Parameter im *MPU1-F* gemeint. **Diese sind nicht mit den CAN-ID's zu verwechseln.**

c.) Master

Wenn in den folgenden Beschreibungen von einem Master oder Mastergerät die Rede ist, ist damit eine SPS, ein PC oder ein anderes Leitbusgerät gemeint, die an das *MPU1-F* Schreib- oder Leseanforderungen über den CAN-Bus senden.

d.) Slave

Wenn in den folgenden Beschreibungen von einem Slave die Rede ist, ist damit das *MPU1-F* gemeint, das auf Schreib- oder Leseanforderungen vom Master antwortet.

e.) Leseanforderung

Ein Mastergerät (SPS, PC o.ä.) möchte einen bestimmten Parameter aus dem *MPU1-F* auslesen.

f.) Schreibenanforderung

Ein Mastergerät (SPS, PC o.ä.) möchte einen bestimmten Wert im *MPU1-F* parametrieren bzw. verändern.

2.2.2 Leseanforderung des Masters

Hier wird die zu generierende CAN-Botschaft beschrieben, die ein Master an den Slave senden muss, um eine Parametrier-ID (Parameteradresse) auszulesen. Leseanforderungen des Masters werden für alle am Bus vorhandenen Geräte auf der gleichen CAN-ID gesendet, diese lautet:

CAN-ID = H'33F

Die gleiche ID wird auch bei Schreibenanforderungen benutzt. Die Adressierung an das jeweilige Gerät und die Unterscheidung, ob es sich um eine Schreib- oder Leseanforderung handelt sind in der CAN-Message selbst enthalten. So empfängt jedes Gerät auf dem Bus zunächst die Anforderung und überprüft dann, ob die eingehende Message an es adressiert war. Das acht Byte lange Sendetelegramm für eine Leseanforderung sieht wie folgt aus:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
H'FF	Slave-Nr.	ID (Adresse) High-Byte	ID (Adresse) Low Byte	0	0	Prüfsumme High-Byte	Prüfsumme Low Byte

a.) Byte 0

In Byte 0 steht bei einer Leseanforderung des Masters immer der hexadezimale Wert FF, daran erkennt der Slave, dass er eine Leseanforderung beantworten soll.

b.) Byte 1

In Byte 1 steht die Slavenummer, die mit der Gerätenummer des *MPU1-F*, der angesprochen werden soll identisch sein muss. Das *MPU1-F* vergleicht die Slavenummer mit der parametrisierten Gerätenummer und kann so entscheiden, ob die eingegangene Leseanforderung für es bestimmt ist.

c.) Byte 2

Die Parametrier-ID (Adresse), die ausgelesen werden soll beansprucht ein 16Bit-Wort, in Byte 2 steht das Highbyte dieses Wortes. Die Parametrier-ID's (Adressen) für die einzelnen Parameter des *MPU1-F* finden sich in Kapitel 5 wieder.

d.) Byte 3

Die Parametrier-ID (Adresse), die ausgelesen werden soll beansprucht ein 16Bit-Wort, in Byte 3 steht das Lowbyte dieses Wortes. Die Parametrier-ID's (Adressen) für die einzelnen Parameter des *MPU1-F* finden sich in Kapitel 5 wieder.

e.) Byte 4

Bei einer Leseanforderung wird der Wert 0 gesendet.

f.) Byte 5

Bei einer Leseanforderung wird der Wert 0 gesendet.

g.) Byte 6

Die Prüfsumme beansprucht ein 16Bit-Wort, dieses wird durch die dargestellte logische Verknüpfung vom Master erzeugt:

$$\text{Prüfsumme} = \text{Wort1}(\text{Byte0}, \text{Byte1}) \text{ XOR Wort2}(\text{Byte2}, \text{Byte3}) \text{ XOR Wort3}(\text{Byte4}, \text{Byte5})$$

Da diese Verknüpfung Bitweise erzeugt wird, kann Byte6 wie folgt vom Master für die Sendung berechnet werden: $\text{Byte6} = \text{Byte0} \text{ XOR Byte2 XOR Byte4}$

h.) Byte 7

Die Prüfsumme beansprucht ein 16Bit-Wort, dieses wird durch die dargestellte logische Verknüpfung vom Master erzeugt:

$$\text{Prüfsumme} = \text{Wort1}(\text{Byte0}, \text{Byte1}) \text{ XOR Wort2}(\text{Byte2}, \text{Byte3}) \text{ XOR Wort3}(\text{Byte4}, \text{Byte5})$$

