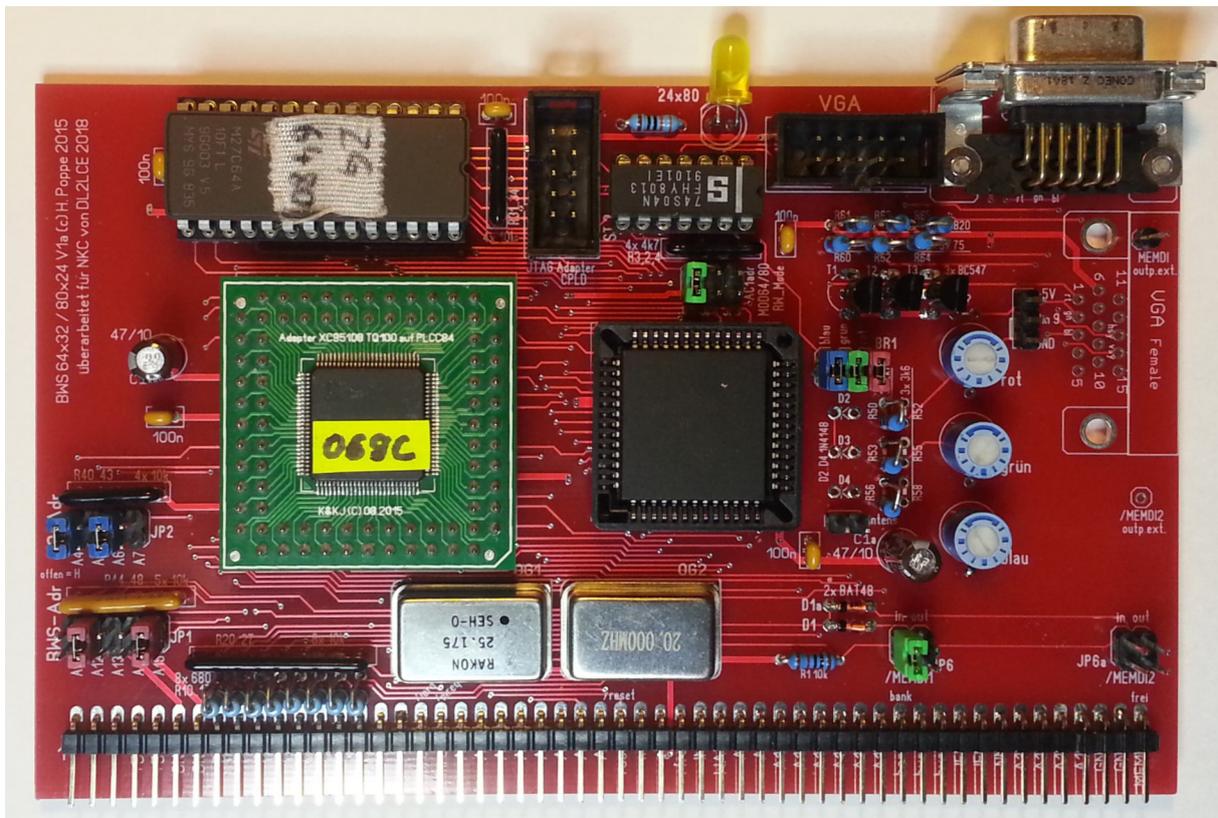


# BWS – NKC

nach der Kombi\_BWS  
von Heiko Poppe  
für den NKC angepasst  
von DL2LCE



Die BWS-NKC ist eine Bildschirm-Karte, die in einen Bildwiderholpeicher geschriebene ASCII-Zeichen, direkt und sofort auf dem Bildschirm darstellt. Eine grafische Darstellung ist nur über einige pseudografische Zeichen im ASCII-Zeichensatz möglich.

Die Ausgabe erfolgt über einen VGA-Anschluss mit einer Auflösung von 800x600 Pixeln.

