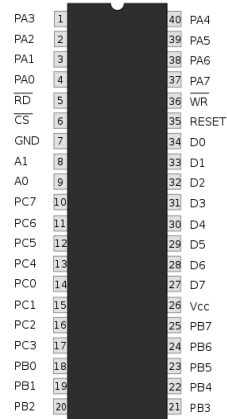


USB-Digital-I/O mit 8255 Portbaustein und CH341A USB Interface Modul für ProfiLab Expert 4.0

Den 8255-Interface-Baustein kann man wohl getrost als "Klassiker" bezeichnen. Ursprünglich als Peripherie-IC für Mikroprozessoren konzipiert, bietet der Chip drei Ports (A,B,C) zusammen 24 TTL-I/O's die portweise als Eingang oder Ausgang programmiert werden können. In der Vergangenheit war der Chip auch auf vielen ISA-I/O-Karten zu finden und sich z.B. auch am LPT-Port betreiben. Mit der Anbindung an den USB wird nun eine weitere Verwendungsmöglichkeit erschlossen. Insbesondere wenn viele I/O's benötigt werden kann der Eigenbau durchaus lohnend sein.



8255 Port Baustein (DIP40)

Der 8255 wird über einen 8-Bit Datenbus (D0...D7) angesteuert und besitzt vier Registeradressen, nämlich drei Statusregister für die drei Ports und ein Control-Register, das u Portrichtungen bestimmt. Die Auswahl eines Register erfolgt über zwei Adressleitungen (A0,A1). Das Lesen und Schreiben der Register wird mit den Steuereleitungen /RD (read) und /WR (write) kontrolliert. Ein Chip-Select-Signal /CS aktiviert den Chip und die Verbindung zum Datenbus. Also ein typisches Mikroprozessor-Peripherie-IC alter Schule.



CH341A (DIP28) USB-Interface Modul

An die Stelle des Microprozessors tritt in diesem Projekt das CH341A-USB-Interface-Modul, welches für diesen Anwendungsfall verschiedene Vorteile hat und so den Entwicklungsaufwand in Grenzen hält. Das Modul bekommt man fertig aufgebaut bei Abacom im Webshop. Mit dem Anschluss an den USB ist das Modul sofort einsatzbereit. Besonders attraktiv ist aber die Tatsache, dass das Modul die Ansteuerung des bi-direktionalen Datenbus bereits "kann", so dass man sich um die Signalerzeugung für /RD, /WR die Datenleitungen nicht selbst kümmern muss. Allerdings besitzt der CH341A nur eine einzige Adressleitung (A0), so dass sich damit nur zwei Registeradressen direkt ansprechen lassen. Daher ist die Verwendung eines Address-Latch (74LS373) erforderlich. A0=LOW adressiert das Address-Latch und A0=HIGH gibt den Bus für andere Chips (hier für den 8255) frei. Die A0-Leitung musste invertiert werden, damit das Latch (zustandsgesteuert!) korrekt arbeitet.

USB PORT I/O (24 lines)
8255 @ CH341A (Expert Profi 2010, ohne Gewähr!)

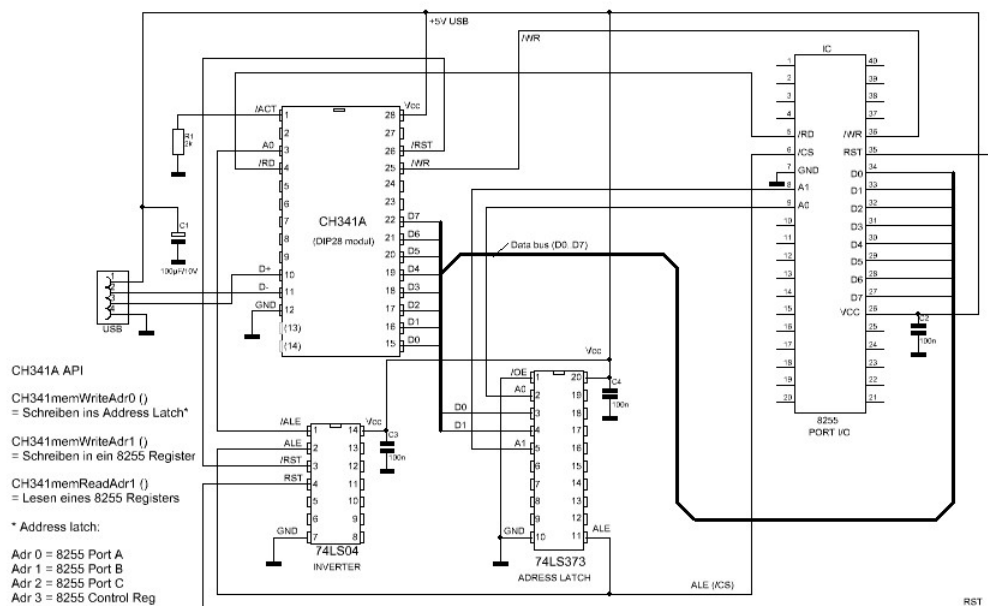


Bild 1: USB-Interface mit nur einem 8255 (ohne Adressdekode)

