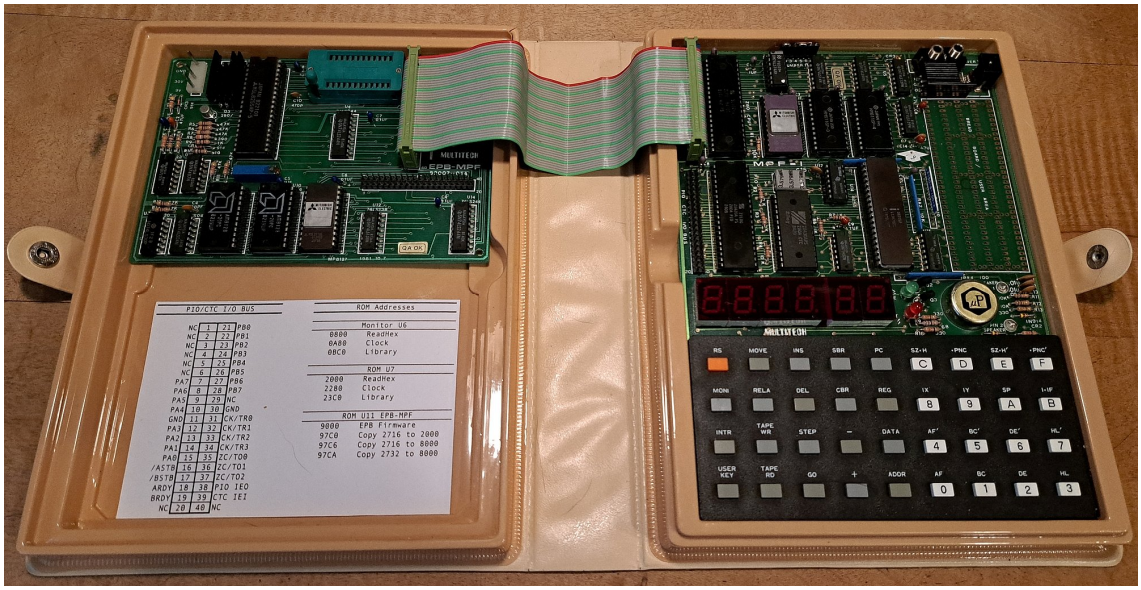


MICROPROFESSOR MPF-1
USAGE NOTES

System

Power Supply

If both, MPF-1 and EPB-MPF are operated with a single powersupply, the MPF-1 type gets very hot and will most likely fail after long usage. It is specified for 9V 600mA, the demand is about 800mA. Use the power supply from PRT-MPF, it is rated for 1A.

I/O Ports

00H	dpPPA	8255 PPI, Port A (keyboard, user key, EAR input)
01H	dpPPB	8255 PPI, Port B (display segments)
02H	dpPPC	8255 PPI, Port C (display digits, key rows, SPK/MIC)
40H	dpC0	Z80-CTC, Channel 0
41H	dpC1	Z80-CTC, Channel 1
42H	dpC2	Z80-CTC, Channel 2
43H	dpC3	Z80-CTC, Channel 3
80H	dpPAD	Z80-PIO, Channel A Data (Christiani printer)
81H	dpPBD	Z80-PIO, Channel B Data
82H	dpPAC	Z80-PIO, Channel A Control
83H	dpPBC	Z80-PIO, Channel B Control
CCH	dpEPA	8255 PPI on EPM-MPF, Port A (ZIF D0-D7)
CDH	dpEPB	8255 PPI on EPM-MPF, Port B (ZIF A0-A7)
CEH	dpEPC	8255 PPI on EPM-MPF, Port C (bits 0-3 ZIF A8-A11, bits 4-7 EPROM control signals)
CFH	dpEPCT	8255 PPI on EPM-MPF, Control

Monitor Routines

05E4	dfMSPK (TONE)	Purpose.....: Call TONE to generate sound Input.......: C: frequency HL: number of periods Output......: None Registers...: AF,BC,DE,HL destroyed
05FE	dfMSC (SCAN)	Purpose.....: Scan keyboard and display, wait for any key pressed, beeps on keypress Input.......: IX: points to the buffer containing display patterns Output......: A: internal code for the pressed key Registers...: AF,B,HL,AF',BC',DE' destroyed
0624	dfMSC1 (SCAN1)	Purpose.....: Scan keyboard and display once, do not wait for key pressed Input.......: IX: points to the buffer containing display patterns Output......: A: position key code if a key was pressed CY: 1= no key pressed 0= key pressed, position key code in A Registers...: AF,AF',BC',DE' destroyed
061D	dfKEYM (KEYMAP)	Purpose.....: Convert key position code returned by SCAN1 to internal code Input.......: A: position code Output......: A: internal code Registers...: AF,HL
0689	dfMH7 (HEX7)	Purpose.....: Convert a binary number to display pattern Input.......: A: bits 0-3 binary value to convert Output......: A: resulting display pattern Registers...: AF destroyed
0678	dfMH7S (HEX7SG)	Purpose.....: Convert a binary number to display pattern Input.......: A: bits 0-3 LSB, 4-7 MSB of binary value to convert HL: where to store the pattern Output......: HL: HL+2 Registers...: AF,HL destroyed

05F6	dfCHKR (RAMCHK)	Purpose.....: Convert a binary number to display pattern Input.....: A: bits 0-3 LSB, 4-7 MSB of binary value to convert HL: where to store the pattern Output.....: HL: HL+2 Registers...: AF,HL destroyed

Library ROM

The library can reside either in U6, instead of Basic at 0BC0 or in U7 at 23C0. A 2532 (U6) or 2732 (U7) EPROM is required. If an application makes use of the library, it must include MPF1.DEF which defines constants and routine addresses.

