Dokumentation Schwabenplan Protokoll Version 4.0

1. Einleitung

Das Schwabenplan MC-Protokoll wurde entwickelt, um einfache Steuerungsbefehle von einer Software oder einen Controller zu einem Busteilnehmer zu übermitteln. Das Protokoll basiert auf einem String und ist über ein Terminalprogramm einfach zu lesen. Die Länge des Protokolls ist fest vorgegeben.

Die Programmiersprache für die Anwendung muss nicht zwingend beachtet werden. Wir verwenden üblicherweise Visual Basic .Net und bieten das Protokoll auch in unserer eigenen DLL-Library an. Somit können Sie eine beliebige Einwicklungsplattform für Ihre Anwendung einsetzen.

Unsere Hauptplatinen können über eine RS-232, RS-485 oder eine CAN-Schnittstelle angebunden werden. Wir favorisieren die RS-485 Schnittstelle, da in diesem Bus mehrere Teilnehmer angebunden werden können. Die Konfiguration Schnittstelle ist frei definierbar. Unsere Programmbeispiele basieren auf folgenden Einstellungen:

Baud-Rate: 9600 / 19200 / 57600 / 115200 Baud

Datenbits: 8

Stopbits: 1

Parität: Keine

Intervallzeit: 80-150ms

Schnittstellen: RS-232 und RS-485 (RS-485 wird von unserer Seite empfohlen)

Das Schwabenplan Protokoll kann für eine einzelne Datenübertragung und im Polling-Modus betrieben werden.

Wichtiger Hinweis:

Die Intervallzeit für die permanente Datenabfrage hängt von verschiedenen Faktoren ab.

Diese sind:

a) Der Taktgeschwindigkeit des Mikrocontrollers. -> Programmdurchlaufzeit

b) Der angeschlossenen Hardware z.B. mit oder ohne LC-Display. -> Programmdurchlaufzeit

c) Den Parameter der verwendeten Schnittstellenbausteine (vom Sender und Empfänger).

d) Der Verkabelung der Hardware selbst Länge, Abschlusswiderstände, Qualität der Kabel usw.

1.1 Integration

Für die Integration dieses Übertragungsprotokolls stellen wir Ihnen sehr gerne unser Demo-Projekt „MC Agent“ zur Verfügung. Der MC Agent wurde mit Visual Basic .Net programmiert. Für den Microcontroller haben wir Beispiele in der Programmiersprache C für den PIC18F46K80 sowie den PIC18F97J60. Sprechen Sie uns im Bedarfsfall einfach an. Es müssen nicht alle Befehle in Ihre Anwendung integriert werden. Wir wollen Ihnen Anhand dieser umfangreichen Protokollbeschreibung die Möglichkeiten für Ihre tägliche Arbeit aufzeigen. Weiterentwicklungen wird es auch in Zukunft geben. Daher behalten wir uns Änderungen vor.

In unseren Projekten haben wir dieses Protokoll als Betriebssystem mit zwei Interrupts integriert. Diese Interrupts werden über die interne Prioritätsverwaltung gesteuert. Der seriellen Schnittstelle Nr. 1 ist die High-Priorität zugeordnet. So wird jede Abfrage oder Steuerungsbefehl verarbeitet. Dem Timer Nr. 0 ist die Low-Priorität zugeordnet, und erzeugt jede Sekunde ein Taktsignal. Dieses Taktsignal wird für die Systemuhr und den Temperatursensor verwendet. So können die Uhrzeit und das Datum unabhängig vom restlichen Programmablauf angezeigt oder geschrieben werden. Tritt ein zeitgleiches Ereignis ein, so werden zuerst die Daten über die serielle Schnittstelle verarbeitet. Im Nachgang werden die Daten für die Echtzeituhr und die Temperaturwerte auf der Platine aktualisiert. Die nachfolgende Tabelle zeigt de einzelnen Einstellungen der Interrupt-Steuerung auf.

Interrupt Bezeichnung	Interrupt Priorität	Interrupt Bit	Interrupt Flag	Parameter	Info
Allgemein	Prioritätsverwaltung	RCONbits.IPEN	-	1	-
Global Interrupt High (Hauptschalter High-Priorität)	INTCONbits.GIEH	INTCONbits.GIEH	-	0 oder 1	-
Global Interrupt Low (Hauptschalter Low-Priorität)	INTCONbits.GIEL	INTCONbits.GIEL	-	0 oder 1	-
USART Nr. 1 Datenempfang	High-Priorität	PIE1bits.RC1IE	PIR1bits.RC1IF	0 oder 1	-
Timer Nr. 0	Low-Priorität	INTCONbits.TMR0IE	INTCONbits.TMR0IF	0 oder 1	Takt 1 Hz.

a) Eine CRC-Prüfung gibt es in diesem Protokoll nicht.

b) Eine korrekt empfange Nachricht wird unmittelbar mit einer Antwort (Quittierung) bestätigt.

c) Erhält der Sender keine Quittierung so hat der Empfänger die Nachricht fehlerhaft oder nicht erhalten.

d) In jedem Fall sind entsprechende Timeout-Zeiten integrierbar.

e) Jede Nachricht in diesem Protokoll ist immer gleich groß.

f) Eine dynamische Anpassung der zu reservierenden Datenfelder ist nicht notwendig.

2. Allgemeines

Das Protokoll ist grundsätzlich in drei Kategorien aufgeteilt. Es gibt:

- ✓ Status-Befehle

  Über die Status-Befehle kann die Peripherie der gesamten Hard- und Firmware abgefragt werden.

- ✓ PORT-Befehle

  Mit diesen Befehlen kann jeder beliebige PORT-Zustand eingelesen oder gesteuert werden.

- ✓ I²C-Bus bzw. SPI-Bus Befehle

  Über diese Befehle können eine Vielzahl von Bausteinen direkt abgefragt und gesteuert werden.

Wichtiger Hinweis:

Es ist zu empfehlen das die empfangenen Daten als globale Variablen zu deklarieren oder mit Pointen zu arbeiten.

3. Befehlsaufbau

Die Befehle sind sehr einfach gehalten und bestehen immer aus 39 Zeichen. Es gibt in diesem Protokoll Lese- und Schreibbefehle. Diese Befehle sind immer gleich formatiert. Sie beginnen mit einem Startzeichen (Header „S“ oder „R“) und werden durch einen „:“ voneinander getrennt. Zum Abschluss von jedem Befehl folgen die Steuerzeichen „\r\n“, welche „Carrige Return“, 0x0D und dem Befehl „Newline“, 0x0A bedeutet.

Die Sende- und Empfangsbefehle werden zur vereinfachten Lesbarkeit mit einem „S“ für **S**end bzw. mit einem „R“ für **R**eceive“ für Empfangen gegenzeichnet.

So können Sie direkt erkennen, ob es sich bei diesem Befehl um einen Sende- oder Rückgabebefehl handelt. Im Schwabenplan-Protokoll der Version 4.0 ist das Trennzeichen der unterschiedlichen Felder als „:“ ausgeführt. Alle zu übertragenden Zahlenwerte sind immer zweistelligen hexadezimalen Werten zu übertragen.

3.1 Protokoll Header

Bezeichnung	ASCII Wert	Dez. Wert	Hex. Wert
Senden	S	83	0x53
Empfangen	R	82	0x52
Trennzeichen	:	58	0x3A
Endzeichen Nr. 1	\r	13	0x0D
Endzeichen Nr. 2	\n	10	0x0A

3.2 Read / Write (R/W) Umschaltung

Im letzten Block des übertragenen Protokoll-Strings ist definiert, ob es sich um einen Lese- oder einen Schreibbefehl handelt.

Befehlsart	Hex. Wert
Schreiben	0x00
Lesen	0x01

Beispiele:

Sendebefehl

S : Adresse des Empfängers: Adresse des Senders : Befehl : P0 : P1 : P2 : P3 : P4 : P5 : P6 : P7 : RW \r \n

Antwort des Empfängers

R : Adresse des Empfängers: Adresse des Senders : Befehl : P0 : P1 : P2 : P3 : P4 : P5 : P6 : P7 : RW \r \n

Hier ein exemplarisches Beispiel mit dem Befehl 0x20 zum Lesen:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
S	:	0	1	:	0	0	:	2	C	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	\r	\n
53	3A	30	31	3A	30	30	3A	32	43	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A
083	058	048	049	058	048	048	058	050	067	058	048	048	058	048	048	058	048	048	058	048	048	058	048	048	058	048	048	058	048	048	058	048	048	058	048	049	013	010
R	:	0	0	:	0	1	:	2	C	:	0	1	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	\r	\n
52	3A	30	30	3A	30	31	3A	32	43	3A	30	31	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A
082	058	048	048	058	048	049	058	050	067	058	048	049	058	048	048	058	048	048	058	048	048	058	048	048	058	048	048	058	048	048	058	048	048	058	048	049	013	010

S:01:00:20:P0:P1:P2:P3:P4:P5:P6:P7:**01:\r\n**

// Der Server mit der Adresse 0 sendet einen Status-Befehl zum Lesen an den Teilnehmer mit der Adresse 1

R:00:01:20:P0:P1:P2:P3:P4:P5:P6:P7:**01:\r\n**

//Die Adresse 1 antwortet dem Server mit einem führenden „R:“ und die entsprechende Adresse 0

Wichtiger Hinweis:

Bitte achten Sie darauf, dass alle hexadezimalen Werte zweistellige Längen angegeben werden müssen. Sollten Sie über unsere DLL-Datei arbeiten, wird dies automatisch angepasst.

Die Bearbeitungszeit für einen Befehl ist von verschiedenen Faktoren wie Microcontroller, Taktgeschwindigkeit, Speicher oder Baudrate abhängig. Diese Intervallzeit kann mit Hilfe von unserem Software-Tool beliebig eingestellt werden.