Da diese Verknüpfung Bitweise erzeugt wird, kann Byte7 wie folgt vom Master für die Sendung berechnet werden: $\text{Byte7} = \text{Byte1} \text{ XOR Byte3 XOR Byte5}$

2.2.3 Schreibenanforderung des Masters

Hier wird die zu generierende CAN-Botschaft beschrieben, die ein Master an den Slave senden muss, um eine Parametrier-ID (Parameteradresse) zu beschreiben. Schreibenanforderungen des Masters werden für alle am Bus vorhandenen Geräte auf der gleichen CAN-ID gesendet, diese lautet:

$$\text{CAN-ID} = \text{H}'33\text{F}$$

Die gleiche ID wird auch bei Leseanforderungen benutzt. Die Adressierung an das jeweilige Gerät und die Unterscheidung, ob es sich um eine Schreib- oder Leseanforderung handelt sind in der CAN-Message selbst enthalten. So empfängt jedes Gerät auf dem Bus zunächst die Anforderung und überprüft dann, ob die eingehende Message an es adressiert war. Das acht Byte lange Sendetelegramm für eine Schreibenanforderung sieht wie folgt aus:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
H'EE	Slave-Nr.	ID (Adresse) High-Byte	ID (Adresse) Low Byte	Datenwort High Byte	Datenwort Low Byte	Prüfsumme High-Byte	Prüfsumme Low Byte

a.) Byte 0

In Byte 0 steht bei einer Schreibenanforderung des Masters immer der hexadezimale Wert EE, daran erkennt der Slave, dass er eine Schreibenanforderung ausführen soll.

b.) Byte 1

In Byte 1 steht die Slavenummer, die mit der Gerätenummer des **MPU1-F**, der angesprochen werden soll identisch sein muss. Das **MPU1-F** vergleicht die Slavenummer mit der parametrisierten Gerätenummer und kann so entscheiden, ob die eingegangene Schreibenanforderung für es bestimmt ist.

c.) Byte 2

Die Parametrier-ID (Adresse), die beschrieben werden soll beansprucht ein 16Bit-Wort, in Byte 2 steht das Highbyte dieses Wortes. Die Parametrier-ID's (Adressen) für die einzelnen Parameter des *MPU1-F* finden sich in Kapitel 5 wieder.

d.) Byte 3

Die Parametrier-ID (Adresse), die beschrieben werden soll beansprucht ein 16Bit-Wort, in Byte 3 steht das Lowbyte dieses Wortes. Die Parametrier-ID's (Adressen) für die einzelnen Parameter des *MPU1-F* finden sich in Kapitel 5 wieder.

e.) Byte 4

Der Wert, der auf eine Parametrier-ID geschrieben werden soll beansprucht ein 16Bit-Wort, in Byte 4 steht das Highbyte dieses Wortes. Die Berechnungsformeln für die einzelnen Parameter finden sich in Kapitel 5 wieder.

f.) Byte 5

Der Wert, der auf eine Parametrier-ID geschrieben werden soll beansprucht ein 16Bit-Wort, in Byte 5 steht das Lowbyte dieses Wortes. Die Berechnungsformeln für die einzelnen Parameter finden sich in Kapitel 5 wieder.

g.) Byte 6

Die Prüfsumme beansprucht ein 16Bit-Wort, dieses wird durch die dargestellte logische Verknüpfung vom Master erzeugt:

$$\text{Prüfsumme} = \text{Wort1}(\text{Byte0}, \text{Byte1}) \text{ XOR Wort2}(\text{Byte2}, \text{Byte3}) \text{ XOR Wort3}(\text{Byte4}, \text{Byte5})$$

Da diese Verknüpfung Bitweise erzeugt wird, kann Byte6 wie folgt vom Master für die Sendung berechnet werden: $\text{Byte6} = \text{Byte0} \text{ XOR Byte2} \text{ XOR Byte4}$

h.) Byte 7

Die Prüfsumme beansprucht ein 16Bit-Wort, dieses wird durch die dargestellte logische Verknüpfung vom Master erzeugt:

$$\text{Prüfsumme} = \text{Wort1}(\text{Byte0}, \text{Byte1}) \text{ XOR Wort2}(\text{Byte2}, \text{Byte3}) \text{ XOR Wort3}(\text{Byte4}, \text{Byte5})$$

Da diese Verknüpfung Bitweise erzeugt wird, kann Byte7 wie folgt vom Master für die Sendung berechnet werden: $\text{Byte7} = \text{Byte1} \text{ XOR Byte3} \text{ XOR Byte5}$

2.2.4 Antwort des Slave (MPU1-F) auf eine Leseanforderung des Masters

Die 8 Byte lange Antwort auf die Leseanforderung eines Masters erfolgt auf der Visualisierungs-CAN-ID. Die CAN-ID, auf der das **MPU1-F** seine Antwort sendet berechnet sich wie folgt:

$CAN-ID = H'320 + \text{Gerätenummer}$ (Die Gerätenummer ist ein am **MPU1-F** einstellbarer Parameter, der unmittelbar die CAN-ID, auf der das Gerät seine Antworten sendet, beeinflusst.)