Interrupt Handlers

0BC0 23C0	lcISV	Purpose.....: Interrupt vector table
0BCC 23CC	lIS	Purpose.....: Setup interrupt vector table and set CPU to int mode 2 Registers...: None saved Labels......: l)ibrary I)nterrupt S)etup
0BE3 23E3	lID	Purpose.....: Check if interrupts are enabled, if so, disable them. Return enabled state, so the caller can enable them again if they were enabled. Output......: CY: 0= ints are not enabled 1= ints were enabled before call Registers...: AF destroyed Labels......: l)ibrary I)nterrupt D)isable
0BEC 23CC	lIPA	Purpose.....: Interrupt handler PIO channel A, increments ints-occurred counter and calls user function if defined Registers...: AF, HL saved, the others are maintained by application Labels......: l)ibrary I)nterrupt handler P)IO channel A)
0C0B 240B	lIPB	Purpose.....: Interrupt handler PIO channel B, increments ints-occurred counter and calls user function if defined Registers...: AF, HL saved, the others are maintained by application Labels......: l)ibrary I)nterrupt handler P)IO channel B)
0C2A 242A	lIC0	Purpose.....: Interrupt handler CTC timer0. Calls user-supplied handler if not NULL, increments int occurred counter which can be checked and reset by application. Registers...: HL saved, the others are maintained by application Labels......: l)ibrary I)nterrupt handler C)TC timer 0)
0C49 2449	lIC1	Purpose.....: Interrupt handler CTC timer1. Calls user-supplied handler if not NULL, increments int occurred counter which can be checked and reset by application. Registers...: HL saved, the others are maintained by application Labels......: l)ibrary I)nterrupt handler C)TC timer 1)
0C68 2468	lIC2	Purpose.....: Interrupt handler CTC timer2. Calls user-supplied handler if not NULL, increments int occurred counter which can be checked and reset by application. This timer is also used for USART baudrate Registers...: HL saved, the others are maintained by application Labels......: l)ibrary I)nterrupt handler C)TC timer 2)
0C87 2487	lIC3	Purpose.....: Interrupt handler CTC timer3. Calls user-supplied handler if not NULL, increments int occurred counter which can be checked and reset by application. Registers...: HL saved, the others are maintained by application Labels......: l)ibrary I)nterrupt handler C)TC timer 3)

Interrupt Variables

1EEB	lvCIC0	CTC timer0 interrupt counter, 1 byte
1EEC	lvCIC1	CTC timer1 interrupt counter, 1 byte
1EED	lvCIC2	CTC timer2 interrupt counter, 1 byte
1EEE	lvCIC3	CTC timer3 interrupt counter, 1 byte
1EEF	lvPICA	PIO channel A interrupt counter, 1 byte
1EF0	lvPICB	PIO channel B interrupt counter, 1 byte

Z80 PIO

0CA6 24A6	lPS	Purpose.....: Initialize one PIO channel, call with interrupts disabled Input.....: D: channel, one of dcIPx 0: channel A 1: channel B E: operation mode, one of dcPMx 00001111B: mode 0, output 01001111B: mode 1, input 10001111B: mode 2, bidirectional (A only) 11001111B: mode 3, control (bit mapped) B: interrupt control, one of dcPIxx 00000111B: disable interrupts 10000111B: enable ints, modes 0-2 11010111B: mode3, AND mask, low state 11110111B: mode3, AND mask, hi state 10010111B: mode3, OR mask, low state 10110111B: mode3, OR mask, hi state A: input/output mask, 1=input, 0=output (mode 3 only) C: interrupt mask (mode 3 only) HL: address of interrupt handler CY: 1= setup PIO 0= disable channel interrupts, set mode 1 (all inputs) Output.....: Interrupts are disabled Registers...: AF destroyed Labels.....: l)ibrary P)IO S)etup

Z80 CTC

0D02 2502	lCS	Purpose.....: Initialize or stop a CTC timer Input.....: D: timer number, 0..3 E: control register contents, one of dcCTx or dcCIx: 00000101B: timer, prescaler 16 00100101B: timer, prescaler 256 01000101B: counter 10000101B: interrupt, timer, prescaler 16 10100101B: interrupt, timer, prescaler 256 10110101B: interrupt, timer, prescaler 256, + trigger 11000101B: interrupt, counter 11010101B: interrupt, counter, positive trigger B: time constant to use or one of dcCOxx: 007H: 1mS 023H: 5mS 046H: 10mS 08CH: 20mS 0D3H: 30mS HL: interrupt handler address, if 0, no call CY: 1= setup timer 0= stop timer and interrupts Registers...: AF destroyed Labels.....: l)ibrary C)TC S)etup