4. Allgemeine Befehlsübersicht

4.1 Status Befehle

Befehl	Dez	Beschreibung
0x20	32	MC-Typ
0x21	33	Hardware Version
0x22	34	Taktfrequenz
0x23	35	Firmware Version
0x24	36	Protokoll Version
0x25	37	RS-485 Adresse
0x26	38	Bootloader Modus
0x27	39	Debug Modus
0x28	40	Interface on Board
0x29	41	Baudrate
0x2A	42	I²C-Geschwindigkeit
0x2B	43	RTC-Baustein
0x2C	44	Temperatur-Baustein
0x2D	45	Board-Edition

4.2 Netzwerk-Befehle

Befehl	Dez	Beschreibung
0x30	48	IP-Adresse
0x31	49	Subnet Maske
0x32	50	Router
0x33	51	Gateway
0x34	52	DNS-Server
0x35	53	Server IP
0x36	54	Kommunikations-Port
0x37	55	MAC-Adresse

4.3 I²C-Bus-Befehle

Befehl	Dez	Beschreibung
0x40	64	LM75 Temperatur Lesen
0x41	65	DS1621 Temperatur Lesen
0x43	66	MCP23008
0x44	67	MCP23017
0x45	68	PCF8574
0x46	69	PCF8591
0x50	80	Uhrzeit
0x51	81	Datum und Wochentag
0x52	82	Stunden
0x53	83	Minuten
0x54	84	Sekunden
0x55	85	Tag
0x56	86	Monat
0x57	87	Jahr
0x58	88	Wochentag
0xAD	173	A/D-Wandler
0xDD	221	Digitalbefehl
0xFF	255	Reset

5. Status-Befehle

Inzwischen haben die meisten Microcontroller einen großen Speicher. Daher werden in Ihnen zum Teil einige Zusatzinformationen wie Microcontroller-Typ, Hardware-Version usw. hinterlegt.

Die „Status-Befehle“ von dem Schwabenplan Protokoll umfassen Befehle, die Ihnen Auskünfte zu dem Microcontroller und der jeweiligen Hardware und Peripherie beinhalten.

Hierzu werden die Befehlsparameter P0 und P1 genützt. Für die Abfrage können die Parameterfelder P0 bis P7 mit einem beliebigen Wert befüllt werden. Wir verwenden hier im Normalfall immer 0x00.

Befehl	Beschreibung	P0	P1
0x20	MC-Typ	Integer	
0x21	Hardware-Version	Integer	Integer (0x2D)
0x22	Taktfrequenz	Integer	
0x23	Firmware Version	Integer	
0x24	Protokoll Version	Integer	
0x25	RS-485 Adresse	Integer	
0x26	Bootloader Modus	Integer	
0x27	Debug Modus	Integer	
0x28	Interface on Board	Integer	Integer (0x2A)
0x29	Baudrate	Integer	
0x2A	I²C-Geschwindigkeit	Integer	
0x2B	RTC-Baustein	Integer	
0x2C	Temperatur-Baustein	Integer	
0x2D	Board-Edition	Integer	

Wichtiger Hinweis:

Einige diese Paramater sollten lediglich als „ein reiner Lesebefehl“ in Ihr Projekt integriert werden.

5.2.1 Befehl 0x20 -> Microcontroller-Typen

Mit diesem Befehl wird der Microcontroller-Typ der auf der Leiterplatte abgefragt. Dies ist ein Befehl, der nur als ein reiner Lesebefehl in Ihr Projekt integriert werden sollte.

Dieser Befehl liefert nur an dem Parameter Nr. 0 einen Rückgabewert. Diesen finden Sie in der nachfolgenden Tabelle.

Hex. Wert	Microcontroller-Typ	Bezeichnung
0x00	unbekannt	
0x01	PIC18F97J60	aktuelle Generation
0x02	PIC18F46K80	aktuelle Generation
0x03	PIC18F26K80	aktuelle Generation
0x04	PIC18F66K80	aktuelle Generation
0x05	PIC18F67K40	aktuelle Generation
0x0A	PIC18F25K50	USB-Typen
0x0B	PIC18F45K50	USB-Typen
0x0C	PIC18F2550	USB-Typen
0x0D	PIC18F4550	USB-Typen
0x0E	PIC18F67J50	USB-Typen
0x0F	PIC18F87J50	USB-Typen
0x14	PIC18F06Q41	neue Generation
0x15	PIC18F16FQ41	neue Generation
0x16	PIC18F27Q43	neue Generation
0x17	PIC18F47Q43	neue Generation
0x18	PIC18F57Q43	neue Generation
0x1E	PIC18F67K22	RTCC-Typen
0x1F	PIC18F87K22	RTCC-Typen
0x20	PIC18F67J94	RTCC-Typen
0x21	PIC18F87J94	RTCC-Typen
0x22	PIC18F97J94	RTCC-Typen
0x5A	PIC16F877A	alte Generation
0x5B	PIC18F452	alte Generation
0x5C	AT89C51CC03	alte Generation

5.2.2 Befehl 0x21 und 0x2D -> Hardware Version

Mit diesem Befehl wird der Microcontroller-Typ der auf der Leiterplatte abgefragt. Dies ist ein Befehl, der nur als ein reiner Lesebefehl in Ihr Projekt integriert werden sollte. Änderungen ergeben sich in aller Regel nicht.

Dieser Befehl liefert zwei Rückgabewerte, als Parameter Nr. 0 und Nr. 1.

Die Versionsnummer 1.0 wird als dezimal 10 oder 0x0A in hexadezimal dargestellt. Mit der Erweiterung des Schwabenplan Protokolls auf die Version 4.0 wurde der Befehl [0x2D] in den Befehl [0x21] mit integriert. Der Befehl [0x2D] existiert als eigenständiger Befehl weiterhin, und antwortet dann als Parameter Nr. 0. In diesem Datenbyte ist die Hardwareausführung beschrieben. Unsere Hauptplatinen sind in verschiedenen Ausbaustufen erhältlich. Daher ist diese zusätzliche Unterscheidung möglich.

Diesen finden Sie in den nachfolgenden Tabellen. Hier ein Beispiel einer Hauptplatine der Version 2.0 in der Enterprise Ausführung.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
S	:	0	1	:	0	0	:	2	1	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	Vr	Vn
53	3A	30	31	3A	30	30	3A	32	31	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A
R	:	0	0	:	0	1	:	2	1	:	1	4	:	4	5	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	Vr	Vn
52	3A	30	30	3A	30	31	3A	32	31	3A	31	34	3A	34	35	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A

Befehl	P0 in Hex.	P1 in Hex.
0x21	0x14	0x45

ASCII-Wert	Dez. Wert	Hex. Wert	Bedeutung
B	66	0x42	Basic-Ausführung
P	80	0x50	Professional Ausführung
E	69	0x45	Enterprise Ausführung

5.2.3 Befehle 0x22 bis 0x25

Diese Befehle werden mit dem Parameter Nr. 0 in einer Ganzzahl (Integer) im hexadezimalen Zahlenformat beantwortet.

Befehl	Beschreibung	P0 in Dez.	Po in Hex.
0x22	Taktfrequenz	25	0x19
0x23	Firmware Version	10	0x0A
0x24	Protokoll Version	40	0x28
0x25	RS-485 Adresse	1	0x01

Für eine Taktfrequenz, Befehl [0x22] von 25 MHz wird der hexadezimale Wert 0x19 gesetzt. Die Taktfrequenz wird in aller Regel nicht vom Anwender verändert. Technisch gesehen, aber möglich. Daher sollten Sie entscheiden, wie Sie diesen Befehl in Ihre Applikation implementieren.

Für den Befehl [0x23] Firmware Version und dem Befehl [0x24] Protokoll Version wird der zu speichernde Wert mit einer durch 10 teilbaren Zahl befüllt. Für die Firmware Version 1.0 gilt hier der der dezimale Wert 10 bzw. hexadezimale Wert 0x0A. Diese Werte sollten Sie lesend und schreibend in Ihre Anwendung implementieren, damit die immer über den aktuellen Stand informiert werden können.

Hier ein Bespiel für den Befehle Firmware Version [0x23]:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
S	:	0	1	:	0	0	:	2	3	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	\r	\n
53	3A	30	31	3A	30	30	3A	32	33	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A
R	:	0	0	:	0	1	:	2	3	:	0	A	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	\r	\n
52	3A	30	30	3A	30	31	3A	32	33	3A	30	41	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A

Für die Version 4.0 gilt in dezimal die Zahl 40 oder bzw. in Hex 0x28.

Hier ein Bespiel für den Befehle Protokoll Version [0x24]:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
S	:	0	1	:	0	0	:	2	4	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	\r	\n
53	3A	30	31	3A	30	30	3A	32	34	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A
R	:	0	0	:	0	1	:	2	4	:	2	8	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	\r	\n
52	3A	30	30	3A	30	31	3A	32	34	3A	32	38	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A

Der Befehl [0x25] zum Setzen oder Lesen einer Bus Adresse z.B. für einen RS-485 Bus kann z.B. über DIP-Schalter per Hardware aber auch per Software gesetzt werden. Dafür dient dieser Befehl. So könne Sie Pins an einem Microcontroller einsparen und diese Funktion per Software realisieren.

