



PC-Parallel-IO-Interface

Ein PC verfügt prinzipiell über alle Anlagen, mit seiner Umgebung kommunizieren zu können, Steueraufgaben auszuführen und externe Signale zu verarbeiten.

Das hier vorgestellte Interface ermöglicht es, über eine serielle Schnittstelle des PCs 16 Input- und 16 Output-Kanäle zu steuern. Die Übernahme der gesetzten Daten ist extern triggerbar. Ein einmal programmiertes Interface kann die Steueraufgaben dank integriertem Mikroprozessor auch ohne den PC ausführen.

Kommunikation mit der Außenwelt

Wollte man noch vor einigen Jahren Schaltaufgaben durch einen Personalcomputer ausführen lassen, so installierte man eine Relaiskarte in einem freien Slot, ein entsprechendes Bedienprogramm und konnte so z. B. seine Lauflichtkette unendlich laufen lassen.

Heute verfügen moderne Geräte über TTL-Ein- und Ausgänge, die direkt vom Computer selbst steuerbar sind. Dies erfordert jedoch vor allem zwei Bedingungen: der Computer muß ständig laufen, und meist ist die eigentlich für den Drucker reservierte (und einzige) Parallelschnittstelle damit besetzt.

Auch die Frage der Rückmeldung an den Computer ist so nur recht aufwendig lösbar.

Und gerade etwa im Laborbetrieb ist ein direktes Ansprechen von gleichen Signalfamilien untereinander ohne Umwege über Optokoppler, Pegelwandler usw. eine immer wünschenswerte Option. Z. B. sind so bequeme Abfragen von Zuständen in untersuchten Schaltungen möglich.

Auch moderne Lichtsteuergeräte verfügen über TTL-kompatible Ein- und Ausgänge, die mit dem Computer steuerbar sind.

Das vorgestellte Parallel-IO-Interface ermöglicht das Setzen und Abfragen von bis zu 2 x 8 TTL-Ausgängen bzw. 2 x 8 TTL-Eingängen über eine serielle Schnittstelle des PCs. So können z. B. Lichtsteuer-

ungen, die Fernbedienung externer Geräte oder nicht zeitkritische Steuer- und Regelaufgaben recht bequem gelöst werden, zumal die Programmierung über die serielle Schnittstelle relativ einfach vorzunehmen ist.

Außerdem ist durch Einsatz eines RS232-Opto-Isolators mit einfachsten Mitteln eine galvanische Trennung vom PC zu erreichen.

Jeder 8-Bit-Block ist durch ein externes Triggersignal (H/L-Flanke, dazu genügt im einfachsten Fall z. B. ein Taster) triggerbar, d. h. das betreffende Datenbyte wird erst bei Eintreffen eines Triggersignals je nach Übertragungsrichtung entweder ins Ausgangsregister oder in den Prozessor des Interfaces übertragen, von wo aus der angeschlossene PC dann die Daten per Software abfragen kann.

Der Prozessor des Interfaces kann so auch autark arbeiten und einmal eingegebene Bitmuster ständig weiter ausgeben. Einfachstes Beispiel ist etwa eine Bitmuster-Ausgabe für ein Lauflicht.

Der PC kann dabei die programmierten Bitmuster sowohl im (Prozessor-) Datenpuffer des Interfaces ablegen, damit es von dort z. B. per Triggersignal abgerufen werden kann oder direkt die Ein- und Ausgänge setzen oder abfragen.

Für die Zwischenpufferung der Daten verfügt jeder 8-Bit-Block (2 Aus- und 2 Eingangsblocks) über einen eigenen 32-Byte-Puffer im Interface.

Jeder einzelne Block kann dabei vom PC aus in einen von 4 Betriebsmodi versetzt werden.

Ausgabe:

1. keine Reaktion auf ein Triggersignal (das Ausgangssignal ändert sich nicht).
2. Nach dem Triggersignal wird nur das erste Byte des Puffers ausgegeben, weitere Triggersignale sind wirkungslos.
3. Bei jedem Triggersignal wird das nächste Byte des Ausgangspuffers ausgegeben. Die Ausgabe stoppt nach Erreichen eines vorgegebenen Maximalstands. Weitere Triggersignale bleiben dann ohne Wirkung.
4. Verhalten wie unter 3. beschrieben, jedoch wird bei Erreichen des Maximalstands wieder beim ersten Byte des Puffers gestartet.