Eine Antwortmessage, die auf der CAN-ID eines bestimmten **MPU1-F** gesendet wird ist wie folgt aufgebaut:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
H'FF	Slave-Nr.	ID (Adresse) High-Byte	ID (Adresse) Low Byte	Datenwort High Byte	Datenwort Low Byte	Prüfsumme High-Byte	Prüfsumme Low Byte

a.) Byte 0

In Byte 0 steht bei einer Antwort des Slave auf eine Leseanforderung des Masters immer der hexadezimale Wert FF, daran erkennt der Master, dass es sich um eine Antwort auf eine Leseanforderung handelt.

b.) Byte 1

In Byte 1 steht die Slavenummer, die mit der Gerätenummer des **MPU1-F**, das eine Antwort sendet identisch ist.

c.) Byte 2

Die Parametrier-ID (Adresse), die vom Slave gesendet wird beansprucht ein 16Bit-Wort, in Byte 2 steht das Highbyte dieses Wortes. Die Parametrier-ID's (Adressen) für die einzelnen Parameter des **MPU1-F** finden sich in Kapitel 5 wieder.

d.) Byte 3

Die Parametrier-ID (Adresse), die vom Slave gesendet wird beansprucht ein 16Bit-Wort, in Byte 3 steht das Lowbyte dieses Wortes. Die Parametrier-ID's (Adressen) für die einzelnen Parameter des **MPU1-F** finden sich in Kapitel 5 wieder.

e.) Byte 4

Der zur Parametrier-ID (Adresse) zugehörige Wert, der vom Slave gesendet wird beansprucht ein 16Bit-Wort, in Byte 4 steht das Highbyte dieses Wortes. Die Berechnungsformeln für die einzelnen Parameter finden sich in Kapitel 5 wieder.

f.) Byte 5

Der zur Parametrier-ID (Adresse) zugehörige Wert, der vom Slave gesendet wird beansprucht ein 16Bit-Wort, in Byte 5 steht das Lowbyte dieses Wortes. Die Berechnungsformeln für die einzelnen Parameter finden sich in Kapitel 5 wieder.

g.) Byte 6

Die Prüfsumme beansprucht ein 16Bit-Wort, diese muss vom Master überprüft werden und wird durch die dargestellte logische Verknüpfung erzeugt:

$$\text{Prüfsumme} = \text{Wort1}(\text{Byte0}, \text{Byte1}) \text{ XOR Wort2}(\text{Byte2}, \text{Byte3}) \text{ XOR Wort3}(\text{Byte4}, \text{Byte5})$$

Da diese Verknüpfung Bitweise erzeugt wird, kann Byte6 wie folgt vom Master ausgewertet werden: $\text{Byte6} = \text{Byte0} \text{ XOR Byte2 XOR Byte4}$

h.) Byte 7

Die Prüfsumme beansprucht ein 16Bit-Wort, diese muss vom Master überprüft werden und wird durch die dargestellte logische Verknüpfung erzeugt:

$$\text{Prüfsumme} = \text{Wort1}(\text{Byte0}, \text{Byte1}) \text{ XOR Wort2}(\text{Byte2}, \text{Byte3}) \text{ XOR Wort3}(\text{Byte4}, \text{Byte5})$$

Da diese Verknüpfung Bitweise erzeugt wird, kann Byte7 wie folgt vom Master ausgewertet werden: $\text{Byte7} = \text{Byte1 XOR Byte3 XOR Byte5}$

2.2.5 Antwort des Slave (MPU1-F) auf eine Schreibanforderung des Masters

Die 8 Byte lange Antwort auf die Schreibanforderung eines Masters erfolgt auf der Visualisierungs-CAN-ID. Die CAN-ID, auf der das **MPU1-F** seine Antwort sendet berechnet sich wie folgt:

$\text{CAN-ID} = \text{H}'320 + \text{Gerätenummer}$ (Die Gerätenummer ist ein am **MPU1-F** einstellbarer Parameter, der unmittelbar die CAN-ID, auf der das Gerät seine Antworten sendet, beeinflusst.)