Hier ein Bespiel für den Befehle [0x25] RS-485 Adresse Lesen und Schreiben:

Lesen:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
S	:	0	1	:	0	0	:	2	5	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	\r	\n
53	3A	30	31	3A	30	30	3A	32	35	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A
R	:	0	0	:	0	1	:	2	5	:	0	1	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	1	\r	\n
52	3A	30	30	3A	30	31	3A	32	35	3A	30	31	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	31	0D	0A

Schreiben Adresse 1 bleibt Adresse 1:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
S	:	0	1	:	0	0	:	2	5	:	0	1	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	\r	\n
53	3A	30	31	3A	30	30	3A	32	35	3A	30	31	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	0D	0A
R	:	0	0	:	0	1	:	2	5	:	0	1	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	:	0	0	\r	\n
52	3A	30	30	3A	30	31	3A	32	35	3A	30	31	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	3A	30	30	0D	0A

Das Schreiben einer neuen RS-485 Adresse bringt eine Besonderheit mit sich. Formell gesehen beantwortet der Microcontroller seinem Bus-Master zuerst den Befehl auf der bisherigen Teilnehmer-Adresse. Danach speichert der Controller die neue Adresse ab und liest diese ein. Je nachdem wie Schnell Ihr Controller, der Speicher, die Schnittstelle usw. ist, kann es sein, dass Ihr Controller eine zweite Bestätigung mit der neuen Teilnehmeradresse an den Bus-Master sendet.

5.2.4 Befehle 0x26 und 0x27

Diese Befehle dienen zur Steuerung eines Bootloaders oder einem zusätzlichen Debug Modus ohne Programmiergerät. Diese können im Bedarfsfall über Ihre Software direkt gesteuert werden. Der Wer 0x00 steht für Off und der Wert 0x01 steht für On.

Befehl	Beschreibung	P0 in Hex.
0x26	Bootloader Modus	0x00 oder 0x01
0x27	Debug Modus	0x00 oder 0x01

5.2.5 Befehle 0x28, 0x27 und 0x02A

Diese Befehle dienen zur Steuerung eines zusätzlichen Bussystems z.B. I²C- oder SPI-Bus. Der Befehl [0x2A] wurde im Zuge der Erweiterung im Befehl [0x28] integriert. Der Befehl [0x2A] steht als Einzelabfrage aber weiterhin zur Verfügung. Die Inhalte werden als Ganzzahl im zweistelligen Hex-Format übertragen.

Befehl	Beschreibung	P0 in Hex.	P1 in Hex.
0x28	Interface on Board	Integer	Integer (0x2A)
0x29	Baudrate	Integer	
0x2A	I²C-Geschwindigkeit	Integer	

Dateninhalte Befehl 0x28 Interface on Board:

Wert	Interface-Typ
0x00	Off
0x01	I2C Bus
0x02	SPI Bus

Dateninhalte Befehl 0x29 verfügbare Baudraten:

Eine RS-232 bzw. RS-485 Schnittstelle ist immer auf unseren Hauptplatinen verbaut.

Wert	Baudrate
0x00	9600
0x01	19200
0x02	57600
0x03	115200

Dieser Befehl hat ebenfalls eine Besonderheit. Wird die Baudrate im laufenden Betrieb geändert, sendet die Hauptplatine zuerst auf der Baudrate eine Bestätigung. Danach wird der neue Wert abgespeichert, und die Schnittstelle neu gestartet. Je nachdem wie schnell Ihr Controller ist, sendet dieser dann nochmals eine Bestätigung mit der neuen Baudrate,

Dateninhalte Befehl 0x29 verfügbare Baudraten:

Die Geschwindigkeit von dem I²C-Bus auf unseren Hauptplatinen ist steuerbar. Diese kann zwischen 100 und 400 KHz gesteuert werden. Der Inhalt von dieser Speicherstelle ist 0x01 für 100 KHz (Normal Speed) und 0x04 für 400 KHz (Fast Speed).

Wert	I²C-Bus Geschwindigkeit
0x01	100 KHz
0x04	400 KHz

5.2.6 Befehle 0x2B, 0x2C und 0x2D

Diese Befehle dienen zur Abfrage und Steuerung eines externen Bus-Teilnehmers wie einer Echtzeituhr (RTC), einem Temperaturbausteins oder einem Speicherbausteins. Die Inhalte werden als Ganzzahl im zweistelligen Hex-Format übertragen. Diese Befehle sind im Wesentlichen nur zum Lesen gedacht.

Befehl	Beschreibung
0x2B	RTC-Baustein
0x2C	Temperatur-Baustein
0x2D	Board-Edition

Die Inhalte bzw. Rückgabewerte für die unterstützten RTC-Typen (Befehl [0x2B]) sind wie folgt:

Hex. Wert	RTC-Typ
0x00	nicht vorhanden
0x01	DS1337
0x02	PCF8583
0x03	MCP794XX
0x04	DS3231

Die Inhalte bzw. Rückgabewerte für die unterstützten Temperatur-Sensoren (Befehl [0x2C]) sind wie folgt:

Hex. Wert	Temperatur-Baustein
0x00	nicht vorhanden
0x01	LM75
0x02	DS1621
0x03	MCP9808
0x04	TC74
0x05	LM35
0x06	DS18B20
0x07	TQS3 / TQS4
0x08	AM2315

Die Inhalte bzw. Rückgabewerte für die unterstützten Board-Editionen (Befehl [0x2D]) sind wie folgt:

ASCII-Wert	Dez. Wert	Hex. Wert	Bedeutung
B	66	0x42	Basic-Ausführung
P	80	0x50	Professional Ausführung
E	69	0x45	Enterprise Ausführung

Dieser Befehl ist im Zuge der Weiterentwicklung auf die Version 4.0 im Befehl [0x21] für die Hardware-Version mit integriert. Steht aber weiterhin als Einzelbefehl zur Verfügung.

5.2.7 I²C-Bus, SPI-Bus, One-Wire

Die Steuerung einer Echtzeituhr (RTC), eines Temperatur-Senors oder einer digitalen oder analogen Erweiterung sind inzwischen nichts außergewöhnliches mehr. Im Schwabenplan Protokoll Version 4.0 können Sie eine Vielzahl von Bauteilen anbinden und steuern. Hierfür bietet dieses Protokoll eine breite Auswahl an Möglichkeiten. Diese können auch erweitert werden.

Die Inhalte werden als Ganzzahl im zweistelligen Hex-Format übertragen. Diese Befehle sind im Wesentlichen zum Lesen und Schreiben gedacht. Bitte beachten Sie hierfür das zusätzliche Datenfeld R/W. Aufgrund der vielen verschiedenen Sensoren, die verfügbar sind, beschränken sich derzeit die Temperaturmessungen auf positive Werte, die als Ganzzahl ausgegeben werden.

Befehl	Bezeichnung	P0	P1	P2	P3
0x40	LM75 Temperatur Lesen	Adresse			
0x41	DS1621 Temperatur Lesen	Adresse			
0x43	MCP23008	Adresse			Port-Wert
0x44	MCP23017	Adresse		Port Nr. 1	Port Nr. 2
0x45	PCF8574	Adresse			Port-Wert
0x46	PCF8591	Adresse	Analog-Port Nr.		
0x50	Uhrzeit	Stunden	Minuten	Sekunden	
0x51	Datum und Wochentag	Tag	Monat	Jahr	Wochentag
0x52	Stunden	Stunden			
0x53	Minuten	Minuten			
0x54	Sekunden	Sekunden			
0x55	Tag	Tag			
0x56	Monat	Monat			
0x57	Jahr	Jahr			
0x58	Wochentag	Wochentag			

5.3 Digitale und Analoge PORT-BEFEHLE

Die Ansteuerung bzw. Abfrage von einzelnen Pins und Ports eines Microcontrollers sind wohl die häufigste Anwendung in der Steuerungstechnik. So auch hier. Unabhängig von Prozessor, Programmiersprache oder Entwicklungsumgebung können Sie mit nur einem Befehl den Port eines Microcontrollers einlesen oder ansteuern.

Befehl	Beschreibung	P0	P1	P2	P3
0xAD	A/D-Wandler	Adresse		ADW_High	ADW_Low
0xDD	Digitalbefehl	Port			Wert

Für die Steuerung eines Digital-Ports stehen folgende Ports zur Auswahl:

Befehl	PORT-Nummer	PORT-Name
0x00	0	PORTA
0x01	1	PORTB
0x02	2	PORTC
0x03	3	PORTD
0x04	4	PORTE
0x05	5	PORTF
0x06	6	PORTG
0x07	7	PORTH
0x08	8	PORTJ

5.4 Sammelruf (Broadcast Message)

Über ein RS-485 Schnittstelle oder CAN-Busschnittstelle können mehrere Teilnehmer miteinander verbunden und gesteuert werden. Hierzu wird die Teilnehmeradresse 0xFF bzw. 255 in Dezimal genutzt.

Ein Sammelruf (Broadcast-Message) wird von keinem Busteilnehmer bestätigt. In der Praxis wird dieser Befehl zum Aktualisieren der Uhrzeit, des Datum, einer Statusmeldung für alle Teilnehmer oder zur Durchführung eines Resets verwendet.

Wichtiger Hinweis:

a) Die Adresse eines Busteilnehmers kann mit dem Befehl 0x25 verwendet werden.

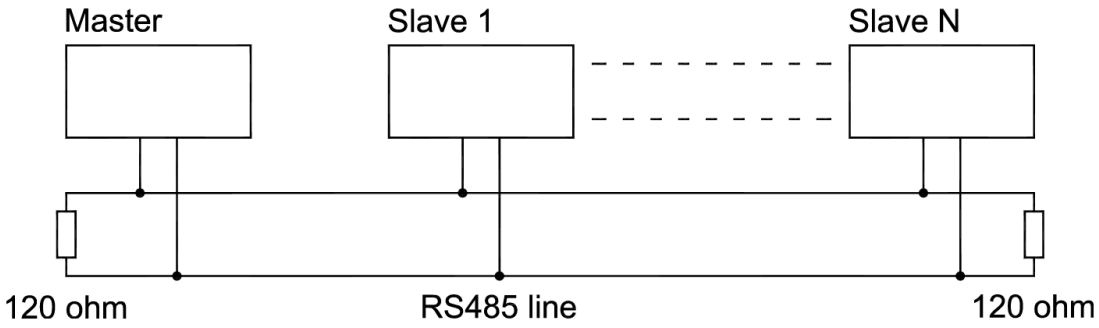
b) Achten Sie bitte darauf, dass Sie diesen Befehl nicht als Sammelruf absenden.

