

Computer-Schaltinterface CSI 7002 Teil 1

Der Nachfolger des erfolgreichen Computer-Schaltinterface CSI 7002 ermöglicht die PC-gesteuerte Realisierung von Schalt- und Steueraufgaben über 8 Schaltausgänge und 8 Schalteingänge. Nach der Programmierung des Interfaces über die serielle Schnittstelle des PC kann dieses durch den integrierten Mikroprozessor programmierte Schaltsequenzen automatisch und autark ausführen.

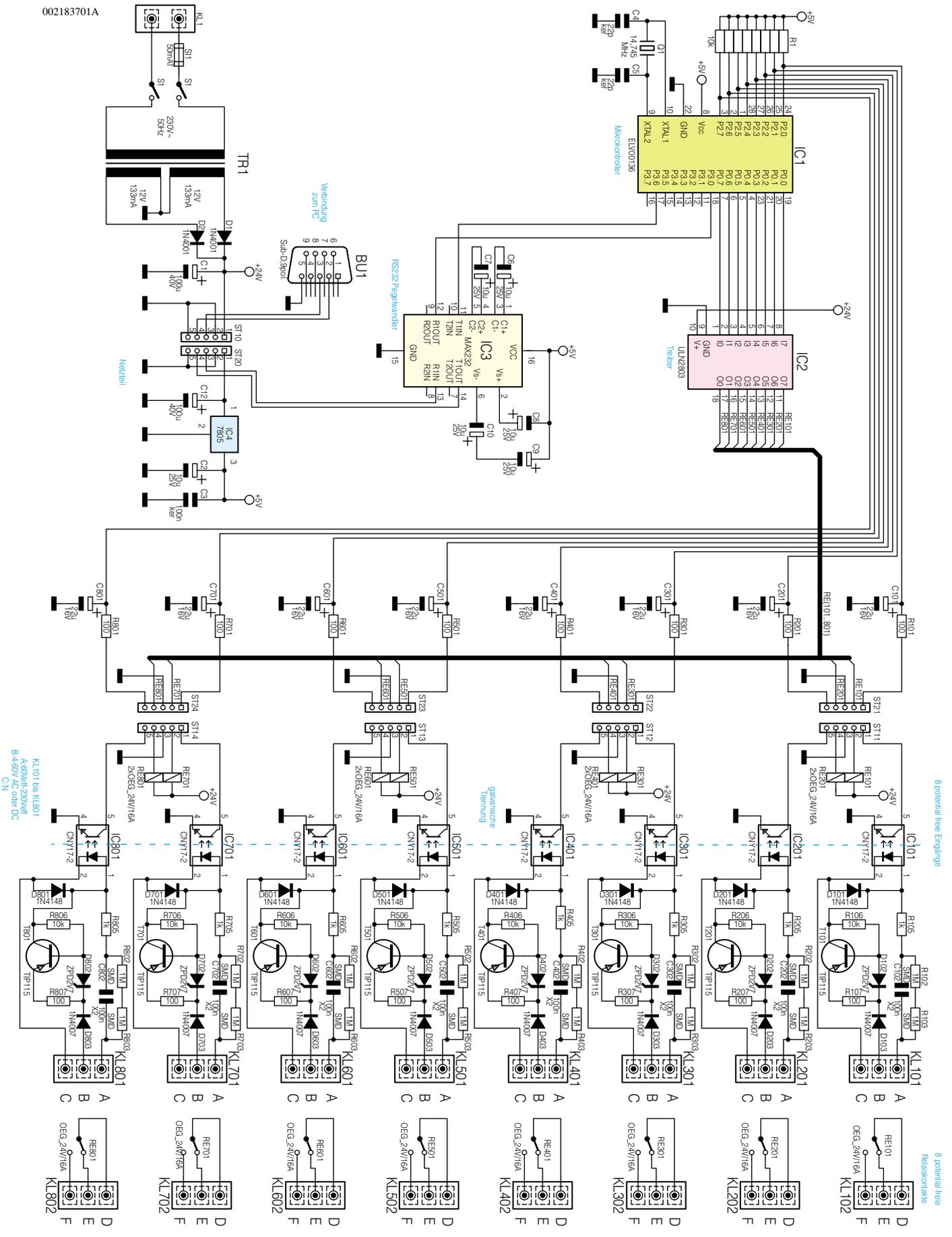
Universelles Interface

So universell ein moderner PC über seine Schnittstellen mit der Außenwelt kommunizieren kann, fehlt es jedoch zur realen Ausführung von Überwachungs- und Steuerungsaufgaben an einem Interface, das so-

wohl Lasten, auch Netzlasten, schalten als auch das Vorhandensein von Spannungen oberhalb des TTL-Pegels, also z. B. ebenfalls Netzspannung, abfragen kann.