Der Vorteil dieser Bildschirm-Karte liegt darin, dass die Darstellung von Text, Zahlen und Zeichen in Echtzeit erfolgt und die jeweiligen Zeichen nicht erst in Form, Größe und Position über einzelne Unterprogramme einem Grafikchip mitgeteilt werden muss. Zum Darstellen eines Zeichens auf einer bestimmten Schreibposition genügt ein einfaches Beschreiben des BWS-Speichers. Damit können auch das Rücklesen der Bildschirmzeichen und das Scrolling ohne merklichen Zeitverzug erfolgen.

Im Folgenden zunächst die originale Kurzbeschreibung von H. Poppe:

## Kurzbeschreibung für den Kombi-BWS mit 80x24 und 64x32 Darstellung

### Version 1a

Auf der Grundlage des Color-BWS für den AC1 wurde eine BWS-Karte entwickelt, die in verschiedenen Rechnern eingesetzt werden kann. Sie ist zwischen dem 64x32 Modus und 80x24 Modus umschaltbar und liefert ein VGA-Signal, passend für TFT-Monitore. Dazu gibt es die bereits beim AC1 und bei der ersten Auflage erprobte Farberweiterung.

#### **Die BWS-Karte besitzt diese Eigenschaften:**

- 2k Bildspeicher mit Dualport-RAM, optionale Farberweiterung wie beim AC1-Color-BWS
- Basisadresse der Karte in 2k-Schritten beliebig einstellbar über JP1
- die oberen 4 Bit der I/O-Adresse sind mittels JP2 einstellbar
- Auswahl von Schreib/Lese-Modus oder nur Schreibmodus über JP3
- Umschaltung der Anzeige 64x32 oder 80x24 mit OUT-Befehl
- Auswahl des Anzeigemodus nach dem Einschalten oder Reset über JP4
- BWS-Adressierung bei 64x32 zwischen normal und invertiert mittels JP5 umschaltbar
- Deaktivierung der Schaltung über ein externes /MEMDI-Signal möglich
- alternativ liefert die Schaltung ein /MEMDI-Signal zum Ausblenden des Speichers, die Auswahl der Funktion erfolgt über JP6
- ein Zeichensatz für 80x24 Darstellung und zwei Zeichensätze für 64x32 Darstellung
- Zeichengenerator ist ein Eprom 2764-150ns
- Leiterplattenformat 170x95mm mit EGS-Steckverbinder für K1520-Bus
- VGA-Signal im 80x24-Modus 640x480@60 Hz, im 64x32-Modus 800x600@60 Hz

#### **Hinweise zum Einsatz:**

Die Schaltung ist nicht vollständig zum AC1-BWS kompatibel, es fehlt die Hardware-Invers-Steuerung. Wer diese nicht braucht, kann die Karte auch im AC1 einsetzen. Beim Ausbau mit Farberweiterung ist eine Invers- und Intensivdarstellung über Software möglich. Für CP/M muss dazu das BIOS angepasst werden.

Für den Monochrombetrieb ist ein vereinfachter Aufbau möglich. Die IC 5,6 und 7 entfallen, die Umschaltung erfolgt über JP7. Bei Monochrombetrieb können die Ausgangsstufen für rot und blau entfallen.

Die Karte benötigt für den Betrieb ein /IORQ- und ein /Reset-Signal.

Bei der Neuauflage 2015 gibt es einige kleine Veränderungen. Ein Layoutfehler am JP6 wurde beseitigt und am Pin 10 der Ausgangsbuchse liegen +5V, die für eigene Erweiterungen genutzt werden können. Das Layout der Leiterplatte wurde so verändert, dass die Bestückung der Widerstände in verschiedenen Varianten erfolgen kann (siehe Schaltplan und Stückliste). Die Funktion der Karte und die Programmierung des CPLD sind identisch zur Version 2013.

