

## RTC 72421 auslesen



Die GIDE+USB-Platine enthält mit dem bei Bedarf zu bestückenden 72421 auch Echtzeituhr und Kalender. Nachfolgend werden dazu einige Hinweise incl. lauffähiger Beispiele gegeben.

Sie stellen lediglich die nötigen Angaben für das Grundverständnis dar und können als Anregung für eigene Experimente dienen. So wird z.B. auch auf das doppelte Auslesen der Zeit (um Probleme bei Änderungen der Zeit im Auslesemoment zu vermeiden) verzichtet.

Nicht eingegangen wird hier auf die Programmierung (Stellen der Uhr). Das kann z.B. bequem mit dem Programm **GIDE-Check** von Ralph Hänsel erfolgen.

### Grundlagen

Für das Ansprechen der Echtzeituhr ist eine 16-Bit-Adresse nötig:

Register-adresse	Bedeutung
0085	Sekunden, Einer
0185	Sekunden, Zehner
0285	Minuten, Einer
0385	Minuten, Zehner
0485	Stunden, Einer
0585	Stunden, Zehner & AM/PM flag
0685	Tag, Einer
0785	Tag, Zehner
0885	Monat, Einer
0985	Monat, Zehner
0A85	Jahr, Einer
0B85	Jahr, Zehner
0C85	Wochentag (0=Sonntag, 1=Montag,...)
0D85	Steuerregister D (status/control)
0E85	Steuerregister E (pulse output control)
0F85	Steuerregister F (master control)

Der niederwertige Teil **85h** steht hier für den Uhrenport, was sich aus der u.a. beim LLC2 angewandten Basis-Adressierung des GIDE+USB-Moduls (80h) ergibt. Der höherwertige Teil der Adresse identifiziert eines der insgesamt 16 Register.

Einen einfachen 16-Bit-Lese-Zugriff realisiert automatisch z.B. der Befehl **IN A,(C)**:

Mit 85h in Register C (Daten liegen beim Lesen an A0...A7)  
und 00...0Fh in Register B (Daten liegen beim Lesen an A8...A15)

holt **IN A,(C)** den Wert aus dem entsprechenden Register nach A. Da die oberen 4 Bit des Registerinhalts nicht relevant sind, führt man anschließend noch **AND 0Fh** aus. Das Ergebnis ist eine Zahl von 0...15 und ist ggf. noch zu konvertieren.

## RTC 72421 auslesen

### Beispiele:

#### MC-Programm zum Lesen von Datum und Uhrzeit (RTC\_1900.BIN)

PUFFER: EQU 1880H RTCPORT: EQU 85H ORG 1900H	
DEFM 0,9,'z',0DH ;VORSPANN, KENNBUCHSTABE 'z' LD HL,PUFFER ;PUFFERADRESSE PUSH HL ;ANFANG SICHERN LD (HL),0DFH ;RST18 FÜR AUSGABE ABLEGEN INC HL LD C,RTCPORT ;IO-Adresse RTC LD B,0CH ;ZU LESENDES RTC-REGISTER (START) LD D,6 ;6 x 2 STELLEN LESEN: DEC B ;ALLE GEWÜNSCHTEN REGISTER ABKLAPPERN IN A,(C) ;RTC BYTE1 HOLEN (B=REGISTER!) AND 0FH ;NUR NIEDERWERTIGER TEIL OR 30H ;ALS ASCII-WERT LD (HL),A ;ABLEGEN INC HL DEC B IN A,(C) ;RTC BYTE2 HOLEN AND 0FH OR 30H LD (HL),A INC HL LD (HL),'/' ;TRENNER FÜR DATUM INC HL DEC D JR NZ,LESEN POP IX ;PUFFERANFANG=STARTADRESSE LD (IX+09H),' ' LD (IX+0CH),':' ;NEUE TRENNER FÜR ZEIT LD (IX+0FH),':' LD (IX+12H),0A0H ;ENDE FÜR RST18-AUSGABE LD (IX+13H),0C9H ;RÜCKKEHR ;AUSGABEART----- LD A,(185BH) ;ARGUMENT HOLEN CP 3 ;>=3? RET NC ;JA, NUR PUFFER BEFÜLLT CP 2 ;2? JR Z,KOMPL ;JA = KOMPLETTAUSGABE CP 0 ;0 ODER KEIN ARGUMENT? JR Z,ZEIT ;JA = NUR ZEIT DAT: SET 7,(IX+09H) ;NUR DATUM AUSGEBEN LD (IX+0AH),0C9H ;VORZEITIGES RET JR KOMPL ZEIT: LD (IX+09H),0DFH ;VERZÖGERTES RST18 LD IX,PUFFER+9 ;NEUER ANFANG AUSGABE KOMPL: JP (IX) ;ZK AUSGEBEN UND ZURÜCK ZUM AUFRUFENDEN PROGRAMM	
Das Programm gibt (wenn Option <>3) den Puffer als ASCII an der momentan aktuellen BS-Position aus.	
Aufruf aus Monitor: z 0 nur Uhrzeit (0 kann entfallen) z 1 nur Datum z 2 Datum+Zeit z 3 keine Anzeige, nur Puffer befüllen	Aufruf aus Anwenderprogramm, z.B.:  LD A,1 CALL 1904
PUFFERSTRUKTUR/INHALT ab 1880h: 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 DF J J / M M / D D H H : M M : S S C9 10 1 10 1 10 1 10 1 10 1 10 1	