Genau diese Aufgabe erfüllt das neue CSI 7002. Es bildet eine potentialgetrennte Schnittstelle zwischen Computer und den zu überwachenden und zu steuernden

Stromkreisen. Aber nicht nur das, das Interface ist dank des integrierten Mikroprozessors auch in der Lage, autark zu arbeiten und einmal vom PC aus eingeladene Bitmuster ständig wieder auszugeben. Als Beispiel kann hier die Bitmuster-Ausgabe für ein Lauflicht angeführt werden. Diese Bitmuster (Sequenzen) können bis zu 128



002183701A

8 potential free Engänge

8 potential free Relaiskontakte

Bild 1: Das Schaltbild des Interfaces

Startbyte 1 [SOH]	Nachricht 1-34 Nachricht	Längsparity 1 Längsparity	Endbyte 1 [EOT]
--------------------------------	---------------------------------------	--	------------------------------

Bild 2: Der Rahmen für die Datenübertragung

Byte lang sein, entweder am Programmieren stoppen oder automatisch von vorn beginnen.

Das Zeitintervall zum Weiterschalten innerhalb der Sequenz ist dabei zwischen 100 ms und 25,5 s einstellbar.

Damit sind nahezu beliebige eigene Steuer-Applikationen erstellbar. Mit dem Bausatz bzw. dem Fertiggerät wird eine Windows-Demo-Software geliefert, die das Abfragen der Ein- und Ausgänge, das Setzen der Ausgänge, das Erstellen von Sequenzen und die Datenübertragung zum Interface sowie den Start einer Sequenz erlaubt.

Damit ist eine sehr schnelle Einarbeitung in die Steuerung möglich, bevor man an das Erstellen eigener Programme geht.

Das Interface verfügt über acht potentialfreie Relais-Schaltausgänge, die als Umschalter ausgelegt sind und mit bis zu 16 A bei 230 V~ belastbar sind. Der Anschluss der Last erfolgt über Schraubklemmen.

Die Schalteingänge sind über Optokoppler potentialgetrennt und können entweder mit 4 bis 60 V AC/DC oder mit 60-230 V AC angesteuert werden.

Die Übertragung der Daten vom/zum PC erfolgt über eine serielle Datenverbindung entweder direkt mit einer freien seriellen Schnittstelle des PCs oder über den ELV-RS-232-Multiplexer, der den gleichzeitigen Anschluss mehrerer serieller Geräte an einen Com-Port des PCs erlaubt. Die hierzu nötige Software zur Portauswahl ist in die Demosoftware integriert.

Schaltung

Die zentrale Funktionsgruppe des Interfaces (Abbildung 1) ist der Mikroprozessor IC 1, der alle internen Abläufe steuert und den Pufferspeicher für die Ein- und Ausgabe bereitstellt. Seine Takterzeugung erfolgt mit Hilfe von Q 1 (14,745-MHz-Quarz), C 4 und C 5.

Da der verwendete Mikrocontroller genügend E/A-Ports bietet, kann der Anschluss der E/A-Interface-Baugruppen sehr einfach direkt erfolgen.

Der 8-Bit-Eingangs-Datenbus liegt so an den Ports P 2.0 bis 2.7, der ebenfalls 8 Bit breite Ausgangs-Datenbus an Port P 0.0 bis 0.7.

IC 3 stellt, in Standardbeschaltung ausgeführt, das RS-232-Interface dar, das über die 9-polige Sub-D-Buchse BU 1 die normgerechte RS-232-Verbindung zur seriellen Schnittstelle des PCs realisiert.