## Beschreibung Kombi-BWS 64x32 / 80x24

Der Kombi-BWS wurde ursprünglich für den AC1 entwickelt, um hier für CP/M eine 80x24 Darstellung zu ermöglichen. Die Funktionen des 80x24-Modus wurden nach dem Vorbild der Video3-Karte vom NANOS-System übernommen. Durch den universellen Aufbau ist die Schaltung auch in anderen Rechnern einsetzbar. Beim BWS ist die gesamte Logik außer RAM, Zeichengenerator und Farbausgabe in einem CPLD enthalten.

Der BWS stellt einen RAM von 2 kByte dar, welcher mittels JP1 in 2k-Schritten über den gesamten Adressbereich einstellbar ist. Ein gesetzter Jumper bedeutet Low für das jeweilige Bit. Für den Adressbereich F800h bis FFFFh sind alle Jumper offen.

Die Adressierung im Modus 64x32 erfolgt entweder AC1-typisch rückwärts (die höchste BWS-Adresse ist links oben) oder vorwärts, das ist mittels Jumper JP5 umschaltbar. Dabei werden immer 32 Zeilen mit je 64 Zeichen dargestellt. Im Modus 80x24 erfolgt die Adressierung immer vorwärts. Der vom BWS belegte Adressbereich ist bei beiden Modi identisch.

In der Farbversion kommt zum normalen RAM (Text-RAM) ein weiterer RAM, der Farbspeicher (Farb-RAM), hinzu. Er liegt parallel zum Text-RAM im gleichen Adressbereich und wird durch Setzen von Bit 0 im I/O-Register statt dem Text-RAM für die CPU sichtbar. Bit 0 entscheidet also, welcher RAM gerade aktiv gelesen/geschrieben werden kann. Für jedes Byte im Text-RAM gibt es an gleicher Adresse ein FarbByte, welches die Vordergrund- und Hintergrundfarbe bestimmt. Es sind damit 7 Farben + INTENS jeweils für Vorder- und Hintergrund möglich.

Bit 0 Zeichen ROT	Bit 4 Hintergrund ROT
Bit 1 Zeichen GRÜN	Bit 5 Hintergrund GRÜN
Bit 2 Zeichen BLAU	Bit 6 Hintergrund BLAU
Bit 3 Zeichen INTENSIV	Bit 7 Hintergrund INTENSIV

Für die Bilddarstellung werden Text-RAM und Farb-RAM parallel ausgelesen. Wird vom ZG ein Hell-Signal geliefert, werden Bit 0 - 3 ausgegeben, bei einem Dunkel-Signal werden die Bits 4 - 7 zur Anzeige geschaltet. Dieses Auslesen läuft für den Programmierer völlig transparent.

In der Monochrom-Version wird über die Verbindung BR1A-BR1B das das Hell-Dunkel-Signal direkt an die Ausgangsstufe für Grün geschaltet.

Der Betrieb des BWS-Speichers im Rechner kann auf verschiedene Weise erfolgen. Bei vielen CP/M-Systemen wird der BWS-Speicher nur beschrieben und liegt ständig aktiviert parallel zum RAM. Für diesen Fall ist der BWS in den Nur-Schreib-Modus zu schalten, JP3 ist dabei offen. Hierbei ist ein Rücklesen des BWS-RAM (auch des Farb-RAM) nicht möglich, zurückgelesen wird immer der Wert aus dem RAM. In diesem Fall wird die MEMDI-Steuerung nicht benötigt, JP6 bleibt offen.

Beim AC1 wird der BWS mittels /MEMDI-Signal gesteuert und nur beim Schreib/Lesezugriff eingeschaltet, der parallel liegende RAM wird gleichzeitig deaktiviert. Hier ist JP3 geschlossen und bei JP6 1-2 geschlossen. Ein Rücklesen des BWS-RAM und des Farb-RAM ist abhängig von Bit 0 im I/O-Register möglich. Die dazu nötige MEMDI-Steuerung muss außerhalb des BWS vorhanden sein.