## RTC 72421 auslesen

Unter GS-BASIC ist mit der **INP()-Funktion** zwar ein direkter Portzugriff möglich, aber nicht in 16 Bit<sup>1</sup>. Es kann nur die Uhrenportadresse (85h) übergeben (gelesen) werden. Der höherwertige Teil der Adresse ist immer 00, womit lediglich die Sekunden/Einer gelesen werden. Der Zugriff auf die anderen Daten sowie die Steuerregister ist damit verwehrt. **INP(133)** holt also nur die Sekunden/Einer. Mit  $X = \text{INP}(133) \text{ AND } 15$  werden die nicht relevanten oberen 4 Bits ausgeblendet, sodass sich ein Wert von 0...15 ergibt. Das kann als Zeitgeber in einigen Fällen bereits genügen.

Für die Erlangung von aktueller Zeit sowie Datum ist etwas MC-Code zu benutzen. Er läuft hier im Bereich 1900h...190Bh. Für einen anderen Bereich ist der Code entsprechend anzupassen. Als Übergabezelle für das gewünschte Uhrenregister ist 1901h definiert.

### MC-Teil:

1900	06 0C	LD B,0CH	;UHRENREGISTER-NR. IN 1901
1902	0E 85	LD C,85H	;UHRENPORT
1904	ED 78	IN A,(C)	;RTC BYTE HOLEN
1906	E6 0F	AND 0FH	;HWT AUSBLENDEN
1908	32 01 19	LD (1901),A	;RÜCKGABEWERT IN 1901
190B	C9	RET	

### BASIC-Teil:

Der MC-Code kann z.B. mit der **POKE-Methode** zusammen mit dem BASIC-Programm abgespeichert werden.

### Vollständiges Beispiel in GS-BASIC (RTCBAS.z80)

10	CLS:PRINT"RTC-DATUM UND UHRZEIT AUSLESEN:"	
20	GOSUB240	:REM MC poken
30	F=7:GOSUB210	:REM Tag 10er
40	F=6:GOSUB210	:REM Tag 1er
50	PRINT". ";	
60	F=9:GOSUB210	:REM Monat 10er
70	F=8:GOSUB210	:REM Monat 1er
80	PRINT". ";	
90	F=11:GOSUB210	:REM Jahr 10er
100	F=10:GOSUB210	:REM Jahr 1er
110	PRINT", ";	
120	F=5:GOSUB210	:REM Stunden 10er
130	F=4:GOSUB210	:REM Stunden 1er
140	PRINT": ";	
150	F=3:GOSUB210	:REM Minuten 10er
160	F=2:GOSUB210	:REM Minuten 1er
170	PRINT": ";	
180	F=1:GOSUB210	:REM Sekunden 10er
190	F=0:GOSUB210	:REM Sekunden 1er
200	END	
210	POKE6401,F:CALL*1900:X=PEEK(6401)	:REM RTC lesen
220	X=X+48:PRINTCHR\$(X);	:REM als ASCII ausgeben
230	RETURN	
240	RESTORE270	:REM MC-Code
250	FORI=0TO11:READX:POKE6400+I,X:NEXT	:REM poken
260	RETURN	
270	DATA 6,12,14,133,237,120,230,15,50,1,25,201	:REM obiger MC-Code dezimal

Erstellt:

WeRo

Stand: Nov. 2013

<sup>1</sup> Zumindest ist mir aus den verfügbaren Unterlagen keine Möglichkeit bekannt.