Helper Routines

0D3D 253D	lHDLY	Purpose.....: Delay for given number of 500 microseconds Execution times calculated for 1.7897725 Mhz clock Input.....: DE=Number of 500 microseconds Registers...: All saved Labels.....: l)lbrary H)elper D)eL)aY)
0D4F 254F	lHMF	Purpose.....: Fill memory area with constant Input.....: A : Fill byte HL: Starting address BC: Number of bytes to write Registers...: All saved Labels.....: l)ibrary H)elper M)emory F)ill

0D62 2562	lHHLDE	Purpose.....: Compare HL and DE Input.......: HL, DE to check Output......: SBC HL,DE flags Registers...: F destroyed Labels......: l)ibrary H)elper HL) DE)
0D6B 256B	lHDIV	Purpose.....: Divide and get modulo Input.......: HL: Dividend DE: Divisor Output......: HL: HL % DE, -1 if divisor = zero DE: HL / DE Registers...: HL,DE Destroyed Labels......: l)ibrary U)tility DIV)ide
0D80 2580	lHID	Purpose.....: Check if character is decimal (0..9) Input.......: A: Character to check Output......: CY: 0= It is decimal 1= It is not Registers...: AF destroyed (F only) Labels......: l)ibrary H)elper I)s D)ecimal
0D88 2588	lHIH	Purpose.....: Check if character is HEX (0..9, A..F) Input.......: A: Character to check Output......: CY: 0= It is HEX 1= It is not Registers...: AF destroyed (F only) Labels......: l)ibrary H)elper I)s H)ex

Converting Routines

0D98 2598	lOBH	Purpose.....: Convert binary to hex (0..FFFF), leading zeros always shown Input.......: DE: Value to convert HL: Where to store result C : Number of digits to convert (1..4) Output......: (HL): Converted ASCII string Registers...: All saved Labels......: l)ibrary cO)nvert B)inary to H)ex
0DDC 25DC	lOHB	Purpose.....: Convert HEX ASCII to binary (0..65535) Conversion stops at the 1st non-HEX or C gets zero Input.......: DE: start of ASCII digits C : Number of digits to convert (1..4) Output......: HL: Binary value DE: +number of digits converted CY: 1= overflow or number of digits=0 Registers...: AF,DE,HL destroyed Labels......: l)ibrary cO)nvert H)ex to B)inary
0E12 2612	lOUC	Purpose.....: Convert letter to uppercase Input.......: A: Letter to convert Output......: A: Converted letter Registers...: AF destroyed Labels......: l)ibrary cO)nvert to U)pperC)ase

Display/Keyboard Routines

0E1D 261D	lKBP	Purpose.....: Do a keyboard beep as in monitor. The monitor routine BEEP cannot be used, it jumps to the keydispatcher. Code is taken from monitor routine BEEP at 06CB Registers...: All saved Labels......: l)ibrary K)eyboard B)eeP)
0E37 2637	lKU	Purpose.....: Read status of user key Output......: CY: 1= user key is pressed Registers...: AF destroyed Labels......: l)ibrary K)eyboard U)ser key
0E53 2653	lDM	Purpose.....: Convert ASCII to display pattern, write to display buffer Input.......: DE: address of characters to display, length must be 6 IX: address of display buffer Registers...: AF destroyed Labels......: l)ibrary D)isplay M)essage

0E6B 266B	lDT	Purpose.....: Display a message for a given time Input.......: IX: address of display buffer DE: address of message to display B: time to display the message. 030H ~ 1s CY: 1= DE has the address of ASCII buffer, convert to pattern 0= its already display pattern, no conversion Registers...: All saved Labels......: l)ibrary D)isplay T)imed
0E7F 267F	lDP	Purpose.....: Display a pattern consisting of same characters. Uses direct port output, required in time-critical situations Input.......: A: character(s) to display C: bitmap of digits to use, bit0=rightmost digit Registers...: All saved Labels......: l)ibrary D)isplay P)attern
0E8C 268C	lDPG	Purpose.....: Get display pattern of ASCII character Input.......: A: character to convert Output......: A: display pattern, 0 (=space) if not displayable Registers...: AF destroyed Labels......: l)ibrary D)isplay P)attern G)et
0EF1 26F1	lDK	Purpose.....: Display chars in ASCII display buffer, scan keyboard and user ley. Does not wait for keypress Input.......: DE: Address of characters to display, length must be 6 If 0, no ASCII to pattern conversion is done, (IX) must already contain display pattern IX: Address of pattern display buffer A: Decipoints to display, bit0=rightmost digit CY: 1= update display buffer and decipoints 0= no update, just display and read, DE and A are ignored Output......: A: Pressed key in internal format Z: 1= no key was pressed or key is still pressed 0= a new key was pressed CY: 0= user key not pressed 1= user key was pressed Registers...: AF destroyed Labels......: l)ibrary D)isplay K)eyboard
0F35 2735	lDKN	Purpose.....: Read number from keyboard, update display. Waits for input 'GO' terminates, 'ADDR' restarts, set value to 0, 'USER KEY' aborts Input.......: DE: address of ASCII display buffer, length must be 6 IX: address of pattern display buffer A: digits to use for entry, bit0=rightmost digit CY: 1= accept HEX range keys 0= accept decimal range keys Output......: HL: entered number CY: 0= user key not pressed 1= user key was pressed Registers...: AF,HL destroyed Labels......: l)ibrary D)isplay K)eyboard N)umber