Als Treiber für die Leistungs-Schaltrelais RE 101 bis 801 fungiert IC 2, ein Open-Kollektor-Treiber-Schaltkreis.

Die Wechsler-Schaltkontakte der Relais liegen an den Schraubklemmen KL 102 bis KL 802.

An den Schraubklemmen KL 101 bis KL 801 werden die logischen Zustände der angeschlossenen Peripherie abgefragt.

Diese Schraubklemmen-Eingänge besitzen je drei Anschlüsse: A, B und C.

Der Steuerungsbereich ist, wie bereits erwähnt, in zwei Bereiche aufgeteilt. Eingangsgleich- und Wechselspannungen von 4 bis 60 V sind an die jeweiligen Klemmen B und C, Wechselspannungen ab 60 V bis 230 V an die Klemmen A und C anzuschließen.

Liegt z. B. am Kanal 1 (KL 101) eine Wechselspannung an „A“ an, so gelangt diese über die Vorwiderstands-/Kondensator-Kombination R 102/103/105/C 102 auf die Leuchtdiode des Optokopplers IC 101. Die o. g. RC-Kombination bestimmt zum einen den Strom durch die LED und begrenzt zum anderen Stromspitzen beim Einschalten. Die LED wird dabei im Halbwellenbetrieb angesteuert, während die antiparallel geschaltete Diode D101 jeweils die 2. Halbwelle „übernimmt“.

Eingangsspannungen von 4 bis 60 V AC/DC gelangen über die jeweilige Klemme „B“ auf den Gleichrichter D 103. Der daran anschließende Leistungstransistor T101 bildet zusammen mit R 106/R 107 und der Z-Diode D 102 eine Konstantstromquelle, die den Strom durch die LED im Optokoppler auf ca. 20 mA begrenzt.

Der jeweilige Anschluss C an der Schraubklemme bildet den Bezugspunkt für beide Eingangsspannungsbereiche.

Die Optokoppler IC 101 bis IC 801 realisieren den potentialfreien Anschluss der Eingänge sowohl gegen Erde als auch gegen die anderen Eingänge.

Innen ist jeweils ein Siebglied (bei Kanal 1 : R 101/C 101) nachgeschaltet, das eine pulsierende Gleichspannung glättet. So steht oberhalb einer Steuerfrequenz von ca. 10 Hz, also z. B. auch bei 50-Hz-Wechselspannung, ein konstanter Low-Pegel an den Eingangsports von IC 1.

Bleibt schließlich das in Abbildung 1 unten links abgebildete interne 230-V-Netzteil, das zum einen die benötigte unstabilisierte Schaltspannung für die 24-V-Relais und zum anderen die mit IC 4 stabilisierte 5-V-Betriebsspannung für IC 1 bis IC 3 bereitstellt.

Der Datenaustausch - PC-Interface

Der Datenaustausch zwischen Interface und PC erfolgt über eine Standard-RS-232-

Schnittstelle, wobei folgende Parameter fest vorgegeben sind:

38400 Baud, 8 Bit, ungerade Parität, 2 Stopp-Bits.

Jede Datenübertragung hat einen festen Rahmen (Abbildung 2) und wird stets durch das ASCII-Zeichen SOH eingeleitet. Anschließend folgen die zu übertragenden Daten (bis zu 34 Byte), danach ein Längsparity-Byte und als Abschluss das ASCII-Zeichen EOT.

Zur Bildung des Längsparity-Bytes werden alle Datenbytes der Nachricht sowie des Startbytes XOR-verknüpft.

Da in der Nachricht und im Längsparity-Byte die Steuercodes [SOH] und [EOT] nicht vorkommen dürfen, müssen vor dem Senden der Daten alle Bytes zwischen [SOH] und [EOT] geprüft und eventuell durch eine besondere Zeichenfolge ersetzt werden. Dabei muss die Längsparity-Bildung vorher erfolgt sein.

Folgende Zeichen werden durch die angegebene Zeichenfolge ersetzt:

[SOH] (01h) - [DLE] [11h] (10h, 11h)
[EOT] (04h) - [DLE] [14h] (10h, 14h)
[DLE] (10h) - [DLE] [20h] (10h, 20h).