Eingabe:

1. keine Reaktion auf ein Triggersignal (das Eingangssignal wird nicht übernommen).
2. Mit jedem Triggersignal wird das Eingangssignal im Puffer abgelegt.
3. Bei jedem Triggersignal wird das Eingangssignal im Puffer abgelegt und der Zeiger um 1 erhöht. Nach Erreichen des vorgegebenen Maximalstands findet keine weitere Erhöhung des Zeigers statt.

4. Verhalten wie unter 3. beschrieben, jedoch wird bei Erreichen des Maximalstands wieder beim ersten Byte des Puffers gestartet.

Die Ein- und Ausgabe ist auch automatisch im Zeitraster von 5 ms bis 1,275 s je Takt vornehmbar. Das Triggersignal schaltet hierbei die Aus-/Eingabe ein oder aus (High=Ein; Low=Aus). So kann man ein einmal in den Puffer des Interfaces geladenes Bitmuster beliebig auch ohne den PC laufen lassen.

Die serielle Datenübertragung vom/zum PC erfolgt mit 38.400 Baud, 8 Bit, 2 Stopp-Bits, ungerade Parität.

Schaltung

Zentrale Funktionsgruppe des Interfaces ist der Mikroprozessor IC 1, der alle internen Abläufe regelt und dazu die Pufferspeicher für Ein- und Ausgabe bereitstellt. Seine Takterzeugung erfolgt mit Hilfe der Beschaltung an Pin 9 und 10. Es kommt ein 14,745MHz-Quarz zum Einsatz.

Die zur Verfügung stehenden Ports erlauben eine großzügige Datenbusbedien- ung ohne Multiplexer etc.

Der eigentliche 8-Bit-Datenbus liegt an den Portpins P 0.0 bis P 0.7. Mittels der Portpins 3.0 und P 2.5 erfolgt die serielle Kommunikation über den V.24-Pegelwandler IC 2 mit dem PC.

Über P 3.4 und P 3.5 erfolgt die Ausgabe des Übernahmetaktes an die Ausgangslatches IC 3 und IC 4, während über P 3.6 und P 3.7 die Freigabe der Eingangsregister IC 5 und IC 6 erfolgt.

Die Portpins P 2.0 bis P 2.3 schließlich verarbeiten die externen Triggersignale, und an P 2.7 wird die Datenübertragung vom und zum PC optisch für min. 100 ms über die LED D 1 angezeigt.

IC 2 stellt, in Standardbeschaltung ausgeführt, das RS232-Interface dar, das über die 9polige Sub-D-Buchse BU 1 die normgerechte RS232-Verbindung zur seriellen Schnittstelle des PC realisiert.

Wie bereits erwähnt, erfolgt über IC 3 und IC 4 die Ausgabe der 2 x 8-Bit-Ausgangssignale an ST 2. Die Latches geben das eingeschriebene Signal byteweise aus, sobald an CP eine positive Taktflanke auftritt. Über die Pins 12 und 24 des Steckers ST 2 erfolgt die externe Triggerung für die Eingangskanäle wie beschrieben.

IC 5 und IC 6 stellen das Eingangsregi-

ster dar. Hier eingeschriebene, über BU 3 eintreffende Daten werden ebenfalls byteweise bei Freigabe des Registers über \bar{G} (Freigabe, wenn $G=Low$) in den Prozessor-Puffer übernommen.

Auch hier dienen die Pins 12 und 24 der Buchse BU 3 als Triggereingänge für die externe Triggerung.

Da das Interface auch autark arbeiten können soll, ist eine eigene Spannungsversorgung erforderlich, die über ein externes Steckernetzteil und den Spannungsregler IC 7 realisiert wird.

Die an BU 4 (Klinkenstecker 3,5 mm, Plus am Mittenkontakt) anliegende Gleichspannung (9 V bis 15 V) gelangt über eine Schutzdiode gegen Falschpolung (D 3) an den Spannungsregler IC 7, der eine stabilisierte Spannung von 5 V für die Schaltung bereitstellt. Die Elkos C 8 und C 9 dienen zur Pufferung und Siebung, C 10 bis C 16 der Schwingunterdrückung und Pufferung.