Library Setup

0FD4 27D4	lLS	Purpose.....: Initialize library variables. Must be called once at start of application which uses the library Registers...: All saved Labels......: l)ibrary L)ibrary S)etup

Library Version

0FFE 27FE	lVER	Library version, 2 bytes, minor part first

Serial Receive Utility		
0800 2000	SerRcv	**Allows to download Intel HEX or binary files to the MPF-1. The sender is a regular RS232 with the TX line connected to MPF's EAR input. The MPF has protection circuitry so the +/- voltages will not cause problems. To reduce the load on the TX pin, a series resistor, 2k2 can be inserted. The RS232 has to be set at 300,N,8,2. That slow because the MPF has no UART, data is received using bit-banging. After every hex line there have to be done some checks (checksum, write verify, system RAM overwrite) for all bytes in one line.** **Operation** • **Start the program with 0800 GO (2000 if EPROM in U7 is used). It can be terminated any time with the 'USER KEY'** • `'nnnn rh'` **is shown, where nnnn stands for the version number. Select the mode with '+' key between h)ex b)inary and binary c)hecksum. The current selection is shown in the rightmost digit.** • **Press** `'GO'` • `'0000 -L'` **is shown, allowing to enter a L)oad address. If left at zero, the addresses in the hex file will be used, in binary mode 1800. A check is done if at the given address is RAM and it is not in the system area (1E00-1FFF). Error** `'E6'` **or** `'E4'` **is shown respectively in that case. The address can be reentered. During address input,** `'ADDR'` **can be used anytime to clear to zero.** • **Press** `'GO'` • `'  ..'` **is shown, signals waiting for the first startbit. If it is detected, downloading begins and every hex line or 16 bytes in binary mode the display changes between '_' and '-' to show the progress. During download, the checksum (HEX mode), write success and system RAM overwite attempt are checked. Download stops on error, at end of hex file or silence for about 500mS in binary mode. If binary checksum is active, the last byte received is the checksum, which makes the 8-bit sum of all bytes equal to zero.** • `'nnnn to'` **is shown on sucessful download, where nnnn is the last written RAM address. On error '**`En`**' is shown instead of '**`to`**' and** `nnnn` **is the error address:** `E1`**: No start mark (J found in HEX line** `E2`**: Checksum error. In HEX mode shown immediately after detection, in binary checksum mode after download has finished.** `E3`**: RAM verify error. Either there is ROM or no RAM.** `E4`**: System RAM (1E00-1FFF) overwrite attempt.** `E5`**: Receive buffer full. HEX mode only, a line is longer than 80 characters.** • **Press** `'USER KEY'` **or** `'RS'` **to exit**

Clock Utility		
0A80 2280	Clock	**This is a modified version of the example program in the MPF-1 Operating Manual. It uses CTC interrupts as time base. It requires the library code V01.20 or higher to be present at 0BC0 in 2532 Monitor EPROM U6 or at 23C0 in 2716 U7 EPROM.** **Modifications** • **All variables are initialized at cold start, the clock begins with '00.00.00'. Continue is possible by starting at 0AAC (22AC if in U7 EPROM) and no other program has been run in the meantime which uses the same data addresses. A cold start is done in this case.** • **Moved counter and display update routines from interrupt to main loop.** • **Added a seconds click using the speaker. Can be toggled with the '+' key.** • **Initially, the clock is stopped. Start/stop with 'GO'. Stopped state: no decipoints, running: decipoints shown.** • **The clock can be reset anytime with the '0' key.** **Operation** • **Start at address 0A80 (2280 if using U7 EPROM) and** 'GO' • 'nnnn cL' **is shown for a moment, where nnnn is the program version. This happens on cold starts only.** • '00.00.00' **is shown, press 'GO' to start. Decipoints appear between hour.minute.seconds digits.** • **Use** '+' **key to toggle speaker seconds-click** • **Use** '0' **key to reset clock to zero** • **Use** 'GO' **key to stop/start the clock** • **Terminate at anytime with** 'USER KEY' **or** 'RS' **A warm start can be done at 0AAC (22AC in U7 EPROM) and** 'GO'**. The clock uses the time when it was last terminated unless the data area was destroyed in the meantime, which leads to a cold start.**

EPB-MPF ROM

EPROM U11 on the EPB-MPF contains the programmer firmware at 9000 and the EPROM copy utility.

ROM Read		
97C0	RomRead	**This program requires the EPB-MPF, it's stored in the unused area at the end of the Eprom. It allows to copy an Eprom to RAM with a few keystrokes. It has three entry addresses:** • **97C0: Copy 2732 EPROM to 8000, the RAM on the EPB** • **97C3: Copy 2716 EPROM to 8000** • **97C6: Copy 2716 EPROM to 2000, the RAM in U7**

Christiani Printer ROM

This 2732 EPROM contains printing support for the printer which came with the Christiani Mikroprozessor Labor. The printer is connected to either the PIO port A on the MPF or to the ZIF socket on the EPB-MPF Eprom programmer. It contains Disassember, Hexdump and several printer utilities. The programs can also be run from EPB-RAM at 8000 after loading it using RomRead, SerialReceive or from tape.

