

## a. Beschreibung EPC-1

### 1. Allgemeines

Der EPC-1 ist ein universeller Einplatinen-Computer im Standard-Euroformat, der sowohl "stand-alone" als auch als Slave parallel am ECB-Bus betrieben werden kann.

Seine Anwendungen reichen von einfachsten Aufgaben wie selbstaendiger Rechner mit Tastatur und Ausgabe (LCD u.d.) bis zu hochkomplexen Aufgaben als Slave-Rechner am ECB-BUS fuer Datenkonzentration, Datenwandlung, Maschinensteuerung mit einem oder mehreren EPC-1 u.s.w.

Applikationen: Robuster Steuerrechner, ideal fuer Maschinensteuerungen, Spooler/Despooler, universeller Slave am ECB-Bus ( Multiprozessor-Betrieb ), Schnittstellen-Wandler ( IEC etc. ), Uhr, Lehrcomputer ( a.A. mit 16 K Basic )

Mit aufsteckbarer Sandwichplatine lassen sich beliebige eigene Erweiterungen ohne direkte Eingriffe realisieren.

#### Technische Daten:

CPU Z-80-A mit 2,4576 MHz oder 4,9152, einstellbar mit Jumper

Speicher Vier 28-polige Steckplaetze fuer max. 64 K Byte.  
Eprom 2716 bis 27256 ( 2 K x 8 bis 32 K x 8 ) und/oder  
CMOS-RAM 6116 bis 6264 ( 2 K x 8 bis 8 K x 8 ).  
Eprom und RAM beliebig mischbar.  
CMOS-RAM's koennen mit Akku gepuffert werden.

#### Schnittstellen

Seriell Z-80-A SIO/0 mit 2 V 24-Schnittstellen, alle Signale voll gebuffert.

Baudraten hardwaremaessig einstellbar von 600, 1200, 2400, 4800, 9600 Baud bei 2,4576 MHz, 1200, 2400, 4800, 9600, 19200 Baud bei 4,9152 MHz.

Parallel sechs 8-Bit Parallelports

an 30 pol. Stiftleiste: 2 x TTL bidirektional (74LS245 )  
2 x TTL mit Latch (74LS374 )

an 20 pol. Stiftleiste: 1 x TTL bidirektional mit Latch ( 3 x 74LS374 ) ECB-Bus

Uhr Uhrenchip 58174 von NS (optional)

Stromversorgung am ECB-Bus : a/c 1 +5V  
a/c 32 GND  
a 13 +12V  
c15/a22 -12V

Akku 3,6 V/60mA fuer CMOS-RAM's und Uhr, am ECB-BUS aufladbar (optional)

## Portbelegung Grundadressen der internen Ports:

IC 19 Adr. 80H  
IC 17 Adr. 88H  
IC 18 Adr. 90H  
IC 8 Adr. 98H  
IC 9 Adr. A0h  
IC 11 Adr. A8H  
IC 10 Adr. B0H  
IC 20 Adr. B8H Sio Grundadresse

Die externen Ports vom ECB-Bus koennen per Jumper X1 eingestellt werden.

Adressierung der RAM/ROM's : Steckplatz 4 sollte fuer ein Eprom reserviert sein, moegliche Eprom-Typen sind 2716-27256 und ihre Prom-Aequivalente. Die Speichersockel 5, 6, 7 sind wahlweise mit Eprom oder Ram bestueckbar. Es sind Jumper vorhanden, um sowohl 6116 als auch 6264 in kontinuierlichen RAM-Bereichen anzusprechen.

Die Bestueckung sollte bei gemischter Bestueckung immer bei den niedrigeren Steckplaetzen Rom's vorsehen und bei den hoeheren Ram's.

Falls die drei hoeheren Steckplaetze mit Ram's bestueckt werden, beginnt der RAM-Bereich bei 32 K.

Adressierung am ECB-Bus. Der EPC-1 ist am ECB-Bus wie ein normaler Portbaustein ansprechbar. Dabei wird IC 17 als Status-Port verwendet, um der Haupt-CPU die Kommunikationsfaehigkeit des Slaves mitzuteilen. Die interne Synchronisation sollte so erfolgen, dass ein Zugriff der Haupt-CPU auf IC-12 CS3 einen Interrupt der Slave-CPU auf dem EPC ausloest. Alternativ kann die Slave-EPC-CPU auch den Port pollen, dann sollte jedoch ein spezielles Protokoll eingefuehrt werden.