Eine Antwortmessage, die auf der CAN-ID eines bestimmten **MPU1-F** gesendet wird ist wie folgt aufgebaut:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
H'EE	Slave-Nr.	ID (Adresse) High-Byte	ID (Adresse) Low Byte	Datenwort High Byte	Datenwort Low Byte	Prüfsumme High-Byte	Prüfsumme Low Byte

a.) Byte 0

In Byte 0 steht bei einer Antwort des Slave auf eine Schreibanforderung des Masters immer der hexadezimale Wert EE, daran erkennt der Master, dass es sich um eine Antwort auf eine Schreibanforderung handelt.

b.) Byte 1

In Byte 1 steht die Slavenummer, die mit der Gerätenummer des **MPU1-F**, das eine Antwort sendet identisch ist.

c.) Byte 2

Die Parametrier-ID (Adresse), die vom Slave gesendet wird beansprucht ein 16Bit-Wort, in Byte 2 steht das Highbyte dieses Wortes. Die Parametrier-ID's (Adressen) für die einzelnen Parameter des **MPU1-F** finden sich in Kapitel 5 wieder.

d.) Byte 3

Die Parametrier-ID (Adresse), die vom Slave gesendet wird beansprucht ein 16Bit-Wort, in Byte 3 steht das Lowbyte dieses Wortes. Die Parametrier-ID's (Adressen) für die einzelnen Parameter des *MPU1-F* finden sich in Kapitel 5 wieder.

e.) Byte 4

Der zur Parametrier-ID (Adresse) zugehörige Wert, der vom Slave gesendet wird beansprucht ein 16Bit-Wort, in Byte 4 steht das Highbyte dieses Wortes. Die Berechnungsformeln für die einzelnen Parameter finden sich in Kapitel 5 wieder.

f.) Byte 5

Der zur Parametrier-ID (Adresse) zugehörige Wert, der vom Slave gesendet wird beansprucht ein 16Bit-Wort, in Byte 5 steht das Lowbyte dieses Wortes. Die Berechnungsformeln für die einzelnen Parameter finden sich in Kapitel 5 wieder.

g.) Byte 6

Die Prüfsumme beansprucht ein 16Bit-Wort, diese muss vom Master überprüft werden und wird durch die dargestellte logische Verknüpfung erzeugt:

Prüfsumme = Wort1(Byte0, Byte1) XOR Wort2(Byte2, Byte3) XOR Wort3(Byte4, Byte5)

Da diese Verknüpfung Bitweise erzeugt wird, kann Byte6 wie folgt vom Master ausgewertet werden: **Byte6 = Byte0 XOR Byte2 XOR Byte4**

h.) Byte 7

Die Prüfsumme beansprucht ein 16Bit-Wort, diese muss vom Master überprüft werden und wird durch die dargestellte logische Verknüpfung erzeugt:

Prüfsumme = Wort1(Byte0, Byte1) XOR Wort2(Byte2, Byte3) XOR Wort3(Byte4, Byte5)

Da diese Verknüpfung Bitweise erzeugt wird, kann Byte7 wie folgt vom Master ausgewertet werden: **Byte7 = Byte1 XOR Byte3 XOR Byte5**