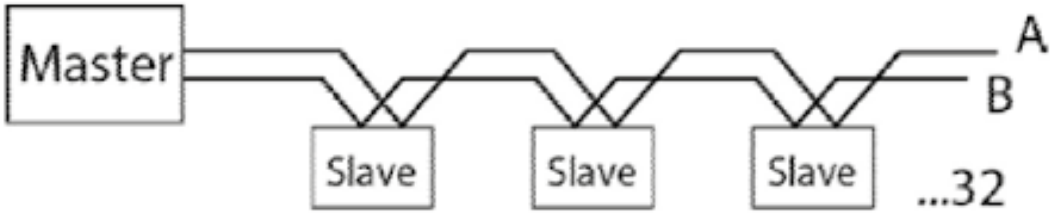
c) In diesem Fall hätten alle am Bus angeschlossenen Teilnehmer die gleiche Adresse.

6. Verkabelung

Bitte achten Sie bei einem RS-485 bzw. einen CAN-Bus Vernetzung auf eine korrekte Verkabelung. Die maximale Länge der Stichleitungen in so einem Bussystem sowie die korrekte Terminierung des Bussystems sind wichtig. Ohne die Abschlusswiderstände am Anfang und am Ende der Busverkabelung funktioniert der Bus nicht stabil.

Auf unseren Platinen können Sie diese über DIP-Schalter vornehmen. Die nachfolgenden Skizzen sollen Ihnen bei der Projektierung helfen.





7. Umwandlung von HEX zu Integer (Char -> Integer)

Die Daten zwischen den einzelnen Teilnehmern werden als String versendet. Für einen Computer können Sie mit Hilfe von unserer Schwabenplan DLL Datei die Daten entsprechend empfangen und umwandeln. In einem Microcontroller empfiehlt es sich die Daten als Ganzzahlen (Integer Werten) verarbeiten. Das nachfolgende Beispiel soll Ihnen die Arbeit während der Integration erleichtern.

```
//*****************************************************************************
// Hex to Integer
// Programmierer:    Ingo Schick, Fa. Schwabenplan
// Datum:                   19.12.2020
//*****************************************************************************

//*****************************************************************************
// Include-Dateien
//*****************************************************************************
#include <xc.h>
#include "Hex2Int.h"
//*****************************************************************************

//*****************************************************************************
unsigned short int Hex2Int(char* hex)
{

    unsigned short int val = 0;

    while (*hex) {
        // get current character then increment
        uint8_t byte = *hex++;
        // transform hex character to the 4bit equivalent number, using the ascii table
indexes
        if (byte >= '0' && byte <= '9') byte = byte - '0';
        else if (byte >= 'a' && byte <= 'f') byte = byte - 'a' + 10;
        else if (byte >= 'A' && byte <= 'F') byte = byte - 'A' + 10;
        // shift 4 to make space for new digit, and add the 4 bits of the new digit
        val = (val << 4) | (byte & 0xF);
    }
    return val;
}
//*****************************************************************************
```

8. Erweiterung auf das Ethernet Protokoll

Das Schwabenplan Protokoll 4.0 wurde im Zuge der Weiterentwicklung für den Microcontroller PIC18F97J60 auf Ethernet Übertragung abgeändert. Die Treiber in der Software für den PC und der Firmware des Microcontroller selbst erfordern keine Kennzeichnung mehr das Senden, Empfangen, Empfänger, Absender oder Steuerzeichen. Diese werden automatisch über die Layer (Befehlsstruktur) des TCP-Protokolls hinzugefügt bzw. entfernt. Daher wurden diese Felder für die Ethernet Datenübertragung entfernt, und die einzelnen Befehle vereinfacht.

Mit dem PIC18F97J60 können die Daten auch per Ethernet über das TCP-Protokoll und einen vordefinierten Port zwischen Server und Client übertragen werden. Der Controller antwortet nur an einen vordefinierten Server im Netzwerk.

In den Clients (PIC18F97J60) werden eine manuelle IP-Adresse, die Subnetzmaske, Gateway usw. manuell hinterlegt. Der PIC18F97J60 verfügt über ein internes 10Mbit Ethernet Interface, welches Halbduplex arbeitet. Die MAC-Adresse bezieht der PIC18F97J60 aus einem externen Speicherbaustein, welche von Microchip einprogrammiert worden ist. Diese kann auch nicht verändert oder kopiert werden. Somit ist sichergestellt, dass es diese MAC-Adresse nur einmal weltweit gibt. Die Hardware arbeitet an jedem 10/100/1000 Mbit Ethernet-Netzwerk. IM PIC18F97J60 ist eine Autonegotiation (Auto-Sensing) implementiert. Ein Router, Switch usw. erkennen das den Controller automatisch.

Auf der Hauptplatine befindet sich zusätzlich eine Echtzeituhr (RTC) und ein Temperatur-Baustein und eine RS-232 oder RS-485 Schnittstelle.

Die Parameter für die seriellen Schnittstellen und den I²C- bzw. den SPI-Bus sind im Quellcode und über die externen Schnittstellen frei konfigurierbar.

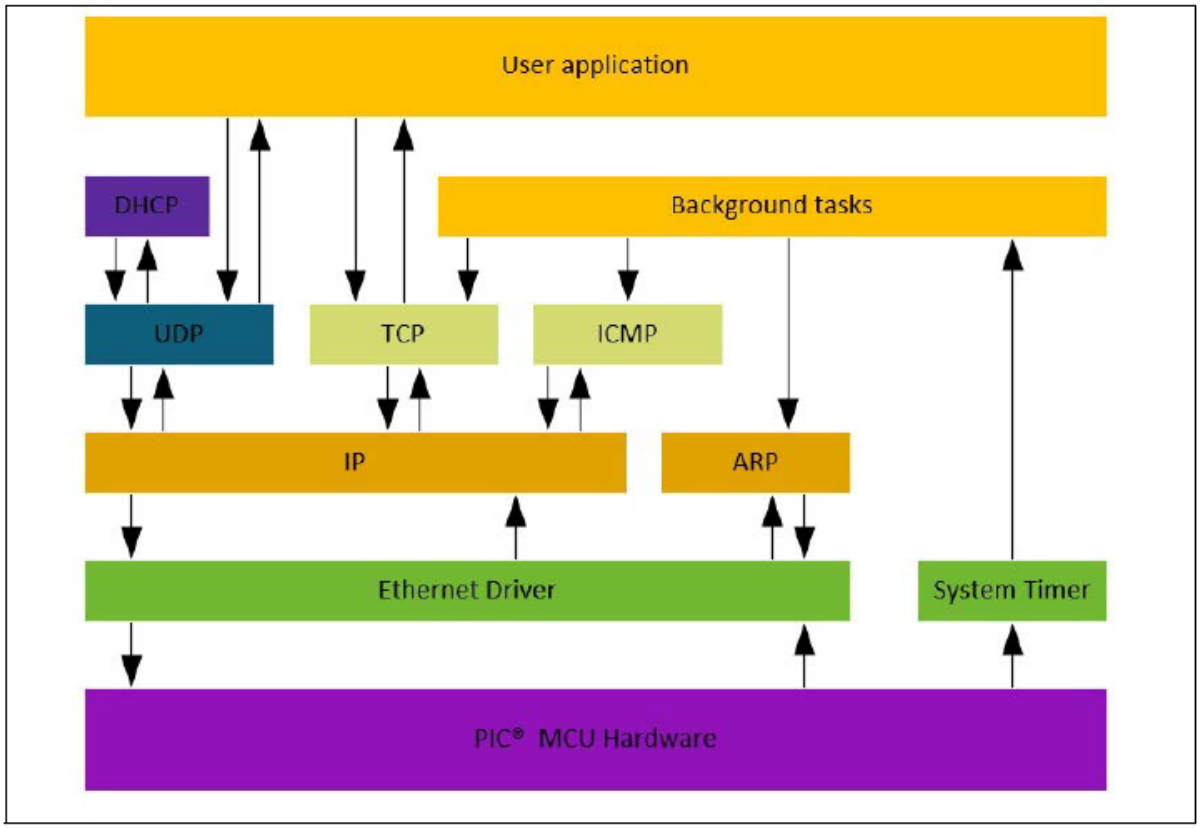
Die Firma Microchip hat hierfür drei MAC-Adressbereiche für sich reserviert. Diese sind:

- 00:04:A3:
- 54:10:EC:
- 80:1F:12:

PIC18F97J60-1.fritz.box	192.168.100.20	Microchip Technology Inc.	80:1F:12:D4:0E:5A
PIC18F97J60-2.fritz.box	192.168.100.21	Microchip Technology Inc.	80:1F:12:D4:0C:8A
PIC18F97J60-3.fritz.box	192.168.100.25	Microchip Technology Inc.	54:10:EC:E4:98:60
PIC18F97J60-4.fritz.box	192.168.100.26	Microchip Technology Inc.	54:10:EC:E4:9A:01
PIC18F97J60-5.fritz.box	192.168.100.27	Microchip Technology Inc.	54:10:EC:E4:99:9E
PIC18F97J60-6.fritz.box	192.168.100.28	Microchip Technology Inc.	54:10:EC:E4:9A:7F
PIC18F97J60-7.fritz.box	192.168.100.29	Microchip Technology Inc.	54:10:EC:E4:DB:40
PIC18F97J60-8.fritz.box	192.168.100.30	Microchip Technology Inc.	54:10:EC:E4:9A:DD

Der TCPIP-Stack stammt von Microchip selbst (AN1921), und wurde auf die Hardwareversion 1.2 und 2.0 angepasst. Die Kommunikation der einzelnen Netzwerkteilnehmer erfolgt über IPv4. Sämtliche Zusatzplatinen können eingesetzt werden.