Mit dem Jumperfeld X1 kann die Grundadresse des EPC auf dem ECB-Bus eingestellt werden :

a	gejumpert	=	0h	20
b	"	=	20h	22 60
c	"	=	40h	A0
d	"	=	80h	E0

Jumper fuer Interrupt: X 21 wahlweise Interrupt von der SIO oder vom ECB-Bus.

### NMI-Jumper:

Fuer Watchdog Anwendungen kann die Uhr einen Nicht-maskierbaren Interrupt ausloesen. Fuer Anwendungen, wo die Zeitaufloesung der Uhr nicht ausreicht, kann der schnelle Counter IC 15 herangezogen werden.

### Reset:

Der Reset erfolgt normalerweise mit Power-On. Es ist dazu eine genuegend schnell ansteigende Spannungsquelle erforderlich.

### Manueller Reset im Stand-alone Betrieb:

Der EPC-1 kann auch manuell zurueckgesetzt werden. Es wird dazu ein Momenttaster parallel zum Kondensatr C3 geschaltet.

### Reset am ECB-BUS:

Fuer universelle Slave Anwendungen am ECB-BUS ist eine Reset Moeglichkeit des Slave von der Haupt-CPU vorgesehen. Es muss dazu nur die Adresse XF angesprochen werden (X=Grundadresse auf ECB-BUS).

### Ansteuerung der Uhr:

Da die heutigen Uhrenchips noch relativ langsam sind, wurden zwei Moeglichkeiten zum Ansprechen der Uhr vorgesehen:

1. Der Uhrenchip erhaelt seine Adresse gepuffert (normale Betriebsart): IC 11 liefert die Chipadressen.
2. Direkte Betriebsart: Die CPU liefert die Uhrenadressen (normalerweise nicht empfohlen).

### Akku-Pufferung:

Die Akku-Pufferung kann so geschaltet werden, dass nur die Uhr und die CMOS-Rams Batteriespeisung erhalten. Eine automatische Aufladung am ECB-Bus (oder der Spannungsquelle im Stand-alone-Betrieb) gewaehrleistet einen sicheren Betrieb. Zur Jumperung siehe Schaltung.

### JUMPERUNG DER STANDARD-AUSLIEFERVERSION

Steckplatz 4: Eprom 2716/2732 (2K)

Steckplatz 5: 6116 (2K)

ab Adr 0000

ab Adr 8000

(832718 NEC-D446C-1)

Reset mit Power-on

Baudrate 9600 Baud Kanal A der Sio fuer Terminal

ECB-Bus nicht gejumpt

Uhr nicht gejumpt

### MONITOR BESCHREIBUNG

Der auf dem EPC-1 implementierte Monitor stellt alle Standard-Monitorfunktionen zur Verfuegung. (Siehe Beschreibung der Monitor-Kommandos). Er bedient in der Normalversion nur die serielle V-24 Schnittstelle zum Terminal.

Die Eprom-Software stellt beim Start das kontinuierlich verfuegbare RAM selbstaendig fest.

Eine entsprechende Meldung wird ausgegeben.

Die hoechsten 40 Bytes sollten fuer den Stack reserviert bleiben.

Ebenso versucht der Monitor, die belegten I/O-Ports zu lokalisieren. Zu beachten ist, dass diese Routine nur bei angeschlossener Peripherie richtige Resultate bringen kann.

Der Monitor belegt etwa 2 KByte des Roms. Es ist also genuegend Platz vorhanden, um eigene Erweiterungen unterzubringen.

### Sonderfunktionen:

Sonderfunktionen wie serieller Programm-Download oder Bedienung der ECB-Bus Schnittstelle sind bewusst nicht implementiert, da es sich gezeigt hat, dass die Einsatzgebiete des EPC-1 dermassen vielfaeltig sind, dass keine generalisierten Routinen implementierbar sind.

Hierfuer ist von uns das Monitorlisting zum Selbstkostenpreis erhaeltlich.

Hinweise und Tips fuer spezielle Applikationen:

1. Einfache Stand-alone Anwendungen

Eine oft gefragte Anwendung des EPC-1 ist die absolute Stand-Alone Anwendung mit LED oder LCD-Anzeige und kleiner Tastatur. Diese Anwendung laesst sich am einfachsten durch ein Multiplexen der Anzeige als auch der Tastatur auf den Ports IC 8-9-10-11 erreichen. Es koennen so Tastaturen bis zur vollen Ascii Groesse und Anzeigen (sowohl hex als auch dez als auch Ascii) bis zu 65000 Zeichen jen nach Komplexitaet des Anzeigeelements bedient werden. Wegen der unterschiedlichen Tastaturen und Displays kann hier kein Beispiel-Listing gegeben werden.

