

# Retrocomputing

## EIN / AUSGABE - EINHEITEN

CP/M verfügt über vier *logische* E/A-Einheiten. Eine *logische* E/A-Einheit entspricht dabei einem symbolischen Namen für eine Gruppe von E/A-Einheiten, die alle von CP/M bedient werden können. Diese Gruppen sind:

**CON:** für „Console“ oder ein Terminal das Bildschirm und Tastatur besitzt (Eingabe/Ausgabe)

**RDR:** Eingabe vom Lochstreifenleser

**PUN:** Ausgabe auf Lochstreifenstanzer

**LST:** für „List“ Geräte wie z.B. Drucker

Über das I/O-Byte, kann jedem der *logischen* E/A-Einheiten eine tatsächliche *physikalische* vorhandene E/A-Einheit zugewiesen werden. Das I/O-Byte befindet sich im Arbeitsspeicher auf Adresse 0003h und ist folgendermaßen aufgebaut:

**Tab. 1:** Aufbau des I/O-Byte

LST		PUN		RDR		CON	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	x	x	x	x	x	x

Jedes Feld (zwei Bit) kann somit einen Wert zwischen 0 und 3 einnehmen und bestimmt dadurch die Zuordnung zu einer bestimmten, tatsächlich existierenden, *physikalischen* E/A-Einheit. CP/M erlaubt die folgenden *physikalischen* Einheiten:

**TTY:** Teletype device (slow speed console)

**CRT:** Cathode ray tube device (high speed console)

**BAT:** Batch processing (console is current RDR; output goes to current LST: device)

**UC1:** User-defined console

**PTR:** Paper tape reader (high speed reader)

**UR1:** User-defined reader #1

**UR2:** User-defined reader #2

**PTP:** Paper tape punch (high speed punch)

**UP1:** User-defined punch #1

**UP2:** User-defined punch #2

**LPT:** Line printer

**UL1:** User-defined list device #1

Welche E/A-Einheiten tatsächlich implementiert sind, hängt von der jeweiligen Hardware des eigenen Computers ab. Im aktuellen System existiert die folgende Zuordnung zu den *physikalischen* E/A-Einheiten:

<b>TTY:</b>	AVR-UART	Console	(input and output)
<b>CRT:</b>	I2C-UART	Console	(input and output)
<b>LPT:</b>	I2C-GPIO	Printer	(output only)
<b>UL1:</b>	I2C-UART	Reader	(input only)
<b>UP1:</b>	I2C-UART	Punch	(output only)
<b>BAT:</b>	AVR-UART	Console	(input and output)
<b>UC1:</b>	AVR-UART	Console	(input and output)

Über den **STAT-Befehl** (status) kann der Zustand des eigenen Systems abgefragt und auch verändert werden. Die Abfrage zeigt, dass die Console (Bildschirmausgabe und Tastatureingabe) über die AVR-UART-

---

```
A>stat dev:
CON: is TTY:
RDR: is PTR:
PUN: is PTP:
LST: is LPT:
```

---

Schnittstelle (**TTY**) bedient wird. **RDR** und **PUN** werden nicht bedient, da **PTR** keinem *physikalischen* Gerät zugeordnet ist. Weiterhin können mit dem **STAT-Befehl** alle möglichen Zuordnungen des I/O-Bytes abgefragt werden.

---

```
A>stat val:

Temp R/O Disk: d:=R/O
Set Indicator: d:filename.typ $R/O $R/W $SYS
$DIR
Disk Status   : DSK: d:DSK:
User Status   : USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

---

So kann der Console z.B. die tatsächlich vorhandene **CRT**-Einheit (I2C-UART) zugeordnet werden. Alle Tastatureingaben und Bildschirmausgaben erfolgen ab dem Zeitpunkt der Zuordnung über die I2C-UART Schnittstelle.

```
A>STAT CON:=CRT:
```

Ist die **CRT**-Einheit nicht korrekt angeschlossen oder arbeitet fehlerhaft, so kann das CP/M-System nicht mehr über die Tastatur bedient werden. Bei korrekter Arbeitsweise lässt sich jedoch die nun geänderte Gerätezuordnung überprüfen.

```
A>STAT DEV:
CON: is CRT:
RDR: is PTR:
PUN: is PTP:
LST: is LPT:
```

Ein entsprechend umgekehrter Befehl schaltet die Console wieder auf die ursprüngliche E/A-Einheit zurück (AVR-UART).

```
A>STAT CON:=TTY:
```

CP/M bietet auch die Möglichkeit die *physikalischen* Geräte direkt anzusprechen. Dazu werden die Gerätenamen direkt angesprochen.

```
A>PIP TTY:=readme.txt
```

Der Inhalt der Datei „readme.txt“ wird unabhängig von der jeweiligen Zuordnung des I/O-Byte direkt auf **TTY** (AVR-UART) gesendet. Der Befehl

```
A>PIP LPT:=readme.txt
```

sendet die Datei „readme.txt“ direkt, unter Umgehung des I/O-Byte, an den Drucker (**LPT**).