FIGURE 1: MULTILAYER TCP/IP COMMUNICATION MODEL



Befehlsaufbau

Alle Befehle werden im hexadezimalen Format als String übertragen. Der Doppelpunkt dient als Trennzeichen. Der letzte Parameter teilt dem Controller mit, ob es sich um ein Lese- oder Schreibbefehl handelt. Andere Steuerzeichen werden nicht benötigt. Dies wird von den Netzwerkteilenehmern selbst erledigt.

CMD : P0 : P1 : P2 : P3 : P4 : P5 : P6 : P7 : R/W

P0 bis P7 -> Parameter für die Übergabe bzw. Rückgabe

R/W -> Parameter für das Lesen oder Schreiben eines Befehles

Befehlsart	Hex. Wert
Schreiben	0x00
Lesen	0x01

Beispiel Nr. 1: 20:00:00:00:00:00:00:00:00:01
Beispiel Nr. 2: DD:08:00:00:FF:00:00:00:00:00

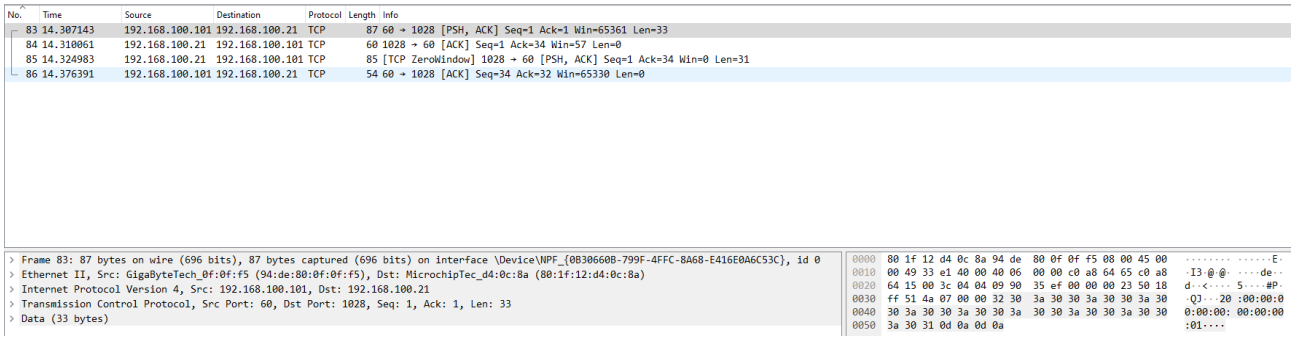
Das Protokoll basiert auf dem Schwabenplan Protokoll 4.0, und kann mit sehr wenig Aufwand in eine eigene Anwendung implementiert werden. Der Header für die serielle Schnittstelle kann für Ethernet in entfallen. Die Aushandlung der Teilnehmeradresse/n erfolgt direkt in den Ethernet Treibern der jeweiligen Teilnehmer im Netzwerk.

Eine einfache und unkomplizierte Hausautomatisierung ist über Ethernet in beliebiger Größe möglich. Eine einfache VB.net Applikation inkl. Einer eigenen DLL-Datei mit vielen Funktionen steht ebenfalls zur Verfügung.

Sämtliche Busaktivitäten können über eine optional zuschaltbare Protokoll-Datei mitgeschrieben werden.

```
14.03.2024 20:58:35 Protokoll-Datei ON
14.03.2024 20:58:35 DEBUG-Mode ON
14.03.2024 20:58:35 Broadcast-Message ON
14.03.2024 20:58:35 Der Server 192.168.100.101 ist gestartet.
14.03.2024 20:58:35 Der Client 192.168.100.26 ist verbunden.
14.03.2024 20:58:35 Der Client 192.168.100.30 ist verbunden.
14.03.2024 20:58:35 Der Client 192.168.100.21 ist verbunden.
14.03.2024 20:58:35 Der Client 192.168.100.20 ist verbunden.
14.03.2024 20:58:35 Der Client 192.168.100.29 ist verbunden.
14.03.2024 20:58:36 Der Client 192.168.100.27 ist verbunden.
14.03.2024 20:58:36 Der Client 192.168.100.28 ist verbunden.
14.03.2024 20:58:36 Der Client 192.168.100.25 ist verbunden.
```

Der Datenverkehr kann ebenfalls mit dem Tool Wireshark analysiert werden.



Vorteile von TCP

Das Transmission Control Protocol (TCP) ist das Protokoll für maximale Zuverlässigkeit und Qualität. Es ist vielleicht nicht das schnellste, aber es erledigt seine Arbeit ordentlich. Hier finden Sie ein paar typische Beispiele:

- ✓ Es baut eine Verbindung zwischen Sender und Empfänger auf und hält sie aufrecht.
- ✓ Es arbeitet unabhängig vom Betriebssystem.
- ✓ Es unterstützt viele Routing-Protokolle.
- ✓ Es prüft auf Fehler und garantiert, dass die Daten unverändert am Ziel ankommen.
- ✓ Es bestätigt den Eingang der Daten nach der Zustellung bzw. versucht, sie erneut zu übertragen.
- ✓ Es ist in der Lage, Daten in einer bestimmten Reihenfolge zu senden.
- ✓ Es optimiert die Geschwindigkeit der Datenübertragung in Abhängigkeit vom Empfänger.
- ✓ Trotz seiner geringeren Geschwindigkeit ist TCP das einzige Protokoll, das verlorene Datenpakete erneut übertragen kann. Wenn Zuverlässigkeit entscheidend ist, ist TCP die beste Option.

9. Ergänzungen ASCII-Zeichensatz

Zahlen

ASCII-Zeichen	Wert Hex.	Wert Dez.
0	0x30	48
1	0x31	49
2	0x32	50
3	0x33	51
4	0x34	52
5	0x35	53
6	0x36	54
7	0x37	55
8	0x38	56
9	0x39	57

Beliebte Trenn- und Endzeichen

ASCII-Zeichen	Wert Hex.	Wert Dez.
:	0x3A	58
;	0x3B	59

Großbuchstaben

ASCII-Zeichen	Wert Hex.	Wert Dez.
A	0x41	65
B	0x42	66
C	0x43	67
D	0x44	68
E	0x45	69
F	0x46	70
G	0x47	71
H	0x48	72
I	0x49	73
J	0x4A	74
K	0x4B	75
L	0x4C	76
M	0x4D	77
N	0x4E	78
O	0x4F	79
P	0x50	80
Q	0x51	81
R	0x52	82
S	0x53	83
T	0x54	84
U	0x55	85
V	0x56	86
W	0x57	87
X	0x58	88
Y	0x59	89
Z	0x5A	90

Kleinbuchstaben

ASCII-Zeichen	Wert Hex.	Wert Dez.
a	0x61	97
b	0x62	98
c	0x63	99
d	0x64	100
e	0x65	101
f	0x66	102
g	0x67	103
h	0x68	104
i	0x69	105
j	0x6A	106
k	0x6B	107
l	0x6C	108
m	0x6D	109
n	0x6E	110
o	0x6F	111
p	0x70	112
q	0x71	113
r	0x72	114
s	0x73	115
t	0x74	116
u	0x75	117
v	0x76	118
w	0x77	119
x	0x78	120
y	0x79	121
z	0x7A	122

Steuerzeichen

ASCII-Zeichen	Wert Hex.	Wert Dez.	Bedeutung	Kurzzeichen
NUL	0x0	0	(Null)	
SOH	0x1	1	Start of Header	
STX	0x2	2	Start of Text	
ETX	0x3	3	End of Text	
EOT	0x4	4	End of Transmit	
ENQ	0x5	5	Enquiry	
ACK	0x6	6	Acknowledge	
BEL	0x7	7	Bell	
BS	0x8	8	Backspace	
HT	0x9	9	Horizontal Tab	
LF	0xA	10	Line Feed	\n
VT	0xB	11	Vertical Tab	
FF	0xC	12	Form Feed	
CR	0xD	13	Carriage Return	\r
SO	0xE	14	Shift Out	
SI	0xF	15	Shift In	
DLE	0x10	16	Data Line Escape	
DC1	0x11	17	Device Control 1	
DC2	0x12	18	Device Control 2	
DC3	0x13	19	Device Control 3	
DC4	0x14	20	Device Control 4	
NAK	0x15	21	Negative Acknowledge	
SYN	0x16	22	Synchronous Idle	
ETB	0x17	23	End of Transmit Block	
CAN	0x18	24	Cancel	
EM	0x19	25	End of Medium	
SUB	0x1A	26	Substitute	
ESC	0x1B	27	Escape	
FS	0x1C	28	File Separator	
GS	0x1D	29	Group Separator	
RS	0x1E	30	Record Separator	
US	0x1F	31	Unit Separator	
SP	0x20	32	Space	