2. Despooler

Hierfuer empfiehlt sich die derzeitig groesstmoeegliche C-MOS Ram Version. Die folgende Beispiel-Software versucht dabei moeglichst viel der auszugebenden Daten intern zu puffern, um den Hauptrechner zu entlasten.

interuptroutine:

```
LD    A,BUSY
OUT   (STATUSPORT),A
IN    A,DATAPORT
LD    (HL),A
INC   HL
LD    A,H
CP    D           ;D HAT BUFFMAX
JP    Z,BUFFOUT
LD    A,READY
OUT   (STATUSPORT),A
RETI
```

BUFFOUT:

```
POP   AF           ;STACK
LD    A,PRINT      ;HAUPTCPU SIGNALISIEREN BUFFER VOLL
OUT   (STATUSPORT),A
CALL  PRINTOUT     ;AKTUELLES LEEREN DES BUFFERS
LD    A,READY
OUT   (STATUSPORT),A
JP    HERE         ;WAIT FOR INTERRUPT
```



Monitorfunktionen

EPC-Monitor

A Put ASCII into Memory

Einfache Eingabemöglichkeit für ASCII.  
^C bildet den Abschluß.

Dxxxx,yyyy Display

Display-Memory in Hex und ASCII. 16 Bytes in Hex und ASCII werden in einer Zeile von Anfangsadresse X bis Endadresse Y gezeigt.

Fxxxx,yyyy,zz Fill

Fill Memory mit zz. Von Anfangsadresse X bis Endadresse Y wird das Byte Z eingetragen.

Gxxx Goto

Hiermit kann ein ausführungsfähiges Programm gestartet werden. X ist Startadresse.

Jxxxx,yyyy Test

Memory-Test. Gestattet einen einfachen RAM-Test. Falls ein Fehler gefunden wurde, sollten komplexere Testprogramme eingesetzt werden, um zwischen Hard und Softerrors unterscheiden zu können. (Siehe auch Verify-Kommando).

Mxxxx,yyyy,zzzz

Beachten sollte man, das die Speicherstelle, in der sich das Programm selbst modifiziert, selbstverständlich einen Fehler anzeigt.

Oxx,yy Query Out

Move Memory-Bereich X bis Y nach Z. Der Speicherbereich X bis Y wird nach Z bewegt mit der Länge Y-Z. Auf Überschneidungen der Bereiche ist zu achten.

Ixx,yy Query In  
Sxxxx,yy Substitute

Output-Byte Y to Port X. Sehr brauchbarer Befehl, um die Peripherie in Betrieb zu bringen. Erfordert allerdings Hardware-Kenntnisse.

Txxxx,yyyy Type

Input-Byte Y from X. Siehe oben.  
Ersetze Speicherstelle xxxx durch Wert Y. Mit dem S-Befehl kann jede beliebige Speicherstelle mit dem Wert Y belegt werden. Fortschaltung zur nächsten Speicherstelle geschieht mit der Leertaste.

Vxxxx,yyyy,zzzz

Der Type-Memory-Befehl erlaubt es, den Speicherbereich X bis Y in ASCII auszudrucken. Er erlaubt eine schnelle Übersicht über das geladene Programm. Auch diesen Befehl findet man selten in anderen Debuggern.

Yxx,yy,... YS there

Verify Speicherbereich X bis Y gegen Z.  
Mit dem Verify-Befehl können zwei Speicherbereiche gegeneinander verglichen werden. Nützlich nach Move-Befehl.

Z

Das Y-Kommando sucht nach einer Byte-Folge X;Y;Z;.... Bis zu 255 Byte können eingegeben werden. Die Eingabe erfolgt in Hex.

Show End of RAM.

Master

Download

16.3.88

MEC 832718  
346C-1 PART  
? K

Slave

Cmd Reset für Slave (Adr CF)

Cmd Adr/adresse a (2 Bytes)  
an Slave

Cmd Programmlänge (2 Bytes)  
an Slave

Cmd Start adr. s (2 Bytes)

Cmd Programm (n Bytes)

<Reset>  
Init. Adress Zähler  
Hole Anfangsadr. a

Hole Progs. Länge n

Hole Startadr. s

Hole Programm

Starte Programm bei  
Startadresse

Byte vom Master zum Slave

Werte auf 01 in Chip 17  
(Adr C1, → CS1)

Schreibe Byte in  
Chip 19 (Adr. C0, → CS3,  
löst gleichzeitig Interrupt aus)

Schreibe 01 in Chip 17 (ready)  
Warte auf Interrupt

Interrupt service:

Schreibe 00 in Chip 17 ("not ready")

Lies Byte aus Chip 19 (Adr. C0, CS6)