Als dritte Möglichkeit kann der CPLD ein /MEMDI-Signal liefern, sobald der für den BWS gewählte Adressbereich angesprochen wird. Mit Hilfe dieses Signals kann ein parallel

liegender RAM für die Zeit des BWS-Zugriffs deaktiviert werden. Diese Möglichkeit ist an einem NANOS-System erfolgreich praktisch erprobt worden.

Für die Steuerung der BWS-Funktionen befindet sich ein rücklesbares I/O-Register im CPLD. Dieses belegt 2 aufeinanderfolgende Adressen. Die oberen 4 Bit der Adresse werden über JP2 eingestellt, ein gesetzter Jumper bedeutet Low für das jeweilige Bit.

Die Umschaltung zwischen der 64x32 und der 80x24 Darstellung erfolgt über das I/O-Register im CPLD. Mit JP4 wird der gewünschte Zustand nach dem Einschalten oder nach Reset festgelegt. Die Umschaltung während des Betriebes erfolgt durch Setzen/Rücksetzen des Bit 1 im I/O-Register. Die LED leuchtet bei aktiviertem 80x24 Modus.

**Hinweis:** Mit der Umschaltung des Anzeigemodus muss auch die Ausgaberroutine für den Bildschirm umgestellt werden!

## Zeichengenerator

Für die 64x32 Darstellung wird ein 6x8 Zeichensatz vergleichbar wie beim AC1 verwendet. Damit ist eine Darstellung von Pseudografik möglich. Der Originalzeichensatz vom AC1 kann jedoch nicht unverändert verwendet werden, die Ausleserichtung der Zeichen ist genau umgekehrt. Für den 64x32 Modus passen zwei Zeichensätze in den ZG-Eprom. Diese sind mittels Bit 3 des I/O-Registers umschaltbar. Beim Einschalten bzw. nach Reset wird immer der Zeichensatz 1 ausgewählt.

Für die 80x24 Darstellung wird ein 8x8 Zeichensatz (z.B. NANOS Video3) verwendet. Die Zwischenräume zwischen den Zeilen (je zwei Pixelzeilen) sind standardmäßig dunkelgetastet. Für die Darstellung von Pseudografik kann diese Dunkeltastung durch Setzen von Bit 2 im I/O-Register aufgehoben werden, der Zeichengenerator muss dazu entsprechend ergänzt werden. Für den 80x24 Modus passt ein Zeichensatz plus Ergänzung für die Zwischenräume in den ZG-Eprom.

### Programmierung des Zeichengenerator-Eprom:

0000h - 07FFh	erster Zeichensatz für 64-Zeichen-Modus
0800h - 0FFFh	zweiter Zeichensatz für 64-Zeichen-Modus
1000h - 17FFh	Zeichensatz für 80-Zeichen-Modus (Pixelzeilen 1 - 8)
1800h - 1FFFh	Zeichensatz Teil 2 für 80-Zeichen-Modus (Pixelzeilen 9 und 10)

## Monochrom oder Farbe?

Die Entscheidung, welche Variante aufgebaut wird, muss jeder selbst treffen. An dieser Stelle sollen nur ein paar Hinweise zur Entscheidungsfindung gegeben werden.

Für einen sinnvollen Betrieb des Color-BWS muss auf jeden Fall das Betriebssystem bzw. das Monitorprogramm erweitert werden. Wer das nicht tun möchte oder keine Möglichkeit dazu hat, sollte erst einmal die Monochromversion aufbauen.

Die minimale Voraussetzung für den Einsatz des Color-BWS besteht darin, dass der Farb-RAM beim Einschalten des Rechners initialisiert, also mit einem sinnvollen Wert für die Zeichen- und Hintergrundfarbe gefüllt wird. Danach kann er wie ein Monochrom-BWS betrieben werden. Wer eine „echte“ Farbausgabe möchte, muss die Ausgaberroutine für den Bildschirm erweitern und natürlich über entsprechende Software verfügen, die das auch nutzen kann. Ein Beispiel, wie das im CP/M beim AC1 gelöst werden kann, wird im beiliegenden Quelltext-Schnipsel gezeigt.