Um die vier Register des 8255 zu adressieren, schreibt man zunächst die Registeradresse (0 = Port A; 1 = Port B; 2 = PortC; 3 = Control Register) ins Adress-Latch. Dies geschieht beim CH341A mit der API-Funktion CH341MemWriteAdr0.

Anschließend kann man nun über den Datenbus mit dem zuvor adressierten Register Daten austauschen. Die API-Funktionen dafür sind CH341MemWriteAddr1 zum Schreiben in Register und CH341MemReadAddr1 zum Lesen des Registers. Will man nun diese Elementarschaltung um weitere 8255-Chips erweitern, muss man sich noch um die Erzeugung notwendiger Chip-Select-Signale kümmern. Bei nur EINEM 8255 reichte dazu bisher die Adressleitung A0 des CH341A aus, die zwischen Adresslatch und 8255 umschaltet.

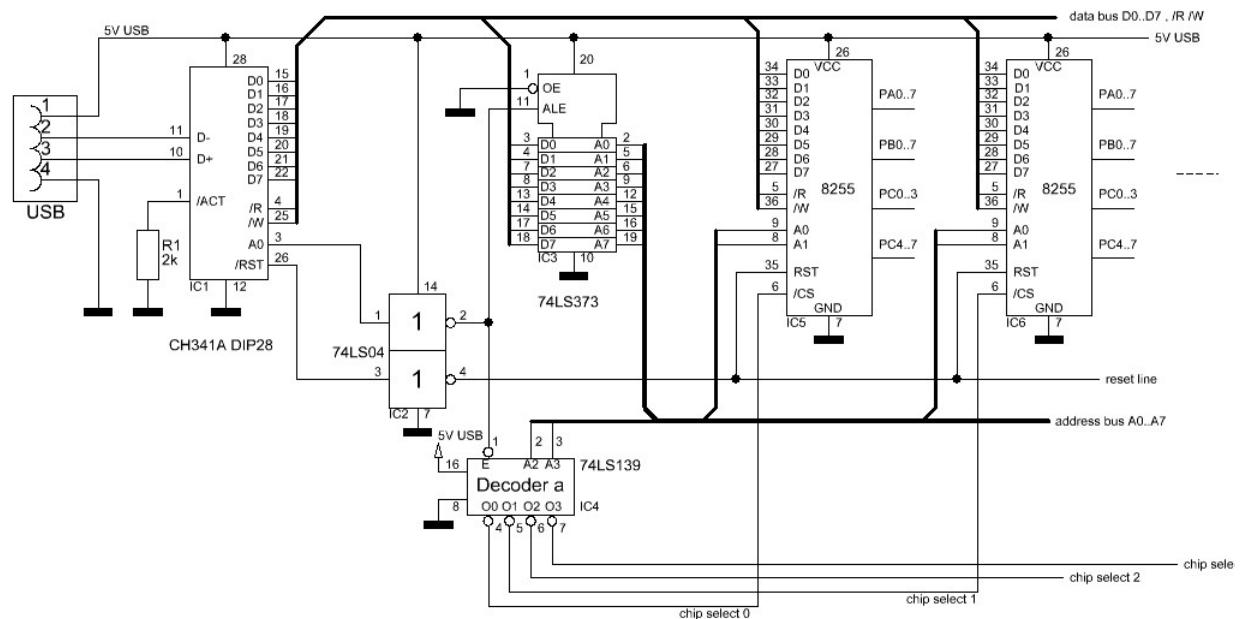


Bild 2. USB-Digital-I/O mit bis zu vier 8255 (mit 74LS139 Adressdekode)

Für den Anschluss mehrerer Chips ist es zunächst einmal erforderlich weitere Adressleitungen bereit zu stellen. Dieses geschieht einfach, indem man den Datenbus vollständig auf Adresslatch aufschaltet. Damit hat man nun einen 8-Bitt-Adressbus zur Verfügung, wobei die unteren beiden Adressleitungen (A0,A1) weiterhin die vier Portregister sämtlicher 8-Chips adressieren. Die oberen Adressleitungen (A2..A7) sind hingegen frei und können für die Dekodierung der Chip Select-Signale genutzt werden.