3 Sendetelegramm des MPU1-F

Nr	CAN-Bus	INHALT (16BIT-WORT)	Einheit	Bemerkung												
1	Mux 0 Wort 1	Kennung	"1400"	Telegrammtyp												
2	Mux 0 Wort 2	Gen.-Spannung L1 L2	$(10^{\text{UGNEXPO}})\text{V}$	Der Master muss zusammen mit dem Spannungsexponenten UGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
3	Mux 0 Wort 3	Gen.-Spannung L2 L3	$(10^{\text{UGNEXPO}})\text{V}$	Der Master muss zusammen mit dem Spannungsexponenten UGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
4	Mux 1 Wort 1	Gen.-Spannung L3 L1	$(10^{\text{UGNEXPO}})\text{V}$	Der Master muss zusammen mit dem Spannungsexponenten UGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
5	Mux 1 Wort 2	Gen.-Spannung L1 N	$(10^{\text{UGNEXPO}})\text{V}$	Der Master muss zusammen mit dem Spannungsexponenten UGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
6	Mux 1 Wort 3	Gen.-Spannung L2 N	$(10^{\text{UGNEXPO}})\text{V}$	Der Master muss zusammen mit dem Spannungsexponenten UGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
7	Mux 2 Wort 1	Gen.-Spannung L3 N	$(10^{\text{UGNEXPO}})\text{V}$	Der Master muss zusammen mit dem Spannungsexponenten UGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
8	Mux 2 Wort 2	Gen-Frequenz L1/L2	$\frac{1}{100}\text{Hz}$													
9	Mux 2 Wort 3	Gen.-Strom L1	$(10^{\text{IGNEXPO}})\text{A}$	Der Master muss zusammen mit dem Stromexponenten IGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
10	Mux 3 Wort 1	Gen.-Strom L2	$(10^{\text{IGNEXPO}})\text{A}$	Der Master muss zusammen mit dem Stromexponenten IGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
11	Mux 3 Wort 2	Gen.-Strom L3	$(10^{\text{IGNEXPO}})\text{A}$	Der Master muss zusammen mit dem Stromexponenten IGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
12	Mux 3 Wort 3	Gen.- $\cos\varphi$	$\frac{1}{100}$	<table><tr><th>Realer Wert</th><th>Übertragener Wert</th></tr><tr><td>$\cos\varphi = -0,98\text{ k}$</td><td>Hex FF9E</td></tr><tr><td>$\cos\varphi = -0,99\text{ k}$</td><td>Hex FF9D</td></tr><tr><td>$\cos\varphi = 1,00$</td><td>Hex 0064</td></tr><tr><td>$\cos\varphi = +0,99\text{ i}$</td><td>Hex 0063</td></tr><tr><td>$\cos\varphi = +0,98\text{ i}$</td><td>Hex 0062</td></tr></table>	Realer Wert	Übertragener Wert	$\cos\varphi = -0,98\text{ k}$	Hex FF9E	$\cos\varphi = -0,99\text{ k}$	Hex FF9D	$\cos\varphi = 1,00$	Hex 0064	$\cos\varphi = +0,99\text{ i}$	Hex 0063	$\cos\varphi = +0,98\text{ i}$	Hex 0062
Realer Wert	Übertragener Wert															
$\cos\varphi = -0,98\text{ k}$	Hex FF9E															
$\cos\varphi = -0,99\text{ k}$	Hex FF9D															
$\cos\varphi = 1,00$	Hex 0064															
$\cos\varphi = +0,99\text{ i}$	Hex 0063															
$\cos\varphi = +0,98\text{ i}$	Hex 0062															
13	Mux 4 Wort 1	Gen.-Wirkleistung	$(10^{\text{PGNEXPO}})\text{W}$	Der Master muss zusammen mit dem Leistungsexponenten aus dem Busprotokoll den tatsächlichen Wert errechnen.												
14	Mux 4 Wort 2	Gen.-Blindleistung	$(10^{\text{QGNEXPO}})\text{var}$	Der Master muss zusammen mit dem Blindleistungsexponenten QGNEXPO aus dem Bus-protokoll den tatsächlichen Wert errechnen.												
15	Mux 4 Wort 3	Intern	Intern	Intern												
16	Mux 5 Wort 1	Intern	Intern	Intern												
17	Mux 5 Wort 2	Exponenten	1	Highbyte: Ausmaskieren Lowbyte: UGNEXPO (Spannung)												
18	Mux 5 Wort 3	Exponenten	1	Highbyte: IGNEXPO (Strom) Lowbyte: PGNEXPO (Leistung)												
19	Mux 6 Wort 1	Gen.-Wirkarbeit Highwort	2^{16} kWh	Positive Wirkarbeit (Abgabe)												
20	Mux 6 Wort 2	Gen.-Wirkarbeit Lowwort	kWh	Positive Wirkarbeit (Abgabe)												
21	Mux 6 Wort 3	Digitaleingänge	Bits	Bit 0: DI 1 Klemme 3 (Schalter EIN/AUS) Bit 1: DI 2 Klemme 4 (Schalter Test/Betrieb) Bit 2: DI 3 Klemme 5 (Erdungsschalter EIN/AUS) Bit 3: DI 4 Klemme 6												
22	Mux 7 Wort 1	Intern	Intern	Intern												
23	Mux 7 Wort 2	Intern	Intern	Intern												
24	Mux 7 Wort 3	Intern	Intern	Intern												
25	Mux 8 Wort 1	Intern	Intern	Intern												
26	Mux 8 Wort 2	Intern	Intern	Intern												
27	Mux 8 Wort 3	Intern	Intern	Intern												
28	Mux 9 Wort 1	Gen.-Wirkarbeit Highwort	2^{16} kWh	Negative Wirkarbeit (Bezug)												
29	Mux 9 Wort 2	Gen.-Wirkarbeit Lowwort	kWh	Negative Wirkarbeit (Bezug)												
30	Mux 9 Wort 3	Gen.-Blindarbeit Highwort	2^{16} kvarh	Positive Blindarbeit (Induktiv)												
31	Mux 10 Wort 1	Gen.-Blindarbeit Lowwort	kvarh	Positive Blindarbeit (Induktiv)												
32	Mux 10 Wort 2	Gen.-Blindarbeit Highwort	2^{16} kvarh	Negative Blindarbeit (Kapazitiv)												
33	Mux 10 Wort 3	Gen.-Blindarbeit Lowwort	kvarh	Negative Blindarbeit (Kapazitiv)												