Schwabenplan Protokoll Version 4.0 - Befehlsübersicht

Befehl Hex.	Befehl Dez.	Beschreibung	P0	P1	P2	P3	P4	P5	P6	P7
0x20	32	MC-Typ	Integer							
0x21	33	Hardware Version	Integer	Integer						
0x22	34	Taktfrequenz	Integer							
0x23	35	Software Version	Integer							
0x24	36	Protokoll Version	Integer							
0x25	37	RS-485 Adresse	Integer							
0x26	38	Bootloader Modus	Integer							
0x27	39	Debug Modus	Integer							
0x28	40	Interface on Board	Integer	Integer						
0x29	41	Baudrate	Integer							
0x2A	42	I²C-Geschwindigkeit	Integer							
0x2B	43	RTC-Baustein	Integer							
0x2C	44	Temperatur-Baustein	Integer							
0x2D	45									
0x30	48	IP-Adresse	IP0	IP1	IP2	IP3				
0x31	49	Subnet Maske	SN0	SN1	SN2	SN3				
0x32	50	Router	ROU0	ROU1	ROU2	ROU3				
0x33	51	Gateway	GW0	GW1	GW2	GW3				
0x34	52	DNS-Server	DNS0	DNS0	DNS0	DNS0				
0x35	53	Server IP	SVR0	SVR0	SVR0	SVR0				
0x36	54	Kommunikations-Port	PO_High	PO_Low						
0x37	55	MAC-Adresse	MAC0	MAC1	MAC2	MAC3	MAC4	MAC5		
0x40	64	LM75 Temperatur Lesen	Adresse							
0x41	65	DS1621 Temperatur Lesen	Adresse							
0x43	66	MCP23008	Integer							
0x44	67	MCP23017	Integer							
0x45	68	PCF8574	Integer							
0x46	69	PCF8591	Integer							
0x50	80	Uhrzeit	Stunden	Minuten	Sekunden					
0x51	81	Datum und Wochentag	Tag	Monat	Jahr	Wochentag				
0x52	82	Stunden	Stunden							
0x53	83	Minuten	Minuten							
0x54	84	Sekunden	Sekunden							
0x55	85	Tag	Tag							
0x56	86	Monat	Monat							
0x57	87	Jahr	Jahr							
0x58	88	Wochentag	Wochentag							
0xAD	173	A/D-Wandler	Adresse		ADW_High	ADW_Low				
0xDD	221	Digitalbefehl	Port			Wert				
0xFF	255	Reset								

ASCII-Tabelle

PC-Zeichen	Controlcode Windows-Zeichen	dezimal
	oktal	hexadezimal
		binär

NUL		^@	000d	SP		032d	@	@	064d	`	`	096d	Ç	Ç	128d	á	á	160d	└	└	192d	α	Ó	224d
		000o	00h		040o	20h		100o	40h		140o	60h		200o	80h		240o	A0h		300o	C0h		340o	E0h
		0000 0000b			0010 0000b			0100 0000b			0110 0000b			1000 0000b			1010 0000b			1100 0000b			1110 0000b	
SOH	☺	^A	001d	!	!	033d	A	A	065d	a	a	097d	ü	ü	129d	í	í	161d	┴	┴	193d	ß	ß	225d
		001o	01h		041o	21		101o	41h		141o	61h		201o	81h		241o	A1h		301o	C1h		341o	E1h
		0000 0001b			0010 0001b			0100 0001b			0110 0001b			1000 0001b			1010 0001b			1100 0001b			1110 0001b	
STX	☻	^B	002d	"	"	034d	B	B	06&d	b	b	098d	é	é	130d	ó	ó	162d	┬	┬	194d	β	Ô	226d
		002o	02h		042o	22h		102o	42h		142o	62h		202o	82h		242o	A2h		302o	C2h		342o	E2h
		0000 0010b			0010 0010b			0100 0010b			0110 0010b			1000 0010b			1010 0010b			1100 0010b			1110 0010b	
ETX	♥	^C	003d	#	#	035d	C	C	067d	c	c	099d	â	â	131d	ú	ú	163d	├	├	195d	π	Ò	227d
		003o	03h		043o	23h		103o	43h		143o	63h		203o	83h		243o	A3h		303o	C3h		343o	E3h
		0000 0011b			0010 0011b			0100 0011b			0110 0011b			1000 0011b			1010 0011b			1100 0011b			1110 0011b	
EOT	♦	^D	004d	$	$	036d	D	D	068d	d	d	100d	ä	ä	132d	ñ	ñ	164d	─	─	196d	õ	õ	228d
		004o	04h		044o	24h		104o	44h		144o	64h		204o	84h		244o	A4h		304o	C4h		344o	E4h
		0000 0100b			0010 0100b			0100 0100b			0110 0100b			1000 0100b			1010 0100b			1100 0100b			1110 0100b	
ENQ	♣	^E	005d	%	%	037d	E	E	069d	e	e	101d	à	à	133d	Ñ	Ñ	165d	┼	┼	197d	σ	Õ	229d
		005o	05h		045o	25h		105o	45h		145o	65h		205o	85h		245o	A5h		305o	C5h		345o	E5h
		0000 0101b			0010 0101b			0100 0101b			0110 0101b			1000 0101b			1010 0101b			1100 0101b			1110 0101b	
ACK	♠	^F	006d	&	&	038d	F	F	070d	f	f	102d	å	å	134d	ª	ª	166d	╞	╞	198d	µ	µ	230d
		006o	06h		046o	26h		106o	46h		146o	66h		206o	86h		246o	A6h		306o	C6h		346o	E6h
		0000 0110b			0010 0110b			0100 0110b			0110 0110b			1000 0110b			1010 0110b			1100 0110b			1110 0110b	
BEL	•	^G	007d	'	'	039d	G	G	071d	g	g	103d	ç	ç	135d	º	º	167d	╟	╟	199d	þ	Þ	231d
		007o	07h		047o	27h		107o	47h		147o	67h		207o	87h		247o	A7h		307o	C7h		347o	E7h
		0000 1111b			0010 1111b			0100 1111b			0110 1111b			1000 1111b			1010 1111b			1100 1111b			1110 1111b	
BS	◘	^H	008d	(	(	040d	H	H	072d	h	h	104d	ê	ê	136d	¿	¿	168d	╙	╙	200d	Φ	Þ	232d
		010o	08h		050o	28h		110o	48h		150o	68h		210o	88h		250o	A8h		310o	C8h		350o	E8h
		0000 1000b			0010 1000b			0100 1000b			0110 1000b			1000 1000b			1010 1000b			1100 1000b			1110 1000b	
HT	○	^I	009d	)	)	041d	I	I	073d	i	i	105d	ë	ë	137d	⌐	®	169d	╘	╘	201d	θ	Ú	233d
		011o	09h		051o	29h		111o	49h		151o	69h		211o	89h		251o	A9h		311o	C9h		351o	E9h
		0000 1001b			0010 1001b			0100 1001b			0110 1001b			1000 1001b			1010 1001b			1100 1001b			1110 1001b	
LF	■	^J	010d	*	*	042d	J	J	074d	j	j	106d	è	è	138d	¬	¬	170d	╓	╓	202d	Ω	Û	234d
		012o	0Ah		052o	2Ah		112o	4Ah		152o	6Ah		212o	8Ah		252o	AAh		312o	CAh		352o	EAh
		0000 1010b			0010 1010b			0100 1010b			0110 1010b			1000 1010b			1010 1010b			1100 1010b			1110 1010b	
VT	♂	^K	011d	+	+	043d	K	K	075d	k	k	107d	ï	ï	139d	½	½	171d	╫	╫	203d	δ	Ù	235d
		0013o	0Bh		053o	2Bh		113o	4Bh		153o	6Bh		213o	8Bh		253o	ABh		313o	CBh		353o	EBh
		0000 1011b			0010 1011b			0100 1011b			0110 1011b			1000 1011b			1010 1011b			1100 1011b			1110 1011b	
FF	♀	^L	012d	,	,	44d	L	L	076d	l	l	108d	î	î	140d	¼	¼	172d	╪	╪	204d	∞	ý	236d
		014o	0Ch		054o	2Ch		114o	4Ch		154o	6Ch		214o	8Ch		254o	ACh		314o	CCh		354o	ECh
		0000 1100b			0010 1100b			0100 1100b			0110 1100b			1000 1100b			1010 1100b			1100 1100b			1110 1100b	
CR	♪	^M	013d	-	-	045d	M	M	077d	m	m	109d	ì	ì	141d	¡	¡	173d	═	═	205d	∅	Ý	237d
		0015o	0Dh		055o	2Dh		115o	4Dh		155o	6Dh		215o	8Dh		255o	ADh		315o	CDh		355o	EDh
		0000 1101b			0010 1101b			0100 1101b			0110 1101b			1000 1101b			1010 1101b			1100 1101b			1110 1101b	
SO	♫	^N	014d	.	.	046d	N	N	078d	n	n	110d	Ä	Ä	142d	«	«	174d	╬	╬	206d	∈		238d
		016o	0Eh		056o	2Eh		116o	4Eh		156o	6Eh		216o	8Eh		256o	AEh		316o	CEh		356o	EEh
		0000 1110b			0010 1110b			0100 1110b			0110 1110b			1000 1110b			1010 1110b			1100 1110b			1110 1110b	
SI	☼	^O	015d	/	/	047d	O	O	079d	o	o	111d	Å	Å	143d	»	»	175d	╧	╧	207d	∩	˙	239d
		017o	0Fh		057o	2Fh		117o	4Fh		157o	6Fh		217o	8Fh		257o	AFh		317o	CFh		357o	EFh
		0000 1111b			0010 1111b			0100 1111b			0110 1111b			1000 1111b			1010 1111b			1100 1111b			1110 1111b	
DLE	►	^P	016d		0	048d	P	P	080d	p	p	112d	É	É	144d	▒	▒	176d	╨	╨	208d	≡	-	240d
		020o	10h		060o	30h		120o	50h		140o	70h		220o	90h		260o	B0h		320o	D0h		360o	F0h
		0001 0000b			0011 0000b			0101 0000b			0111 0000b			1001 0000b			1011 0000b			1101 0000b			1111 0000b	
DC1	◄	^Q	017d	1	1	049d	Q	Q	081d	q	q	113d	æ	æ	145d	▓	▒	177d	╤	╤	Ð	209d	±	±	241d
		021o	11h		061o	31h		121o	51h		141o	71h		221o	91h		261o	B1h		321o	D1h		361o	F1h
		0001 0001b			0011 0001b			0101 0001b			0111 0001b			1001 0001b			1011 0001b			1101 0001b			1111 0001b	
DC2	↕	^R	018d	2	2	050d	R	R	082d	r	r	114d	Æ	Æ	146d	▓	▓	178d	╥	╥	Ê	210d	≥	‗	242d
		022o	12h		062o	32h		122o	52h		142o	72h		222o	92h		262o	B2h		322o	D2h		362o	F2h
		0001 0010b			0011 0010b			0101 0010b			0111 0010b			1001 0010b			1011 0010b			1101 0010b			1111 0010b	
DC3	‼	^S	019d	3	3	051d	S	S	083d	s	s	115d	ô	ô	147d	│	│	179d	╙	╙	Ë	211d	¾	¾	243d
		023o	13h		063o	33h		123o	53h		143o	73h		223o	93h		263o	B3h		323o	D3h		363o	F3h
		0001 0011b			0011 0011b			0101 0011b			0111 0011b			1001 0011b			1011 0011b			1101 0011b			1111 0011b	
DC4	¶	^T	020d	4	4	052d	T	T	084d	t	t	116d	ö	ö	148d	┤	┤	180d	╘	╘	È	212d	¶	¶	244d
		024o	14h		064o	34h		124o	54h		144o	74h		224o	94h		264o	B4h		324o	D4h		364o	F4h
		0001 0100b			0011 0100b			0101 0100b			0111 0100b			1001 0100b			1011 0100b			1101 0100b			1111 0100b	
NAK	§	^U	021d	5	5	053d	U	U	085d	u	u	117d	ò	ò	149d	Á	Á	181d	╡	╡	ı	213d	§	§	245d
		025o	15h		065o	35h		125o	55h		145o	75h		225o	95h		265o	B5h		325o	D5h		365o	F5h
		0001 0101b			0011 0101b			0101 0101b			0111 0101b			1001 0101b			1011 0101b			1101 0101b			1111 0101b	
SYN	■	^V	022d	6	6	054d	V	V	086d	v	v	118d	û	û	150d	Â	Â	182d	╢	╢	Í	214d	÷	÷	246d
		026o	16h		066o	36h		126o	56h		146o	76h		226o	96h		266o	B6h		326o	D6h		366o	F6h
		0001 0110b			0011 0110b			0101 0110b			0111 0110b			1001 0110b			1011 0110b			1101 0110b			1111 0110b	
ETB	↨	^W	023d	7	7	055d	W	W	087d	w	w	119d	ù	ù	151d	À	À	183d	╖	╖	Î	215d	≈	°	247d
		027o	17h		067o	37h		127o	57h		147o	77h		227o	97h		267o	B7h		327o	D7h		367o	F7h
		0001 0111b			0011 0111b			0101 0111b			0111 0111b			1001 0111b			1011 0111b			1101 0111b			1111 0111b	
CAN	↑	^X	024d	8	8	056d	X	X	088d	x	x	120d	ÿ	ÿ	152d	©	©	184d	╕	╕	Ï	216d	°	°	248d
		030o	18h		070o	38h		130o	58h		150o	78h		230o	98h		270o	B8h		330o	D8h		370o	F8h
		0001 1000b			0011 1000b			0101 1000b			0111 1000b			1001 1000b			1011 1000b			1101 1000b			1111 1000b	
EM	↓	^Y	025d	9	9	057d	Y	Y	089d	y	y	121d	Ö	Ö	153d	╣	╣	185d	┘	┘	╛	217d	∙	¨	249d
		031o	19h		071o	39h		131o	59h		151o	79h		231o	99h		271o	B9h		331o	D9h		371o	F9h
		0001 1001b			0011 1001b			0101 1001b			0111 1001b			1001 1001b			1011 1001b			1101 1001b			1111 1001b	
SUB	→	^Z	026d	:	:	058d	Z	Z	090d	z	z	122d	Ü	Ü	154d	╓	╓	186d	┌	┌	┌	218d	·	˙	250d
		032o	1Ah		072o	3Ah		132o	5Ah		152o	7Ah		232o	9Ah		272o	BAh		332o	DAh		372o	FAh
		0001 1010b			0011 1010b			0101 1010b			0111 1010b			1001 1010b			1011 1010b			1101 1010b			1111 1010b	
ESC	←	^[	027d	;	;	059d	[	[	091d	{	{	123d	¢	ø	155d	╖	╖	187d	┐	┐	▌	219d	√	¹	251d
		033o	1Bh		073o	3Bh		133o	5Bh		153o	7Bh		233o	9Bh		273o	BBh		333o	DBh		373o	FBh
		0001 1011b			0011 1011b			0101 1011b			0111 1011b			1001 1011b			1011 1011b			1101 1011b			1111 1011b	
FS	L	^\	028d	<	<	060d	\	\	092d	|	|	124d	£	£	156d	╜	╜	188d	└	└	▄	220d	ⁿ	³	252d
		034o	1Ch		074o	3Ch		134o	5Ch		154o	7Ch		234o	9Ch		274o	BCh		334o	DCh		374o	FCh
		0001 1100b			0011 1100b			0101 1100b			0111 1100b			1001 1100b			1011 1100b			1101 1100b			1111 1100b	
GS	↔	^]	029d	=	=	061d	]	]	093d	}	}	125d	¥	Ø	157d	╜	¢	189d	┴	┴	▐	221d	²	²	253d
		035o	1Dh		075o	3Dh		135o	5Dh		155o	7Dh		235o	9Dh		275o	BDh		335o	DDh		375o	FDh
		0001 1101b			0011 1101b			0101 1101b			0111 1101b			1001 1101b			1011 1101b			1101 1101b			1111 1101b	
RS	▲	^^	030d	>	>	062d	^	^	094d	~	~	126d	₧	×	158d	╛	¥	190d	┬	┬	▐	222d	■	▪	254d
		036o	1Eh		076o	3Eh		136o	5Eh		156o	7Eh		236o	9Eh		276o	BEh		336o	DEh		376o	FEh
		0001 1110b			0011 1110b			0101 1110b			0111 1110b			1001 1110b			1011 1110b			1101 1110b			1111 1110b	
US	▼	^_	031d	?	?	063d	_	_	095d	⌂	⌂	127d	ƒ	ƒ	159d	┐	┐	191d	■	■	▀	223d			255d
		037o	1Fh		077o	3Fh		137o	5Fh		157o	7Fh		237o	9Fh		277o	BFh		337o	DFh		377o	FFh
		0001 1111b			0011 1111b			0101 1111b			0111 1111b			1001 1111b			1011 1111b			1101 1111b			1111 1111b	