Diese Aufgabe wird von einem Adressdekode wie dem 74LS138 oder 74LS139 übernommen. Mit einem 74LS138 lassen sich aus drei Adressleitungen des Address-Latch (A2... die /CS-Signale für acht 8255-Chips erzeugen. Die Busfreigabe (A0 vom CH341A) die bisher einen einzelnen 8255 direkt selektierte, dient jetzt als Freigabe (enable) für den Adressdekode. Ebenso kann ein 74LS139 - allerdings nur für vier 8255-Chips - verwendet werden. Damit hat man aber immerhin noch 4x24=96(!) I/O-Leitungen.

Bauteil-Übersicht

- 8255 (Port I/O)
- CH341A DIP28 Modul (USB-Interface)
- 74LS7404 (Inverter)
- 74LS373 (Latch)
- 74LS139 (Decoder für bis zu 8255) oder 74LS138 (Decoder für bis zu acht 8255)

ausserdem:

- 100nF keramisch (Blockkondensatoren)
- USB-Buchse oder -Kabel
- 1x 2,2 kOhm
- 1x 100µF/10V Siebelko

Software

Für die Programmierung werden im wesentlichen nur drei Funktionen aus dem CH341A-API benötigt.

```
CH341MemWriteAddr0 // Schreiben ins Address-Latch
CH341MemReadAddr1 // Lesen eines 8255-Registers
CH341MemWriteAddr1 // Schreiben eines 8255-Registers
```

Diese Funktionen kann man in ProfiLab mit der CH341A-Komponente aufrufen. Das Projekt LATCH.PRJ geht diesen Weg und diente mir zur Prüfung von Addresslatch und Adressdekode.

Bei vielen Registern wird es aber recht aufwendig. Daher habe ich mich zu einer ProfiLab-DLL entschlossen.

Die DLL wird mit der Komponente "DLL-Import" in ein ProfiLab geladen und repräsentiert EINEN Port eines 8255. Je nach Hardwareaufbau und Bedarf kann die Komponente me in einem Projekt eingesetzt werden. Die DLL besitzt einen Konfigurationsdialog. Neben der Einstellung der CH341A-Gerätenummer, wird hier der 8255-Port ausgewählt und die Richtung eingestellt.

Der Wert "Chip Select" wird auf den oberen sechs Adressleitungen ausgegeben (A2...A7), so dass je nach Aufbau des Adressdekoders theoretisch bis zu 64(!) Portbausteine adressiert werden könnten. Das wären dann 1536 I/O-Kanäle. Und DAS habe ich natürlich NICHT ausprobiert!!!! Eine Prüfung auf unsinnige Konfigurationen (z.B. PortA einmal Eingang und einmal als Ausgang) findet NICHT statt.

Die DLL unterstützt also sowohl mehrere 8255 an einem CH341A, als auch mehrere CH341A am USB. Der Datentransfer per USB dauert natürlich erheblich länger als bei einer Karte. Damit ProfiLab nicht zu sehr ausgebremst wird, habe ich eine Begrenzung der Abtastrate auf maximal 50 Samples/Sekunde eingebaut. Bei entsprechend vielen Port kann aber auch noch weniger sein.

Die Sourcefiles, Beispiele, etc. sind mit unten im Download. Den USB-Treiber für den CH341A lädt man von der [Produktseite im Shop von Abacom](#).

Download

[Das ganze Projekt hier herunterladen!](#)

Viel Spass beim Nachbauen!

Der Download darf frei verwendet werden. Ich erhebe keinerlei Ansprüche darauf. Ich hätte nur die Bitte, mir vom Nachbau doch ein Foto zu schicken

ExpertProfi 04/2010

Nachtrag

Ich habe nun noch eine neue Variante der DLL mit folgenden Erweiterungen gemacht:

- PORT DIRECTION: Die Portrichtungen INPUT und OUTPUT habe ich noch mit der Einstellung ONLY READ ergänzt. Damit kann man einen Port lesen, ohne dass die Portrichtung verändert wird. Dieses erlaubt z.B. das Zurücklesen eines Ausgangs-Ports.

- CLOCK MODE: Das Lesen und Schreiben der Ports kann nun alternativ auf zwei Arten ausgelöst werden:

 - AUTO: Alles wie bisher. Ständig Lesen (Polling) und Schreiben bei Zustandsänderung mit maximal 50 Samples/Sek.

 - USER: Lesen und Schreiben ausgelöst durch eine fallende Flanke am CLK-Eingang der DLL.

Das Beispiel UserClock_and_Readback.prj zeigt die die neuen Möglichkeiten.

Hier die neue Variante herunterladen!