Disassembler / Hexdump

2000 2003	Disasm Hexdump	**This is a modified version of the PRT-MPF EPROM, adapted to run in MPFs U7 at 2000** **Modifications** • **A version information is shown on cold startup for half a second (**mmiidA/hd**), where** dA **stands for DisAssembler and** hd **for HexDump.** • **On startup, the printer is searched on EPB-PPI and on MPF-PIO. If it is not connected or turned off,** ' nP' **is shown for a second before returning to the monitor.** • **Default addresses are 1800 for start and end in disassembler, end is 1803 in hexdump to fill a printer line. The start address is incremented at end of print by length +1 for hexdump and end of last printed instruction +1 in disassembler. So a simple** GO **prints the next four hex bytes or the next instruction. If the start address is modified, the end address is adjusted to the the same address in disassembler, start+3 in hexdump, unless the end address was manually changed.** **On program startup the last entered start address is used instead of the default if no other mode has been run in the meantime.** • **After a printout is done, no return to monitor, a new job can be started.** • **A running printjob can be aborted with the** USER KEY**. The addresses are set after the last printed line.** • **Program can be quit with** USER KEY**, if a print job is running, the job is terminated.** • INS **key prints an empty line.** • DEL **key prints five empty lines, so the paper can be teared off without loosing printed lines.** • STEP **key changes between disassembler and hexdump modes.** • **Removed 3rd input parameter, found no use for it. It's stated as reference counter in the manual.** • **Removed initial linefeeds.** • **Removed** 'LF ' **prompt and end of print.** • **On invalid address entry the beeper sounds instead of** Err- **display and returns to input mode.**

Christiani Printer Driver

To be able to use the driver, the application must include CHPDRV.INC**, which defines some constants and the routine addresses. Before assembling, the code address (**2CC5 **for U7,** 8CC5 **for EPB-RAM) must be defined at equate** dwCPCS **in the include file.**

2CC5 8CC5	dfCPIFS	Purpose.....: Check on which interface the printer is connected, then setup this port. Input.....: CY: 1= initialize 0= de-initialize, set all ports as input Output.....: CY: 1= printer not found 0= printer found Registers...: AF destroyed Labels.....: C)hristiani P)rinter I)nterF)ace S)etup
2D31 8D31	dfCPGUK	Purpose.....: Check status of user key Output.....: Z: 1= user key was pressed 0= it was not Registers...: AF destroyed Labels.....: C)hristiani P)printer G)et U)ser K)ey
2D40 8D3D	dfCPCB	Purpose.....: Clear print buffer, fill with spaces Registers...: All saved Labels.....: C)hristiani P)inter C)lear B)uffer

2D97 8D97	dfCPPR	Purpose.....: Print line buffer contents The Christiani printer prints from right to left, the last character first. The buffer is searched for a CR terminator and the characters until CR are counted. Unused space after CR will be printed as spaces, as the printer always prints the whole line. If no CR is found, the whole buffer is printed as it is. Input.....: IY: line buffer address, can also point to ROM CY: 1= clear print buffer after printing 0= do not touch print buffer (e.g. when printing static text from RAM) Output.....: Z: 1= user key was pressed while printing 0= key not pressed Registers...: AF destroyed Labels.....: C)hristiani P)rinter P)R)int
2E19 8E19	dfCPPT	Purpose.....: Print zero-terminated text. Lines may be terminated with CR Input.....: DE: Address of text to print Output.....: Z: 1= user key was pressed 0= it was not Registers...: AF destroyed Labels.....: C)hristiani P)rinter P)rint T)ext
2E4F 8E4F	dfCPLF	Purpose.....: Print line feed (empty line) Registers...: AF destroyed Labels.....: C)hristiani P)rinter L)ine F)eed

Tiny Basic ROM

This 2732 EPROM contains the Tiny-Basic with the same Christiani Printer Driver as in the Printer ROM. The printer ROM is not required to be present. The lister code has been taken from the MPF-PRT ROM. It is linked to 8000, the EPB-RAM. Before use, it has to be loaded using ReadRom at 97C0. This version does not support U7 on MPF-1 (it does not check the presence of Basic in U7).

8000	Basic	**Basic cold start, writes lister-call program to line 999.**
8017	BasicWB	**Basic warm start**
8800	BasicPrt	**Basic Lister start from monitor**

		Modifications of Basic
		On cold start (8000), the basic program
		`998 STOP`
		`999 CALL -2060`
		is written to the Basic program memory to start the lister with `RUN 999`**.**
		Modifications of Lister
		• **On startup from monitor a check is done if basic is present at 8000. If not, the error beep sounds before returning to the monitor.**
		• **On startup from monitor (8800) or from basic (880C) the presence of the printer is checked on the EPB-PPI and the MPF-PIO. If it is not connected or turned off,** `'  nP'` **is shown for a second and it returns to where it came from.**
		• **The first time it is started a version info is displayed for a second.**
		• **While** `'Prt  '` **is shown, the following keys are active:**
		  § `USER KEY (SHIFT)` **returns to either the monitor or basic, depends on how it was started.**
		  § `INS (CLR)` **prints an empty line.**
		  § `DEL (RET)` **prints six empty lines to allow paper tear off without loosing printed lines.**
		  § `STEP (CONT)` **jumps to basic warmstart (8017).**
		  § `HEX` **keys are used to enter the number of lines to print (as in the original eprom).**
		  § `GO (RUN)` **starts printing.**
		• **A running print job can be cancelled with the** `USER KEY`**.**
		• **Initial linefeeds removed.**
		• `'LF  '` **prompt at end of printing removed.**
		• **No automatic return to basic after printing.**