Die für die Bedienung der seriellen Schnittstelle erforderliche negative Spannung wird durch IC 2 selbst erzeugt.

Im zweiten Teil des Artikels folgt die Beschreibung des Datenprotokolls, der Datenübertragung und des Nachbaus der Interface-Baugruppe. **ELV**

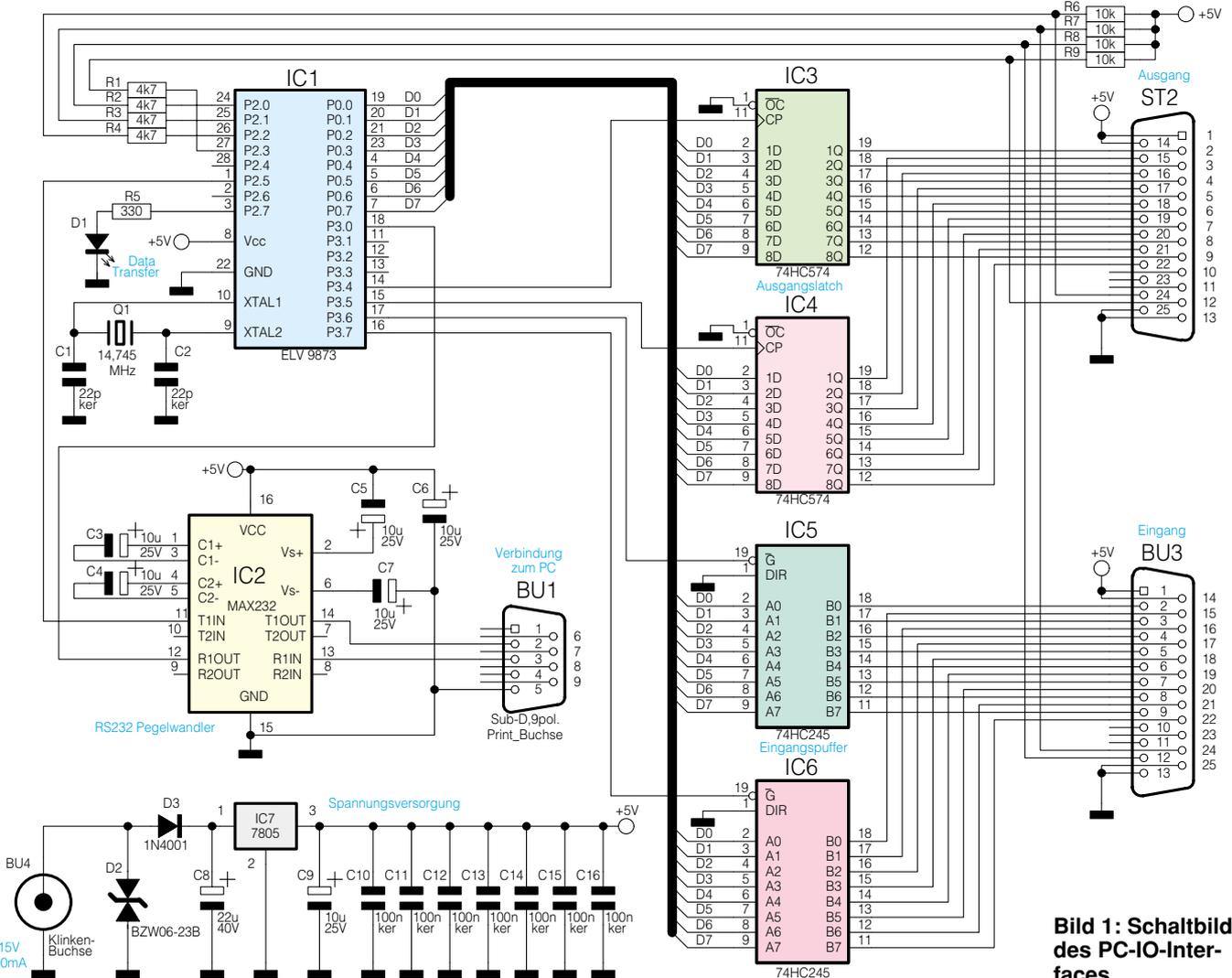


Bild 1: Schaltbild des PC-IO-Interfaces

984158401A