**Hinweis:** Dieser Quelltext bedient nur den 64x32-Modus!

## Hinweise zum Aufbau des BWS

Der BWS ist für den Einsatz in unterschiedlichen Umgebungen konfigurierbar.

Für den Aufbau der Color-Version wird die Leiterplatte komplett bestückt (\*).

Beim Aufbau der Monochrom-Version entfallen die IC 5, 6 und 7. Weiterhin entfallen D2 bis D4, R55 und die kompletten Ausgangsstufen für rot und blau. JP 7 wird auf mono gesetzt.

(\*) Die Widerstände R20 - R27 (RN4) sind Pull-Up-Widerstände für den Datenbus und nur bei Bedarf zu bestücken. Sie sind nur 1x im gesamten System notwendig und u.a. dann nützlich, wenn z.B. der I/O-Port des BWS ausgelesen werden soll.

Die Widerstände R10 - R17 (RN5, RN6) zwischen CPLD und Datenbus sind unbedingt notwendig, um Reflexionen auf dem Datenbus zu verhindern. Ihr Wert beträgt bei mir 680 Ohm und kann eventuell rechnerabhängig sein.

Die Widerstände R50 bis R57 (RN7, RN8, RN9) und R60 bis R65 (RN10, RN11) bestimmen die Helligkeit des Bildes auf dem VGA-Monitor. Die hier angegebenen Werte sind von mir ausprobiert worden, müssen aber nicht in jedem Fall ein optimales Ergebnis liefern.

Die Widerstände können alternativ als Einzelwiderstände, als Widerstandsnetzwerke oder auch mit SMD (auf den normalen Lötäugen) bestückt werden (siehe Stückliste).

Als Zeichengenerator (IC2) muss ein Eprom 2764 mit maximal 150 ns Zugriffszeit eingesetzt werden. Langsamere Eproms führen zu Störungen bei der Bildausgabe.

Für IC6 ist unbedingt ein ACT einzusetzen. Baureihen wie LS oder HCT sind zu langsam und verursachen eine verwaschene Bildausgabe.

Die beiden Quarzoszillatoren müssen immer bestückt werden, auch wenn nur ein Modus benutzt werden soll.

Wer seinen CPLD nicht selbst programmiert, kann ST3 weglassen. Die Widerstände R31 bis R34 (RN3) müssen immer bestückt werden.

### Übersicht der Jumper (\*\*)

JP1: Auswahl der Basisadresse des BWS: JP geschlossen = Bit auf Low

JP2: Einstellung der oberen 4 Adressbit des I/O-Registers: JP geschlossen = Bit auf Low

JP3: offen = BWS nur Schreibmodus, geschlossen = Schreib/Lese-Modus

JP4: Anzeigemodus nach Reset: offen = 80x24, geschlossen = 64x32

JP5: Adressierung im 64x32 Mode: offen = normal, geschlossen = AC1

JP6: MEMDI-Steuerung: 1-2 = Eingang, 2-3 = Ausgang, offen = ohne MEMDI

(\*\*) Wird der BWS in einer fest definierten Umgebung betrieben, können die gesetzten Jumper durch Drahtbrücken ersetzt werden. Die Pull-Up-Widerstände bei geschlossenen Brücken können entfallen.

