

SMI Telegramme

[Mikrokontroller.net Beitrag](#)

Die Telegramme können unterschiedlich lang sein und haben folgenden Aufbau

Adresse:

Bit	7	6	5	4	3	2	1	0
Bemerk.	Typ				Adressetyp			
Wert	2^2	2^1	2^0	2^0	2^3	2^2	2^1	2^0

- **Typ**

- 1: Diagnose
- 2: Fahrbefehl
- 3: Positionsabfrage

- **Adressetyp**

- 0: Hersteller ID
- 1: Motor ID

ID	Hersteller
0	Alle
1	Alcatel
2	Becker
3	ELERO
4	SELVE
5	Stele
6	Vestamatic

Adresserweiterung

- Wenn nur bestimmte Hersteller oder mehrere Motoren gleichzeitig angesprochen werden sollen. Das erste Byte wird dann der Hersteller gesendet

Bit	7	6	5	4	3	2	1	0
Byte1	1	1	1	0	0	0	0	0
Byte2: Motor ID	16	15	14	13	12	11	10	9
Byte3: Motor ID	8	7	6	5	4	3	2	1

- **Beispiel:** Motor 16,5 und 2:

- 0xC0 0x80 0x12

ID setzen

- Motoren werden immer mit ID=0 ausgeliefert. Nach der Motorsuche werden die IDs von 1 herunter bis 1 vergeben. Über diese ID können die Motoren am SMI-Bus angesprochen werden. Wenn schon alle IDs bis 1 vergeben wurden, behält der letzte Motor die ID 0. Über diese ID können an jedem SMI-Bus also 16 Motoren angesprochen werden.

Bit	7	6	5	4	3	2	1	0
Byte1	0	0	1	1	1	1	0	0
Byte2: Motor ID					2^3	2^2	2^1	2^0

- Beispiel:** ID zurücksetzen
 - 0x3C 0x00

Befehl:

Bit	7	6	5	4	3	2	1	0
Bemerk.	Erweiterung		Datenbytes	???	???			Befehl
Wert	2^0	2^1	2^0				2^2	2^1

- Erweiterung**
 - 0: keine weiteren Daten
 - 1: weitere Daten (Tuchstraffung 6°: 0x22 0x06???)
- Befehl**
 - 0: Stop
 - 1: Hoch
 - 2: Runter
 - 3: Position1
 - 4: Position2
 - 5: Position anfahren
- Datenbytes**
 - 1: Relative Position (0x00-0xff | 0-512°)
 - Winkel Hoch (8Bit Winkel [°])
 - Winkel Runter (8Bit Winkel [°])
 - 2: Genaue Positionen (0x0000-0xffff | 0-100%)
 - Position1 speichern (16Bit Position [%])
 - Position2 speichern (16Bit Position [%])
 - Position anfahren (16Bit Position [%])

Prüfsumme:

Es werden alle Bytes addiert. Aus der Summe wird das Zweierkomplement gebildet (Negieren und +1)

Beispiel: Motor 12 Hoch (0x5C 0x01):

Addition:

$$0x5C + 0x01 = 0x5D$$

1er Komplement:



$$! 0x5D = 0A2$$

$$0xA2 + 1 = 0xA3$$

Ergebnis:

$$0x5C \ 0x01 \ 0xA3$$

Antwort:

- Die **Antwort** besteht aus mindestens einem bis fünf Byte(s) (Diagnose mit Motor ID)
 - Bestätigung | ACK (0xff)
 - Negative Bestätigung | NACK (0xE0?)

Beispiele

Motor 12 Hoch: > 5C 01 A3 < FF

Motor 12 100° Hoch: > 5C 21 64 1F < FF

Motor 12 Pos 50%: > 5C 45 7F FF E1 < FF

Diagnose alle Herst: > 20 00 < E0

Diagnose Motor 1: > 31 00 CF < FF FF FF E0 FF (alle stehen)

Diagnose Motor 1: > 31 00 CF < FF E0 FF FF FF (mind. 1 Motor läuft
(ja, 1 nach AUF))

Diagnose Motor 1: > 31 00 CF < FF FF E0 FF FF (mind. 1 Motor läuft
(ja, 1 nach AB))

Motor Suchlauf:

Suchlauf 1: > 24 7D ID ID ID ID XX < FE FE FE C0 C0 (mind. 1 Seriennummer
passt, Motor mit ID vorhanden)

Suchlauf 1: > 24 7D ID ID ID ID XX < FE FE FE C0 FF (mind. 1 Seriennummer
passt. kein Motor mit ID vorhanden)

Suchlauf 1: > 24 7D ID ID ID ID XX < FF E0 FF FF FF (mind. 1 Seriennummer
ist kleiner)

Suchlauf 1: > 24 7D ID ID ID ID XX < FF FF E0 FF FF (mind. 1 Seriennummer
ist größer)

Motor blockieren?

Motor 9 Hoch: > 59 00 A7 < 7F 56 80 FA (Motor blockiert?)

Motor 12 Hoch: > 5C 00 A4 < FF AE 00 A4 (Motor blockiert?)

From:
<https://smiwiki.thefischer.net/> -

Permanent link:
<https://smiwiki.thefischer.net/doku.php?id=wiki:smi:telegramme&rev=1707662613> 

Last update: **2024/02/11 15:43**