Routines CALLable from Basic

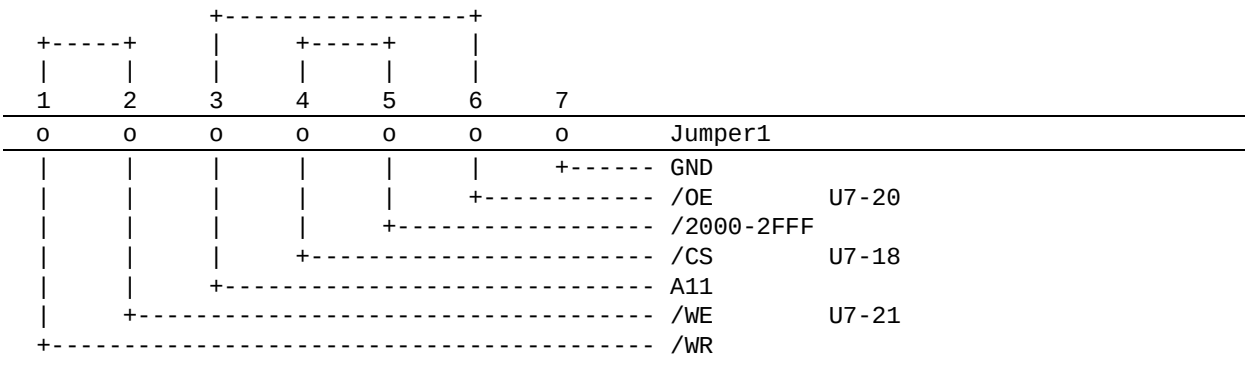
-2051	BPLINE	Purpose.....: Print a text line from a basic program. The print buffer is located at 8026d and is 24 characters long. The interface chip is initialized before and de-initialized after print. Printbuffer contents remain unchanged. To print a line, write the data with LET M 8026+ to the print buffer. The data should be terminated with a CR (13), unless CPCB is called, otherwise the whole buffer is printed. Print the buffer from basic with CALL -2051. Registers...: All saved
-2054	CPCB	Purpose.....: Fill print buffer with spaces, no CR required when printing. Call from basic with CALL -2054. Registers...: All saved
-2057	BCPIF	Purpose.....: Check if the printer is connected. Call from basic with CALL -2057. Output......: (8065) contains the interface ID. Read it with LET <var> = M 8065 LET <var> = <var> ^ 255 (Basic reads two bytes) 0: printer not found 1: printer is on EPB-PPI 2: printer is on MPF-PIO Registers...: All saved

Hardware

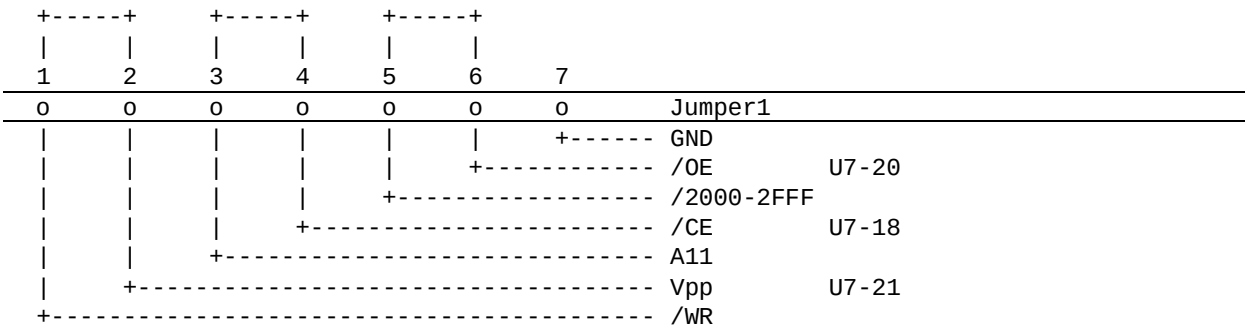
U7 Jumper Settings

6116

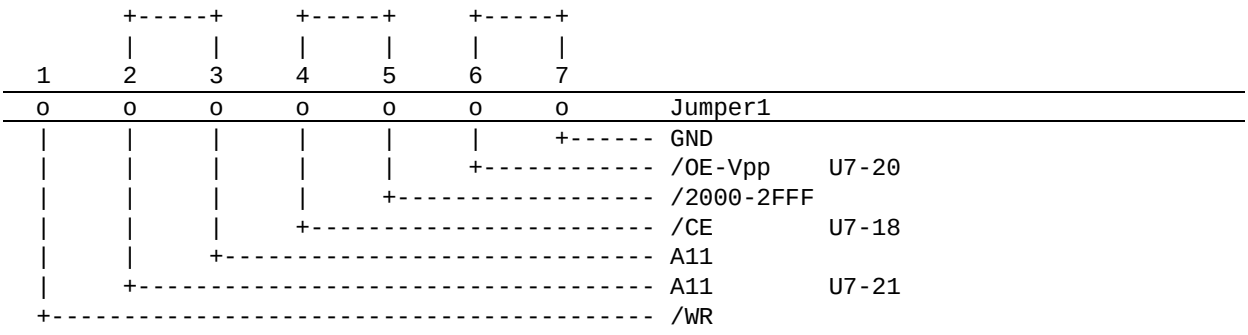
The description in the users manual is not complete for using a 6116 RAM in U7. When jumpering as described (1-2,4-5) the RAM works but it is mirrored at 2800-2FFF. Need to add a connection between pin3 and pin6. /OE will then be inactive if A11 goes high. This solves the mirroring problem.