### Belegung des I/O-Registers:

Bit 0: Umschaltung Zeichen-RAM / Farb-RAM: L = Zeichen, H = Color-RAM

Bit 1: Umschaltung 64/80 Zeichen-Mode: L = 64 Zeichen, H = 80 Zeichen

Bit 2: Zeilenzwischenraum im 80x24 Mode: L = dunkel, H = Pseudografik möglich

Bit 3: Umschaltung Zeichensatz für Mode 64x32: L = Zeichensatz 1, H = Zeichensatz 2

### Zugriff auf das I/O-Register:

Lesen aller 4 Bit des Registers über Adresse 0X2h

Schreiben der Bits 3, 2, 1 über die Adresse 0X1h (Steuerung des BWS)

Schreiben des Bit 0 über die Adresse 0X2h (Umschaltung Zeichen-RAM / Farb-RAM)

( X = mit JP2 eingestellter Wert)

# Stückliste für Version 1a

Stand 24.09.2015

IC1	CPLD XILINX	XC95108PC84
IC2	Eprom 8k x 8, 150ns	2764-150 oder 27C64-150
IC3, IC5	Dualport-RAM 2k x 8	IDT71321 oder CY7C136
IC4	6x Inverter	74LS04 oder DL004
IC6	4x 2 zu 1 Multiplexer	74ACT257
IC7	8 Bit Latch	74LS374
für IC1:	IC-Fassung PLCC-84	
für IC3, IC5:	IC-Fassung PLCC-52	
T1 bis T3	Si npn Transistor	z.B. BC547
D1	Schottky-Diode	BAT48
D2 bis D4	Si-Diode	z.B. 1N4148
LED1	irgendeine LED	
QG1	Quarzoszillator 14/4	25,175 MHz
QG2	Quarzoszillator 14/4	20,000 MHz
JP1	Jumperblock 2x5	
JP2	Jumperblock 2x4	
JP3 bis JP5	Jumper 1x2	
JP6, JP7	Jumper 1x3	
ST1	Systembus	EFS-Stecker 58-polig
ST2	VGA-Ausgang	Steckerwanne 10-polig
ST3	JTAG-Prog.-stecker	Steckerwanne 10-polig (*B)
R1	Widerstand 1 St.	10k (*RN1)
R2, R3	Widerstand 2 St.	10k
R4	Widerstand 1 St.	10k (*RN2)
R9	Widerstand 1 St.	330 Ohm
R10 bis R17	Widerstand 8 St.	680 Ohm (*B) (*RN5, RN6)
R20 bis R27	Widerstand 8 St.	10k (*B) (*RN4)
R31 bis R34	Widerstand 4 St.	10k (*RN3)
R40 bis R43	Widerstand 4 St.	10k (*RN1)
R44 bis R48	Widerstand 5 St.	10k (*RN2)
R50, R53, R56	Widerstand 3 St.	2,7k (*RN7)
R51, R54, R57	Widerstand 3 St.	2,7k (*RN9)
R52, R55, R58	Widerstand 3 St.	3,6k (*RN8)
R60, R62, R64	Widerstand 3 St.	75 Ohm (*RN11)
R61, R63, R65	Widerstand 3 St.	820 Ohm (*RN10)
C1, C2	Elko RM 2,54	100µF oder 47µF
C3 bis C9	Kondensator	100nF oder 68nF

(\*B) Siehe Beschreibung

## Alternative Bestückung der Widerstände:

*RN1	R-Netzwerk SIL 6-5
*RN2	R-Netzwerk SIL 7-6
*RN3	R-Netzwerk SIL 5-4
*RN4	R-Netzwerk SIL 9-8
*RN5, RN6	R-Netzwerk SIL 8-4
*RN7, RN8	R-Netzwerk SIL 6-3
*RN9	R-Netzwerk SIL 5-4
*RN10, RN11	R-Netzwerk SIL 6-3