4 Parametrier-ID's des MPU1-F

4.1 Begriffe

In diesem Abschnitt wird jeder Parameter mit den Angaben Art, ID, Wertebereich und Berechnungsformel beschrieben. Die Bedeutung dieser vier Angaben ist hier definiert.

4.1.1 Art

Hieraus können Sie ersehen, ob Sie ein ganzes Wort schreiben müssen, oder nur ein Bit modifizieren.

4.1.2 ID

Die ID gibt die Adresse des Parameters innerhalb des **MPU1-F** wieder. Mit dieser Angabe können Sie wie in Kapitel 3 beschrieben auf einen Parameter zugreifen.

4.1.3 Wertebereich

Der Wertebereich gibt an, innerhalb welcher Grenzen sich der Wert, den Sie tatsächlich über den Bus senden, befinden darf. Der Wert, den Sie über den Bus senden entspricht in vielen Fällen nicht der realen Größe, die Sie z.B. am Display des **MPU1-F** sehen. Beträgt die reale Größe z.B. 70Hz, lautet der tatsächlich zu parametrierende Wert 17920. Der Wertebereich beschreibt also die Grenzen des tatsächlich über den Bus gesendeten Wertes. Die vorgegebenen Grenzen sind unbedingt einzuhalten.

4.1.4 Berechnungsformel

Die Berechnungsformel beschreibt, wie Sie eine reale Prozessgröße in den Parametrierwert umwandeln müssen, den Sie anschließend über den Bus senden.

4.2 Parametrierung über den CAN-Bus für das MPU1-F

4.2.1 Passwort für Codestufe Level 2

Art	Wort
ID	500
Wertebereich	0 bis 9999
Berechnungsformel	Das Passwort des Gerätes muss bekannt sein. Bevor über den Bus irgendein anderer Wert parametrieren werden kann, muss zuvor das Passwort für den Code Level 2 parametrieren werden.

4.2.2 Spannungswandler sekundär

Art	Wort
ID	534
Wertebereich	50 bis 480, dies entspricht 50V bis 480V.
Berechnungsformel	$\text{Parametrierwert } U [V] \cdot \frac{1}{V}$

4.2.3 Spannungswandler primär

Art	Wort
ID	533
Wertebereich	10 bis 65000, dies entspricht 10V bis 65000V.
Berechnungsformel	Parametrierwert $U [V] \cdot \frac{1}{V}$

4.2.4 Stromwandler primär

Art	Wort
ID	535
Wertebereich	1 bis 9999, dies entspricht 1A bis 9999.
Berechnungsformel	Parametrierwert $I [A] \cdot \frac{1}{A}$

4.2.5 Messung Netzart

Art	Vier Bits
ID	527
Wertebereich	Bit 3 + Bit 2 + Bit 1 + Bit 0 (Zählung bei Null beginnend)
Mögliche Werte	0000 = 1W 0001 = 1W4 0010 = 1W3 0011 = 2W3 0100 = 2W4 Andere Werte, z.B. 1001, sind ungültig !

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben.

4.2.6 Impulsausgang Pulsdauer

Art	Wort
ID	917
Wertebereich	4 bis 100, dies entspricht 0,02s bis 0,50s.
Berechnungsformel	Parametrierwert $t [s] \cdot 200 \frac{1}{s}$

4.2.7 Impulsausgang Größe

Art	Drei Bits
ID	517
Wertebereich	Bit 2 + Bit 1 + Bit 0 (Zählung bei Null beginnend)
Mögliche Werte	000 = +kWh 001 = +kvarh 010 = -kvarh 011 = -kWh Andere Werte, z.B. 100, sind ungültig !

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben.

4.2.8 Impulsausgang Logik

Art	Bit
ID	519
Wertebereich	Bit 3 (Zählung bei Null beginnend)
Mögliche Werte	0 = negativ 1 = positiv

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben.