Beim Empfang von Daten sind diese Zeichenfolgen rechtzeitig zu erkennen und durch die zugehörigen Einzelzeichen zu ersetzen. Erst danach kann eine Überprüfung des Längsparity erfolgen.

Will man zum Beispiel die Bitkombination 00000100 ausgeben, so muss der PC die Nachricht (41h) (04h) zum Interface senden. Gemäß Abbildung 2 muss dann der Datensatz wie folgt aussehen:

(01h) (41h) **(04h)** (44h) (04h).

Da in dieser Nachricht das Steuerzeichen EOT (04h) vorkommt, ist dieses zu ersetzen und somit:

(01h) (41h) **(10h)** **(84h)** (44h) (04h)

zu senden.

Die Nachricht an das Interface selbst besteht aus zwei Teilen, dem Befehlsteil (1 Byte) und den Parametern (0 bis 33 Byte).

Vom Interface zum PC werden auf gleiche Weise zwei Arten von Daten gesendet:

- [Daten] ACK, wobei bei den meisten Befehlen keine Daten zurückkommen und somit nur ACK gesendet wird.
- [Fehlercode] NAK, wobei „Fehlercode“ ein ASCII-Zeichen im Bereich 1 bis 7 ist. Die Tabelle 1 zeigt die Bedeutung der Fehlercodes.

Befehlsbeschreibung

Tabelle 2 zeigt alle Befehle, die das

Interface erkennt. Nachfolgend erfolgt die Erläuterung aller Befehle mit ihren Parametern. Der Befehl ist immer das erste Byte der Nachricht. Falls keine Daten vom Interface erwartet werden, antwortet dieses entweder mit ACK oder mit (Fehlercode) NAK, wie eben beschrieben.

Befehl „A“ (41h)

Ausgabe von 8 Bit direkt

Form: [41h] [w]

w = 8-Bit-Wert, wobei jedes Bit ein Relais ansteuert.

Befehl „D“ (44h)

Einlesen von 8 Bit direkt

Die Antwort vom Interface gibt den momentanen Zustand an den Eingängen wieder.

Form: [44h]

Antwort: [w] ACK

w = 8-Bit-Wert, wobei jedes Bit ein Eingangssignal repräsentiert.

Befehl „G“ (47h)

Daten für eine Sequenz setzen

Form: [47h] [Adr.] [Daten]

Adr.: Startadresse der Daten (0-127)

Daten: Folge von 8-Bit-Daten.

Bitte beachten! Es können jeweils nur 32 Daten übertragen werden. Bei längeren Sequenzen sind die Daten in mehreren Blöcken zu übertragen.

Befehl „L“ (4Ch)

Einlesen des Ausgangszustands

Form: 4Ch

Antwort: [w] ACK

w = 8-Bit-Wert, wobei jedes Bit den Zustand eines Relais anzeigt.

Befehl „M“ (4Dh)

Betriebsmodus setzen

Form: [4Dh][Mode][Max][Zeit]

Es ist das Setzen von 3 Betriebsmodes möglich:

0-127 Automatische Ausgabe aus und Ausgabezeiger auf Position (Mode) setzen.

128 Automatische Ausgabe ein, Stopp am Ende des Datensatzes.

129 Automatische Ausgabe ein, umlaufend.

Max: Länge der Sequenz

Zeit: Triggerabstand (in 100-ms-Schritten), Bereich 100 ms bis 25,5 s.

Als Antwort erfolgt die Übermittlung von ACK.

Test- und Demoprogramm

Zum Lieferumfang gehört ein kleines

Tabelle 1: Fehlercodes vom Interface

“1” (31h): Parity Error beim Datenempfang
“2” (32h): Überlauf des Empfangspuffers
“3” (33h): Fehler beim Längsparity
“4” (34h): unbekannter Befehl
“5” (35h): falscher Parameter
“6” (36h): Datenbereich überschritten

Windows-Demo-Programm, das die ersten Schritte beim Umgang mit dem Interface erleichtert.

Abbildung 3 zeigt das Programmfenster, das einfach ohne aufwendige Installation startbaren Programms.

Nachdem man die belegte serielle Schnittstelle und, falls vorhanden, den belegten Port des ELV-RS-232-Multiplexers

einander alle Relais einschaltet und diese danach wieder nacheinander abschaltet.