2716

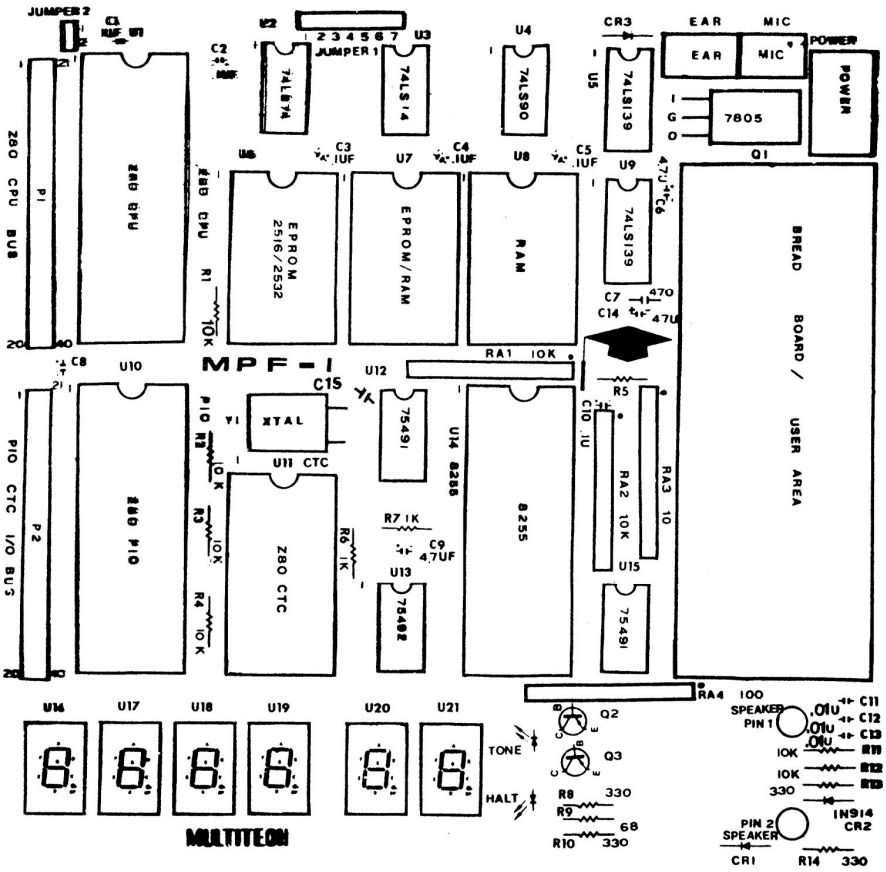


2732

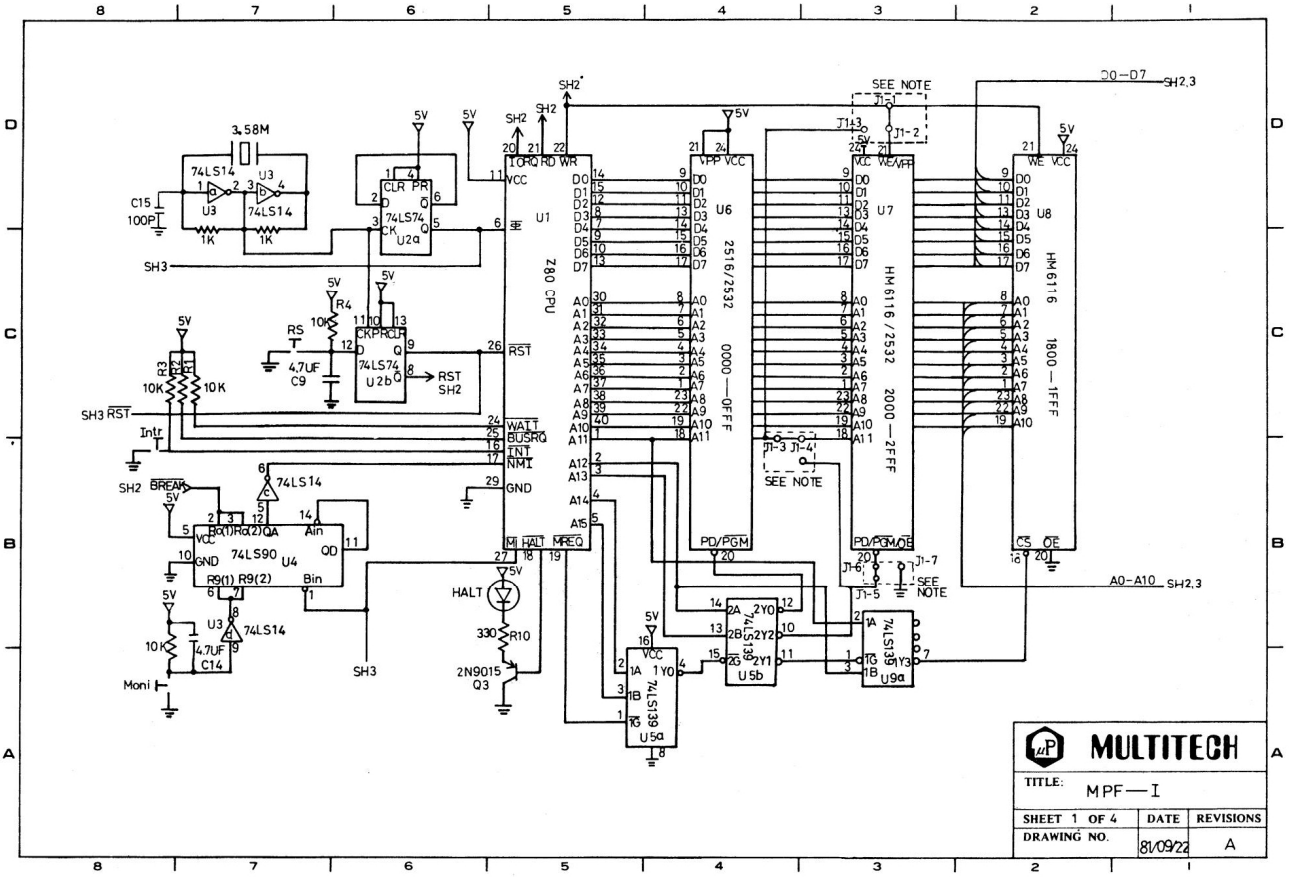