## Abweichungen / Änderungen in der BWS-NKC

- Die BWS-NKC verwirklicht nur die monochrome Variante der originalen Baugruppe.
- Die Bildschirmfarbe kann mittels der drei Trimpotis stufenlos in jede beliebige Farbe eingestellt werden (z.B. weiß / grün / bernstein).
  - weiß **alle Trimpotis auf Mittelstellung**
  - Bernstein: rot – Mittelstellung  
blau – Linksanschlag  
grün – aus Mittelstellung in Richtung Linksanschlag
- Zusätzlich wurde für den VGA-Anschluss Platz für eine VGA-Buchse wahlfrei am oberen oder am rechten Rand der Platine vorgesehen. Pin 9 der VGA-Buchse wird üblicher Weise an GND gelegt. Manche Monitore wollen dort jedoch +5V haben, um den Anschluss an einen PC zu erkennen. Daher kann Pin 9 der VGA-Buchse über einen Jumper wahlfrei an GND oder +5V gelegt werden (Empfohlen wird GND).
- Die Baugruppe hat ein **MEMDI\_in**- und ein **MEMDI\_out**-Signal.  
Das **MEMDI\_in** – Signal auf *high* trennt den BWS-RAM vom BUS, sofern das IO-Bit-0 auf *high* gesetzt wurde. Um den BWS-RAM vom Bus zu trennen muss entweder das IO-Bit-0 ODER das **MEMDI\_in** – Signal auf *low* stehen.

Das **MEMDI-out** – Signal ist *lowaktiv*, wenn sich der BWS-RAM vom BUS her angesprochen „fühlt“.

Diese beiden Signale können jeweils wahlfrei auf das BUS-Signal „**banken**“ = „/**MEMDI1**“ und „**frei**“ = „/**MEMDI2**“ gelegt werden.

Auf die empfohlene Einstellung wird unter empfohlene Betriebsmodi eingegangen.

### Neue CPLD-Programmierung von J. Mewes vom Okt.2018

- Es wird nur noch eine IO-Adresse auf x7h zum Lesen und Schreiben verwendet. Damit kann z.B. die meist noch freie **IO-Adresse A7h** als Standard für diese Baugruppe verwendet werden.

Das Bit#0 = 1 auf diesem IO-Port bewirkt NUR zusammen mit dem MEMDI\_in = *high* eine Trennung des BWS-RAM von NKC-BUS (weiteres unter empfohlene Betriebsmodi).

In einem System mit mehreren RAM-Bänken (BankBoot + 1024SRAM) sollte das **MEMDI-in** – Signal mit dem BUS-Signal **banken** (MEMDI1) verbunden werden, und das IO-Bit-0 auf *high* gesetzt sein. Somit wird der BWS-Ram nur geschrieben, wenn auch die BankBoot aktiv ist.

- Das fehlende 1. Pixel pro Zeile wird nun dargestellt.

### Anmerkung:

- Der schwer beschaffbare Oszillator OG1 von 25,175 MHz kann durch einen mit 25,000 MHz ersetzt werden.

## /MEMDI1 und /MEMDI2

Im Zuge der Doppelbelegung von Speicherbereichen durch einen Bildwiderholtspeicher (BWS\_RAM) ist es u.U. Notwendig den ursprünglich vorhandenen Speicher im Falle des BWS-RAM Zugriffes auszublenden.

Für den RAM auf der ROA64 oder der SRAM1024 erfolgt die Ausblendung bereits über das bisherige Signal „**banken**“ = *low* durch die **BankBoot**. Damit kann dieses Signal auch hier zur Anwendung kommen.

Soll/muss (evtl. AC1\_Mode) der BWS\_RAM im Adressbereich liegen, der bereits durch die **BankBoot** belegt ist, greift das „**banken**“-Signal nicht um den Speicher der **BankBoot** (RAM oder EPROM) auszublenden zu können. Hierfür wird auf der neuen **BankBoot2** das Signal **/MEMDI2** – welches auf Pin 54 des NKC-Busses = „frei“ gelegt wird - eingeführt.