Die Anzeige/Eingabe kann sowohl in hexadezimaler wie auch in binärer Form erfolgen.

Befindet sich eine Sequenz im Puffer, kann diese in das Schaltinterface heruntergeladen und gestartet sowie gestoppt werden. Will man, dass sich die Sequenz laufend wiederholt,

Tabelle 2: Befehlsliste

41h („A“)	Ausgabe 8 Bit direkt
44h („D“)	Einlesen 8 Bit direkt
45h („E“)	Einlesen 8 Bit direkt von Kanal B
47h („G“)	Daten setzen
4Ch („L“)	Ausgangszustand einlesen
4Dh („M“)	Betriebsmodus setzen und Zeiger-Reset

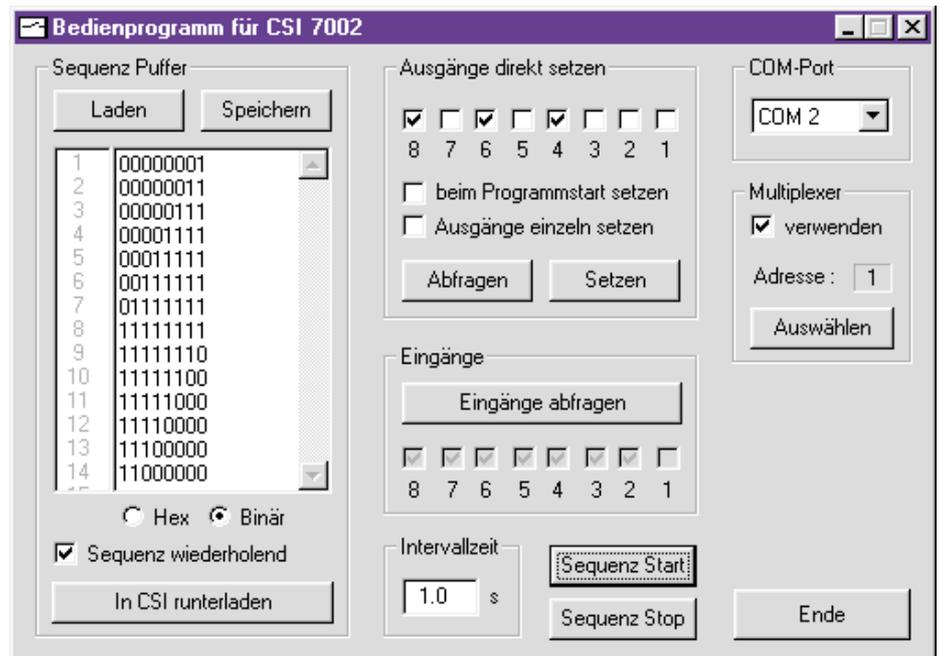


Bild 3: Programmfenster des mitgelieferten Windows-Demo-Programms.

ausgewählt hat, ist die Verbindung zum Interface hergestellt.

Jetzt kann man sowohl die Ausgänge direkt setzen (das jeweilige Relais schaltet) als auch den aktuellen Schaltzustand der Ausgänge abfragen.

Auch der Zustand der Eingänge ist so abfragbar. Ein Häkchen im Fenster bedeutet, dass eine Signalspannung anliegt.

Im linken Fensterteil befindet sich die Puffer-Anzeige für die Programmsequenz. Hier kann man sowohl direkt Programmsequenzen entwerfen als auch solche von der Festplatte laden und dort speichern. Das Beispiel zeigt eine Sequenz, die nach-

markiert man „Sequenz wiederholend“. Die Intervallzeit zwischen den einzelnen Schaltvorgängen ist ebenfalls einstellbar.

Hat man die Sequenz zum Interface geschickt und gestartet, so kann dieses auch vom PC getrennt werden und arbeitet autark weiter.

So kann man bequem und am Objekt lernend die ersten Anwendungen programmieren, bevor man sich an komplette selbstgeschriebene Programme wagt.

Im zweiten Teil des Artikels beschreiben wir den Nachbau und die Inbetriebnahme dieses vielseitig nutzbaren Gerätes.