UL - Null prompt	EOT - End of transmission	BS - Backspace	FF - Form feed	DLE - Data link escape	DC4 -	CAN - Cancel	FS - File separator
SOH - Start of heading	ENQ - Enquiry	HT - Horizontal tab	CR - Carriage return	DC1 - X-ON	NAK - No acknowledge	EM - End of medium	GS - Group separator
STX - Start of text	ACK - Acknowledge	LF - Line feed	SO - Shift out	DC2 -	SYN - Synchronous idle	SUB - Substitute	RS - Record separator
ETX - End of Text	BEL - Bell	VT - Vertical tab	SI - Shift in	DC3 - X-Off	ETB - End transmission blocks	ESC - Escape	US - Unit separator

Einleitung

Der ASCII-Code ist eine standardisierte Zeichen-Definition für den Datenaustausch zwischen Informatik-Systemen. ASCII bedeutet American Standard Code for Information Interchange. Es ist sozusagen «der kleinste gemeinsame Nenner» aller Informatik-Systeme. ASCII wurde 1968 mit der Intention oder Absicht entwickelt, Datenübertragungen zwischen divergierenden Hardware- und Softwaresystemen zu standardisieren.

Standard ASCII-Code

Alle Zeichen im Bereich von 0 bis 127 gehören zum Standard ASCII-Zeichensatz. Dieser Standardzeichensatz von 0 bis 127 ist in der IT-Welt stark normiert. Die Zeichen werden aus einem 1 Byte dargestellt, das heisst es sind maximal 256 Zeichen möglich, auf PC-Systemen kommt der erweiterte ASCII-Zeichensatz (0 bis 255) zum Einsatz.

Die ASCII-Zeichen 0 bis 32 werden als Steuerzeichen bzw. Formatierungszeichen für Ein- und Ausgabegeräte verwendet.

Erweiterter ASCII-Code

Jeder Satz von Zeichen, der den ASCII-Werten zwischen dezimal 128 und 255 (hexadezimal 80 bis FF) zugeordnet ist. Die Zuordnung spezifischer Zeichen zu den erweiterten ASCII-Codes variieren zwischen Computern und zwischen Programmen, Schriften oder grafischen Zeichensätzen. Erweitertes ASCII erhöht auch die möglichen Funktionen, da sich mit den 128 zusätzlichen Zeichen z. B. Akzentbuchstaben, Umlaute, grafische Zeichen und spezielle Symbole darstellen lassen.