4.2.9 Impulsausgang Pulse pro kWh bzw. Pulse pro kvarh

Art	Wort
ID	532
Wertebereich	1 bis 1500, dies entspricht 0,1 bis 150 Impulsen.
Berechnungsformel	Parametrierwert = Pulse · 10

4.2.10 Analogausgang 1 AUS/0..20mA/4..20mA/-20mA-+20mA

Art	Zwei Bit
ID	697
Wertebereich	Bit 1 + Bit 0 (Zählung bei Null beginnend)
Mögliche Werte	00 = AUS 01 = 0..20mA 10 = 4..20mA 11 = -20mA - + 20mA

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben.

4.2.11 Analogausgang Größe

Art	Fünf Bits
ID	698
Wertebereich	Bit 4 + Bit 3 + Bit 2 + Bit 1 + Bit 0 (Zählung bei Null beginnend)
Mögliche Werte	00000 = "U L1N" 00001 = "U L2N" 00010 = "U L3N" 00011 = "U LN mit" 00100 = "U LN max" 00101 = "U LN min" 00110 = "U L12" 00111 = "U L23" 01000 = "U L31" 01001 = "U LL mit" 01010 = "U LL max" 01011 = "U LL min" 01100 = "Frequenz" 01101 = "I L1" 01110 = "I L2" 01111 = "I L3" 10000 = "I mit" 10001 = "I max" 10010 = "I min" 10011 = "I L1±" 10100 = "I L2±" 10101 = "I L3±" 10110 = "I± mit" 10111 = "I± max" 10000 = "I± min" 10001 = "Wirkl." 10010 = "Blindl." 10011 = "Scheinl." 10100 = "cos" Andere Werte, z.B. 10111, sind ungültig !

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben.

4.2.12 Analogausgang unterer Wert

Art	Wort
ID	706
Wertebereich	-32000 bis 32000
Berechnungsformel	Parametrierwert = Wert

4.2.13 Analogausgang oberer Wert

Art	Wort
ID	707
Wertebereich	-32000 bis 32000
Berechnungsformel	Parametrierwert = Wert

4.2.14 Anzeige kWh aktiv JA/NEIN

Art	Doppel-Bit
ID	805
	Bit 7 + 6 (Zählung bei 0 beginnend)
Wertebereich	00 = Keine Anzeige aktiv
	01 = +kWh Anzeige ist aktiv
	10 = -kWh Anzeige ist aktiv
	11 = +kWh und -kWh Anzeige ist aktiv

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben.

4.2.15 Anzeige kvarh aktiv JA/NEIN

Art	Doppel-Bit
ID	805
	Bit: 5 + 4 (Zählung bei 0 beginnend)
Wertebereich	00 = Keine Anzeige aktiv
	01 = +kvarh Anzeige ist aktiv
	10 = -kvarh Anzeige ist aktiv
	11 = +kvarh und -kvarh Anzeige ist aktiv

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben.

4.2.16 Reset kWh/kvarh

Art	Bit
ID	519
Wertebereich	Bit 2 (Zählung bei Null beginnend)
Mögliche Werte	0 = AUS
	1 = EIN

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben.

4.2.17 Highword Zähler pos. kWh stellen

Art	Wort
ID	867
Wertebereich	0 bis 65535, dies entspricht $0 \cdot 2^{16} \text{ kWh}$ bis $65535 \cdot 2^{16} \text{ kWh}$
Berechnungsformel	$\text{Parametrierwert} = \text{Wert} [2^{16} \text{ kWh}] \frac{1}{\text{kWh}}$

4.2.18 Lowword Zähler pos. kWh stellen

Art	Wort
ID	868
Wertebereich	0 bis 65535, dies entspricht 0 kWh bis 65535 kWh
Berechnungsformel	$\text{Parametrierwert} = \text{Wert} [\text{kWh}] \frac{1}{\text{kWh}}$

4.2.19 Zähler pos. Wh stellen

Art	Wort
ID	869
Wertebereich	0 bis 999, dies entspricht 0 Wh bis 999 Wh
Berechnungsformel	Parametrierwert = Wert [Wh] $\frac{1}{\text{Wh}}$

4.2.20 Highword Zähler pos. kvarh stellen

Art	Wort
ID	870
Wertebereich	0 bis 65535, dies entspricht $0 \cdot 2^{16}$ kvarh bis $65535 \cdot 2^{16}$ kvarh
Berechnungsformel	Parametrierwert = Wert [2^{16} kvarh] $\frac{1}{\text{kvarh}}$

4.2.21 Lowword Zähler pos. kvarh stellen

Art	Wort
ID	871
Wertebereich	0 bis 65535, dies entspricht 0 kvarh bis 65535 kvarh
Berechnungsformel	Parametrierwert = Wert [kvarh] $\frac{1}{\text{kvarh}}$