Schreibe 01 in Chip 17 (ready)

Ab speichern bei Adress-Zähler,  
Adress-Zähler inkrementieren

Return from Interrupt

Byte vom Slave zum Master

Werte auf 02 in Chip 17  
(Adr C1, → CS1)

Lies Byte aus Chip 18  
(Adr. C2, → CS2)

Schreibe Byte in Chip 19  
(Adr C0, → CS3, → Interrupt)

Schreibe Byte in Chip 18  
(Adr. 90, → CS5)

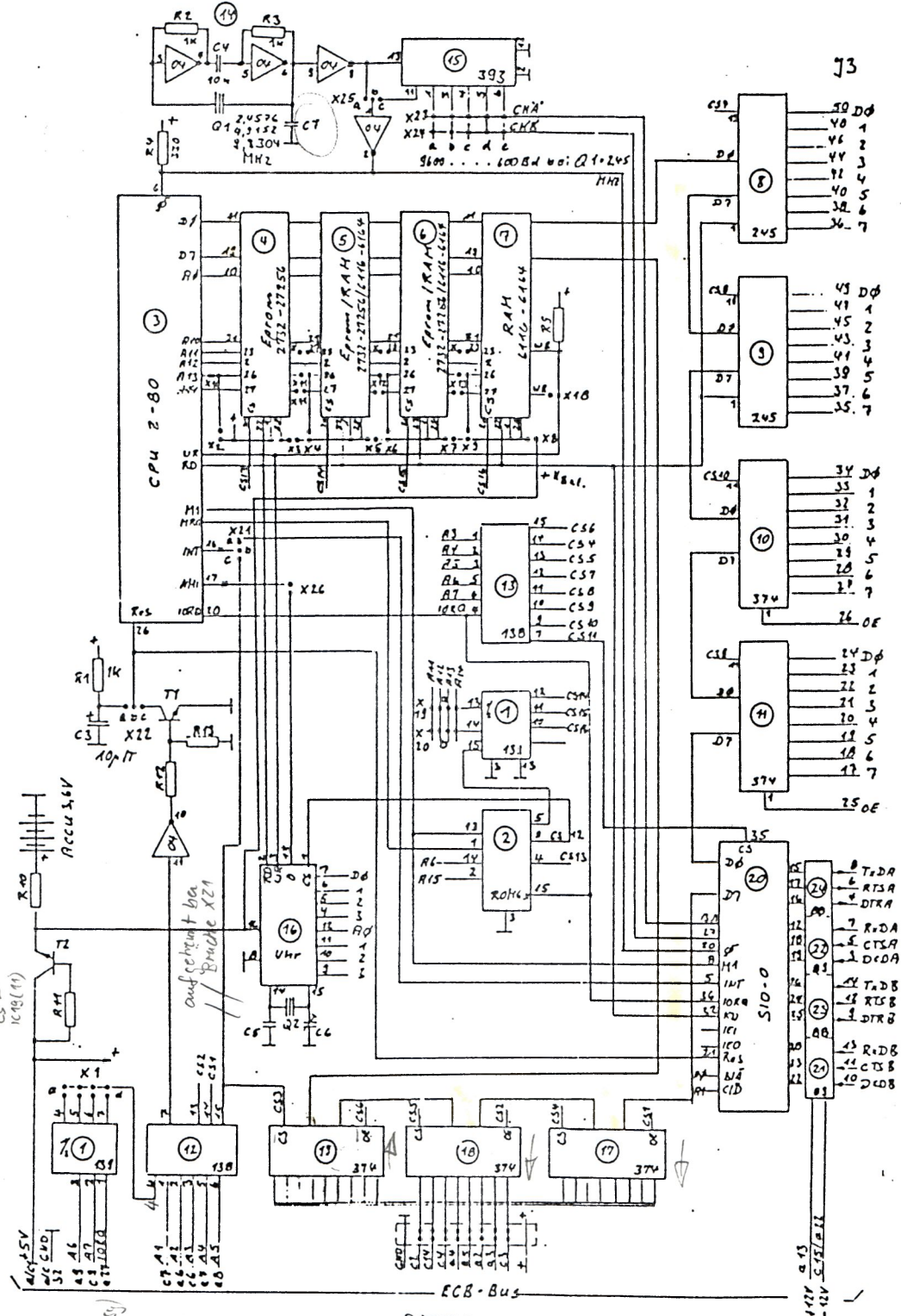
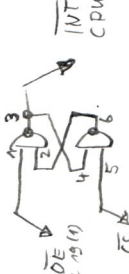
Schreibe 02 (ready) in Chip 17  
(Adr. 88, → CS4)

