

AX81b: Erweiterungen

1 IO-Erweiterung

Die Anpassung an die Hardware erfolgt im Sourcecode über die "config-..." Dateien im Hauptverzeichnis. In diesen Dateien lassen sich Controllertyp, Speichermodell, Videosystem etc. festlegen. In den vorhandenen config-Dateien gibt es Kommentare zu einzelnen Parametern, von daher ist es sinnvoll eine schon einigermaßen passende Konfiguration als Basis für die eigene Hardwarekonstellation zu verwenden.

Je nach Pin-Konfiguration können bis zu zwei 8 Bit breite IO-Ports benutzt werden. Dabei ist darauf zu achten, dass sich LCD und IO-Port A dieselben Portpins vom AVR-PORTA teilen. Solange der LCD-Modus nicht benutzt wird, ist dies kein Problem.

1.1 Input über Erweiterungsport

Über die Input-Befehle lässt sich zum einen der aktuelle Pegel an den Portpins abfragen, zum anderen aber auch die Verfügbarkeit des IO-Ports abfragen. Das Ergebnis dieser Abfrage ist 0, wenn der entsprechende Port existiert und 255 wenn nicht.

Port HEX	Port DEZ	Funktion
0x8F	143	Datenpins IO Port A
0x9F	159	Verfügbarkeit IO Port A
0xAF	175	Datenpins IO Port B
0xBF	191	Verfügbarkeit IO Port B

1.2 Output über Erweiterungsport

Mit den Output-Befehlen werden die Portpins angesteuert sowie die Datenrichtung derselben festgelegt. Bei der Datenrichtung legt eine "1" im Byte den korrespondierenden Pin als Ausgang fest, eine "0" entsprechend als Eingang. Wenn ein Pin als Eingang geschaltet ist, bestimmt das zugehörige Ausgangsbit, ob der Controller-interne Pull-Up aktiviert ist.

Port HEX	Port DEZ	Funktion
0x8F	143	Datenport IO Port A
0x9F	159	Richtungssteuerung IO Port A
0xAF	175	Datenport IO Port B
0xBF	191	Richtungssteuerung IO Port B

1.3 IO-Befehle beim ZX81

Der ZX81 hat keine BASIC-Befehle für Ein- und Ausgabe über IO-Ports. Diese müssen per Software realisiert werden. Die dafür notwendigen Routinen finden sich an Adresse 32576 (OUT) und 32592 (IN). Für die Adress- und Datenübergabe werden die Spare-Bytes benutzt. Dazu dienen die beiden Bytes bei 16507/8 als Adresse (nur das LOW Byte bei 16507 wird ausgewertet) und das Byte bei 16417 als Datenbyte bei der OUT-Routine.

Um den an IOPORT A gelesenen Wert dezimal auszugeben, kann das folgende Programm genutzt werden. Dabei werden nacheinander der Port als Input geschaltet, die Pullups aktiviert und der I/O Port eingelesen.

```
10 POKE 16507,159
11 POKE 16417,0
12 RAND USR 32576
13 POKE 16507,143
14 POKE 16417,255
15 RAND USR 32576
16 PRINT USR 32592
```

1.4 IO-Befehle beim ZX80

Der ZX80 hat keine BASIC-Befehle für Ein- und Ausgabe über IO-Ports. Diese müssen per Software realisiert werden. Im Moment sind keine Routinen dafür eingebaut.

1.5 IO-Befehle beim Jupiter ACE

Für die I/O-Ports stehen zwei Wörter zur Verfügung, nämlich **IN** und **OUT**. Lt. Manual sind sie gleichbedeutend mit den Speicherwörtern **C@** und **C!**.

IN erwartet die Adresse auf dem Stack und legt das gelesene Byte wieder dorthin zurück.

OUT erwartet die Adresse auf dem Stack und das zu schreibende Byte auf dem vorletzten Stackeintrag, nach der Ausführung des Wortes werden beide Stackeinträge entfernt.

```
: AREAD
  0 159 OUT
 255 143 OUT
 143 IN
;
```

AREAD . gibt den an IOPORT A gelesenen Wert dezimal aus.

1.6 IO-Befehle beim ZX SPECTRUM

Für die I/O-Ports stehen der Befehl **OUT** und die Funktion **IN** Verfügung.

IN addr erwartet die Adresse als Parameter und liefert das gelesene Byte als Resultat zurück.

OUT addr,data erwartet die I/O-Adresse und das zu schreibende Byte als Parameter.

```
10 OUT 159,0
20 OUT 143,255
30 PRINT IN 143
```

Dieses kleine Programm gibt den an IOPORT A gelesenen Wert dezimal aus.