4.2.22 Zähler pos. varh stellen

Art	Wort
ID	872
Wertebereich	0 bis 999, dies entspricht 0 varh bis 999 varh
Berechnungsformel	Parametrierwert = Wert [varh] $\frac{1}{\text{varh}}$

4.2.23 Highword Zähler neg. kWh stellen

Art	Wort
ID	873
Wertebereich	0 bis 65535, dies entspricht $0 \cdot 2^{16}$ kWh bis $65535 \cdot 2^{16}$ kWh
Berechnungsformel	Parametrierwert = Wert [2^{16} kWh] $\frac{1}{\text{kWh}}$

4.2.24 Lowword Zähler neg. kWh stellen

Art	Wort
ID	874
Wertebereich	0 bis 65535, dies entspricht 0 kWh bis 65535 kWh
Berechnungsformel	Parametrierwert = Wert [kWh] $\frac{1}{\text{kWh}}$

4.2.25 Zähler neg. Wh stellen

Art	Wort
ID	875
Wertebereich	0 bis 999, dies entspricht 0 Wh bis 999 Wh
Berechnungsformel	$\text{Parametrierwert} = \text{Wert [Wh]} \cdot \frac{1}{\text{Wh}}$

4.2.26 Highword Zähler neg. kvarh stellen

Art	Wort
ID	876
Wertebereich	0 bis 65535, dies entspricht $0 \cdot 2^{16}$ kvarh bis $65535 \cdot 2^{16}$ kvarh
Berechnungsformel	$\text{Parametrierwert} = \text{Wert [} 2^{16} \text{ kvarh]} \cdot \frac{1}{\text{kvarh}}$

4.2.27 Lowword Zähler pos. kvarh stellen

Art	Wort
ID	877
Wertebereich	0 bis 65535, dies entspricht 0 kvarh bis 65535 kvarh
Berechnungsformel	$\text{Parametrierwert} = \text{Wert [kvarh]} \cdot \frac{1}{\text{kvarh}}$

4.2.28 Zähler pos. varh stellen

Art	Wort
ID	878
Wertebereich	0 bis 999, dies entspricht 0 varh bis 999 varh
Berechnungsformel	$\text{Parametrierwert} = \text{Wert [varh]} \cdot \frac{1}{\text{varh}}$

4.2.29 Zykluszeit Display

Art	Wort
ID	916
Wertebereich	2 bis 500, dies entspricht 0,01s bis 2,50s
Berechnungsformel	$\text{Parametrierwert} = t [\text{s}] \cdot 200 \frac{1}{\text{s}}$

4.2.30 Messwertfilter Spannung

Art	Drei Bit
ID	900
Wertebereich	000 = AUS 001 = 0,04s 010 = 0,08s 011 = 0,16s 100 = 0,32s 101 = 0,64s 110 = 1,28s 111 = 2,56s

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben

4.2.31 Messwertfilter Strom

Art	Drei Bit
ID	903
Wertebereich	000 = AUS 001 = 0,04s 010 = 0,08s 011 = 0,16s 100 = 0,32s 101 = 0,64s 110 = 1,28s 111 = 2,56s

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben

4.2.32 Messwertfilter Leistung

Art	Drei Bit
ID	905
Wertebereich	000 = AUS 001 = 0,04s 010 = 0,08s 011 = 0,16s 100 = 0,32s 101 = 0,64s 110 = 1,28s 111 = 2,56s

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben

4.2.33 Messwertfilter Frequenz

Art	Drei Bit
ID	913
Wertebereich	000 = AUS 001 = 0,04s 010 = 0,08s 011 = 0,16s 100 = 0,32s 101 = 0,64s 110 = 1,28s 111 = 2,56s

Bei einer Bitmodifikation ist **immer** zuerst das gesamte Wort auf der entsprechenden ID auszulesen, anschließend sind die betroffenen Bits maskiert zu bearbeiten, erst dann wird das komplette Wort wieder zurückgeschrieben



Woodward SEG GmbH & Co. KG

Krefelder Weg 47 · D – 47906 Kempen (Germany)

Postfach 10 07 55 (P.O.Box) · D – 47884 Kempen (Germany)

Phone: +49 (0) 21 52 145 1

Internet

Homepage <http://www.woodward-seg.com>

Documentation <http://doc.seg-pp.com>

Sales

Phone: +49 (0) 21 52 145 635 · Telefax: +49 (0) 21 52 145 354

e-mail: kemp.electronics@woodward.com

Service

Phone: +49 (0) 21 52 145 614 · Telefax: +49 (0) 21 52 145 455

e-mail: kemp.pd@woodward.com