Damit wird im Folgenden, analog zu anderen BUS-Systemen, folgende Bezeichnung gewählt:

**/MEMDI1** = *MEMoryDisable1*, das bisherige Signal „**banken**“ schaltet lowaktiv den RAM des NKC auf inaktiv

**/MEMDI2** = *MEMoryDisable2*, schaltet lowaktiv den Speicher der **BankBoot** inaktiv

## Empfohlene Betriebsmodi

1. Wenn der Adressbereich des BWS-RAM (Bildwiderholtspeicher) im NKC NICHT mit Hardware belegt ist, MUSS der Jumper **RW\_Mode** gesetzt werden, damit der BWS-RAM vom BUS her gelesen und beschrieben werden kann.
2. Wenn die BWS-NKC in einem NKC ohne **BankBoot** verwendet wird, oder das jeweils laufende System immer nur eine Bank verwendet, sollte der Jumper **RW\_Mode** offen gelassen werden. Damit wird der BWS-RAM vom BUS her ausschließlich schreibend angesprochen. Das bedeutet, dass beim Schreiben in den Bildwiderholtspeicher der BWS-RAM parallel zum Arbeitsspeicher beschrieben wird. Gelesen wird hingegen immer NUR der Arbeitsspeicher, der ja den gleichen Inhalt wie der BWS-RAM hat.
3. ...alternativ in der vorgenannten Konstellation, könnte der Jumper **RW\_Mode** gesetzt werden und das Signal **MEMDI\_out** mit **/MEMDI1 = **banken**** verbunden werden. Damit wird der Arbeitsspeicher (RAM!) hardwaremäßig ausgeblendet, wenn die CPU auf den BWS-RAM zugreift. In diesem Falle kann der BWS-RAM von der CPU gelesen und beschrieben werden.
4. Wird eine BankBoot eingesetzt UND laufen Betriebssysteme (CP/M), die mehrere Bänke verwenden, dann darf der BWS-RAM nur zusammen mit der BankBoot-Baugruppe für den BUS-Zugriff aktiviert werden. Würde man in einem solchen System die vorgenannte Variante 2 oder 3 wählen, würde zwar das Betriebssystem in seinem Lauf nicht gestört, aber beim

Zugriff z.B. auf eine RAM-Floppy in anderen Bänken, würde sich der BWS-RAM ebenfalls angesprochen fühlen und mit beschrieben werden. Damit würde der Bildschirm u.U. Blödsinn anzeigen.

Das gelingt per Software über das Bit#0 im IO-Port (z.B. A7h) der BWS-NKC, das auf 1 gesetzt den Eingang **MEMDI\_in** aktiviert. Dieser muss nun nur noch mit dem BUS-Signal „banken“ (**/MEMDI1**) verbunden werden. Sodann wird der BWS-RAMM immer vom BUS getrennt, wenn nicht der Speicher auf der BankBoot angesprochen wird.

Beim Einsatz im System **GP-EDAS** kann der BWS-RAM auf 6800h gejumpert werden. Dazu muss der Jumper **AC1adr** und **Mod64\_80** gesetzt werden. Der Jumper **BW\_Mode** bleibt im Falle der vorgenannten Betriebs-Variante 2 offen. Damit wird der virtuelle Bildwiderholpeicher des Programmteiles **AC1-EDAS** parallel zur Ausgabe über die **GDP64** auch auf der BWS-NKC angezeigt. Auf den IO-Port der BWS-NKC müssen keine Ausgaben getätigt werden.

Das Titelbild zeigt die Jumper-Einstellung für den Einsatz im GP-EDAS mit CP/M.

### Letzte Änderungen

15.12.2019

Um ein sauberes Bild zu erzeugen mussten die Widerstände R50, R53, R56 auf 1k verringert werden! Als Transistoren kam der BC547B zu Einsatz. Die Einstellwiderstände konnten auf 5k verringert werden. Die Stellung der Einstellwiderstände auf dem Titelfoto zeigt eine Bernsteineinstellung des Bildschirms. Die Dioden D2, D3, D4 und die Widerstände R52, R55, R58 konnten entfallen.

Viel Spaß beim Einsatz  
wünscht

Steffen, DL2LCE

*Vielen Dank auch an J. Mewes für die Anpassung des CPLD-Programmes und für seine Unterstützung.*