Warte auf Interrupt

Schreibe 00 (not ready) in Chip 17

20

74LS00 (Hexaversic auf IC 14)  
 für Synchronisation  
 CPU (16)



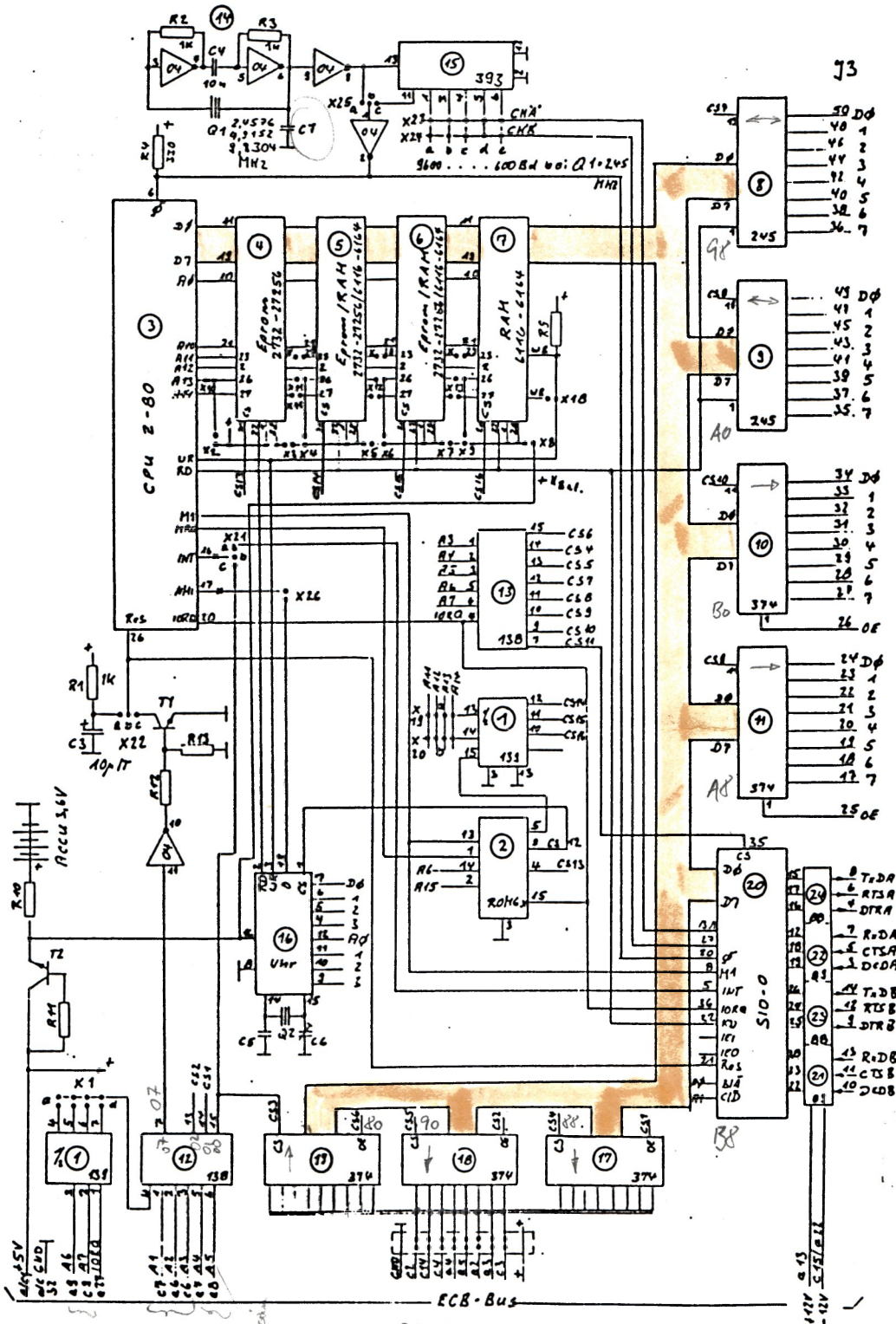
10b...  
 A  
 A1  
 A3  
 14  
 (Ready, Busy)

20  
 20  
 20  
 20  
 20  
 20  
 20  
 20



LS138 1of8-Decoder  
 LS139 Dual 1of4-Decoder

No



bidir.

bidir.

Output Latch

Output Latch

4 - Addr  
 Low Addr.  
 2.0 Hi-Addr  
 1.0 Hi-Addr

ECB-Bus  
 D0  
 D1  
 D2  
 D3  
 D4  
 D5  
 D6  
 D7