ASCII-Tabelle

000	NUL	033	!	066	B	099	c	132	ä	165	Ñ	198	ã	231	þ
001	Start Of Header	034	"	067	C	100	d	133	à	166	ª	199	Ã	232	Þ
002	Start Of Text	035	#	068	D	101	e	134	å	167	º	200	╚	233	Ú
003	End Of Text	036	$	069	E	102	f	135	ç	168	¿	201	╔	234	Û
004	End Of Transmission	037	%	070	F	103	g	136	ê	169	®	202	╩	235	Ù
005	Enquiry	038	&	071	G	104	h	137	ë	170	¬	203	╦	236	ý
006	Acknowledge	039		072	H	105	i	138	è	171	½	204	╠	237	Ý
007	Bell	040	(	073	I	106	j	139	ï	172	¼	205	═	238	¯
008	Backspace	041	)	074	J	107	k	140	î	173	¡	206	╬	239	´
009	Horizontal Tab	042	*	075	K	108	l	141	ì	174	«	207	¤	240	-
010	Line Feed	043	+	076	L	109	m	142	Ä	175	»	208	ð	241	±
011	Vertical Tab	044	,	077	M	110	n	143	Å	176	░	209	Ð	242	‗
012	Form Feed	045	-	078	N	111	o	144	É	177	▒	210	Ê	243	¾
013	Carriage Return	046	.	079	O	112	p	145	æ	178	▓	211	Ë	244	¶
014	Shift Out	047	/	080	P	113	q	146	Æ	179	│	212	È	245	§
015	Shift In	048	0	081	Q	114	r	147	ô	180	┤	213	ı	246	÷
016	Delete	049	1	082	R	115	s	148	ö	181	Á	214	Í	247	¸
017	-- frei --	050	2	083	S	116	t	149	ò	182	Â	215	Î	248	°
018	-- frei --	051	3	084	T	117	u	150	û	183	À	216	Ï	249	¨
019	-- frei --	052	4	085	U	118	v	151	ù	184	©	217	┘	250	·
020	-- frei --	053	5	086	V	119	w	152	ÿ	185	╣	218	┌	251	¹
021	Negative Acknowledge	054	6	087	W	120	x	153	Ö	186	║	219	█	252	³
022	Synchronous Idle	055	7	088	X	121	y	154	Ü	187	╗	220	▄	253	²
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	╝	221	¦	254	■
024	Cancel	057	9	090	Z	123	{	156	£	189	¢	222	Ì	255	
025	End Of Medium	058	:	091	[	124	|	157	Ø	190	¥	223	▀		
026	Substitude	059	;	092	\	125	}	158	×	191	┐	224	Ó		
027	Escape	060	<	093	]	126	~	159	ƒ	192	└	225	ß		
028	File Seperator	061	=	094	^	127	⌂	160	á	193	┴	226	Ô		
029	Group Seperator	062	>	095	_	128	Ç	161	í	194	┬	227	Ò		
030	Record Seperator	063	?	096	`	129	ü	162	ó	195	├	228	õ		
031	Unit Seperator	064	@	097	a	130	é	163	ú	196	─	229	Õ		
032		065	A	098	b	131	â	164	ñ	197	┼	230	µ		

Sonderzeichen der ASCII-Tabelle

In der ASCII-Tabelle (Grafik) sind verschiedene Sonderzeichen eingetragen die unter anderem zur Geräte-Steuerung verwendet werden.

Formatierungszeichen für Geräte

- **127 = BS:** Backspace, Rückschritt
- **009 = HT:** Horizontal Tab, Tabulator in Zeilenrichtung
- **010 = LF:** Line Feed, Zeilenwechsel
- **011 = VT:** Vertical Tab, Tabulator in Spaltenrichtung
- **012 = FF:** Form Feed, Blatt wird vom Drucker ausgegebene («Seitenvorschub»)
- **013 = CR:** Carriage Return, Wagenrücklauf oder Zeilenumbruch

Andere ASCII-Zeichen für Geräte

- **007 = BEL:** Bell, Signalzeichen
- **017 - 020 = DC1, DC2, DC3, DC4:** Keine ASCII-Verwendung, DC1 und DC3 werden oft von **XON und XOFF** und anderen **Software-Handshaking-Schematas** verwendet.

Steuerzeichen für die Datenübertragung

- **001 = SOH:** Start of Header
- **002 = STX:** Start of Text
- **003 = ETX:** End of Text
- **004 = EOT:** End of Transmission
- **005 = ENQ:** Enquiry
- **006 = ACK:** Acknowledge
- **021 = NAK:** Negative Acknowledge
- **022 = SYN:** Synchronous Idle
- **023 = ETB:** End of Transmission Block

Steuerzeichen für Geräte

- **000 = NUL:** Null
- **016 = DEL:** Delete
- **024 = CAN:** Cancel
- **025 = EM:** End of Medium
- **026 = SUB:** Substitute

Trennzeichen für Informationen

- **028 = FS:** File Separator
- **029 = GS:** Group Separator
- **030 = RS:** Record Separator
- **031 = US:** Unit Separator

Steuerzeichen für Code-Erweiterung

- **015 = SI:** Shift In
- **014 = SO:** Shift Out
- **027 = ESC:** Escape

Quelle: http://www.stefan-lenz.ch/glossareintrag_anzeigen.php?file=ascii.htm

Nicht druckbare Zeichen (Steuerzeichen)

Dez	Okt	Hex	Zeichen	Code	Bemerkung
0	000	0x00	Ctrl-@	NUL	Null prompt
1	001	0x01	Ctrl-A	SOH	Start of heading
2	002	0x02	Ctrl-B	STX	Start of text
3	003	0x03	Ctrl-C	ETX	End of Text
4	004	0x04	Ctrl-D	EOT	End of transmission
5	005	0x05	Ctrl-E	ENQ	Enquiry
6	006	0x06	Ctrl-F	ACK	Acknowledge
7	007	0x07	Ctrl-G	BEL	Bell
8	010	0x08	Ctrl-H	BS	Backspace
9	011	0x09	Ctrl-I	HT	Horizontal tab
10	012	0x0A	Ctrl-J	LF	Line feed
11	013	0x0B	Ctrl-K	VT	Vertical tab
12	014	0x0C	Ctrl-L	FF	Form feed
				NP	New page
13	015	0x0D	Ctrl-M	CR	Carriage return
14	016	0x0E	Ctrl-N	SO	Shift out
15	017	0x0F	Ctrl-O	SI	Shift in
16	020	0x10	Ctrl-P	DLE	Data link escape
17	021	0x11	Ctrl-Q	DC1	X-ON
18	022	0x12	Ctrl-R	DC2	
19	023	0x13	Ctrl-S	DC3	X-Off
20	024	0x14	Ctrl-T	DC4	
21	025	0x15	Ctrl-U	NAK	No achnowledge
22	026	0x16	Ctrl-V	SYN	Synchronous idle
23	027	0x17	Ctrl-W	ETB	End transmission blocks
24	030	0x18	Ctrl-X	CAN	Cancel
25	031	0x19	Ctrl-Y	EM	End of medium
26	032	0x1A	Ctrl-Z	SUB	Substitute
27	033	0x1B	Ctrl-[	ESC	Escape
28	034	0x1C	Ctrl-\	FS	File separator
29	035	0x1D	Ctrl-]	GS	Group separator
30	036	0x1E	Ctrl-^	RS	Record separator
31	027	0x1F	Ctrl-_	US	Unit separator
127	0177	0x7F		DEL	Delete or rubout

Druckbare Zeichen

Dez	Okt	Hex	Zeichen	Bemerkung
32	040	0x20		Leerzeichen
33	041	0x21	!	Ausrufungszeichen
34	042	0x22	"	Anführungszeichen
35	043	0x23	#	Doppelkreuz
36	044	0x24	$	Dollarzeichen
37	045	0x25	%	Prozentzeichen
38	046	0x26	&	Kaufmännisches UND
39	047	0x27	'	Apostroph
40	050	0x28	(	Runde Klammer auf
41	051	0x29	)	Runde Klammer zu
42	052	0x2A	*	Stern
43	053	0x2B	+	Pluszeichen
44	054	0x2C	,	Komma
45	055	0x2D	-	Minuszeichen
46	056	0x2E	.	Punkt
47	057	0x2F	/	Schrägstrich (Slash)
48	060	0x30	0	
49	061	0x31	1	
50	062	0x32	2	
51	063	0x33	3	
52	064	0x34	4	
53	065	0x35	5	
54	066	0x36	6	
55	067	0x37	7	
56	070	0x38	8	
57	071	0x39	9	
58	072	0x3A	:	Doppelpunkt
59	073	0x3B	;	Semikolon
60	074	0x3C	<	Kleiner-als-Zeichen
61	075	0x3D	=	Gleichheitszeichen
62	076	0x3E	>	Größer-als-Zeichen
63	077	0x3F	?	Fragezeichen
64	0100	0x40	@	Klammeraffe ("at")

65	0101	0x41	A	
66	0102	0x42	B	
67	0103	0x43	C	
68	0104	0x44	D	
69	0105	0x45	E	
70	0106	0x46	F	
71	0107	0x47	G	
72	0110	0x48	H	
73	0111	0x49	I	
74	0112	0x4A	J	
75	0113	0x4B	K	
76	0114	0x4C	L	
77	0115	0x4D	M	
78	0116	0x4E	N	
79	0117	0x4F	O	
80	0120	0x50	P	
81	0121	0x51	Q	
82	0122	0x52	R	
83	0123	0x53	S	
84	0124	0x54	T	
85	0125	0x55	U	
86	0126	0x56	V	
87	0127	0x57	W	
88	0130	0x58	X	
89	0131	0x59	Y	
90	0132	0x5A	Z	
91	0133	0x5B	[	Eckige Klammer auf
92	0134	0x5C	\	Umgekehrter Schrägstrich (Backslash)
93	0135	0x5D	]	Eckige Klammer zu
94	0136	0x5E	^	Caret (Hut)
95	0137	0x5F	_	Unterstrich
96	0140	0x60	`	"Back quote"
97	0141	0x61	a	
98	0142	0x62	b	
99	0143	0x63	c	
100	0144	0x64	d	

101	0145	0x65	e	
102	0146	0x66	f	
103	0147	0x67	g	
104	0150	0x68	h	
105	0151	0x69	i	
106	0152	0x6A	j	
107	0153	0x6B	k	
108	0154	0x6C	l	
109	0155	0x6D	m	
110	0156	0x6E	n	
111	0157	0x6F	o	
112	0160	0x70	p	
113	0161	0x71	q	
114	0162	0x72	r	
115	0163	0x73	s	
116	0164	0x74	t	
117	0165	0x75	u	
118	0166	0x76	v	
119	0167	0x77	w	
120	0170	0x78	x	
121	0171	0x79	y	
122	0172	0x7A	z	
123	0173	0x7B	{	
124	0174	0x7C	|	
125	0175	0x7D	}	
126	0176	0x7E	~	