MPF-1 Schematics

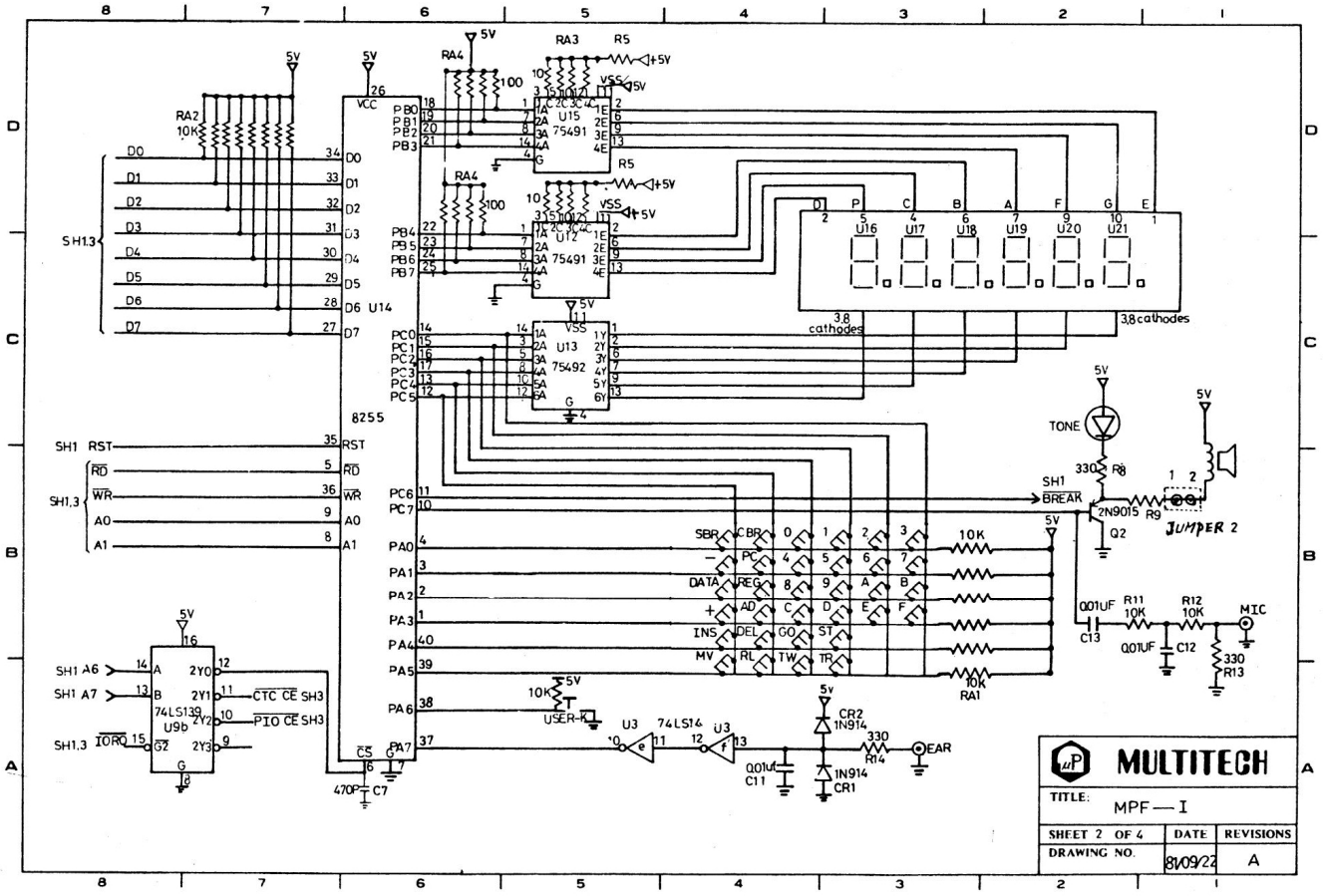
Board Layout



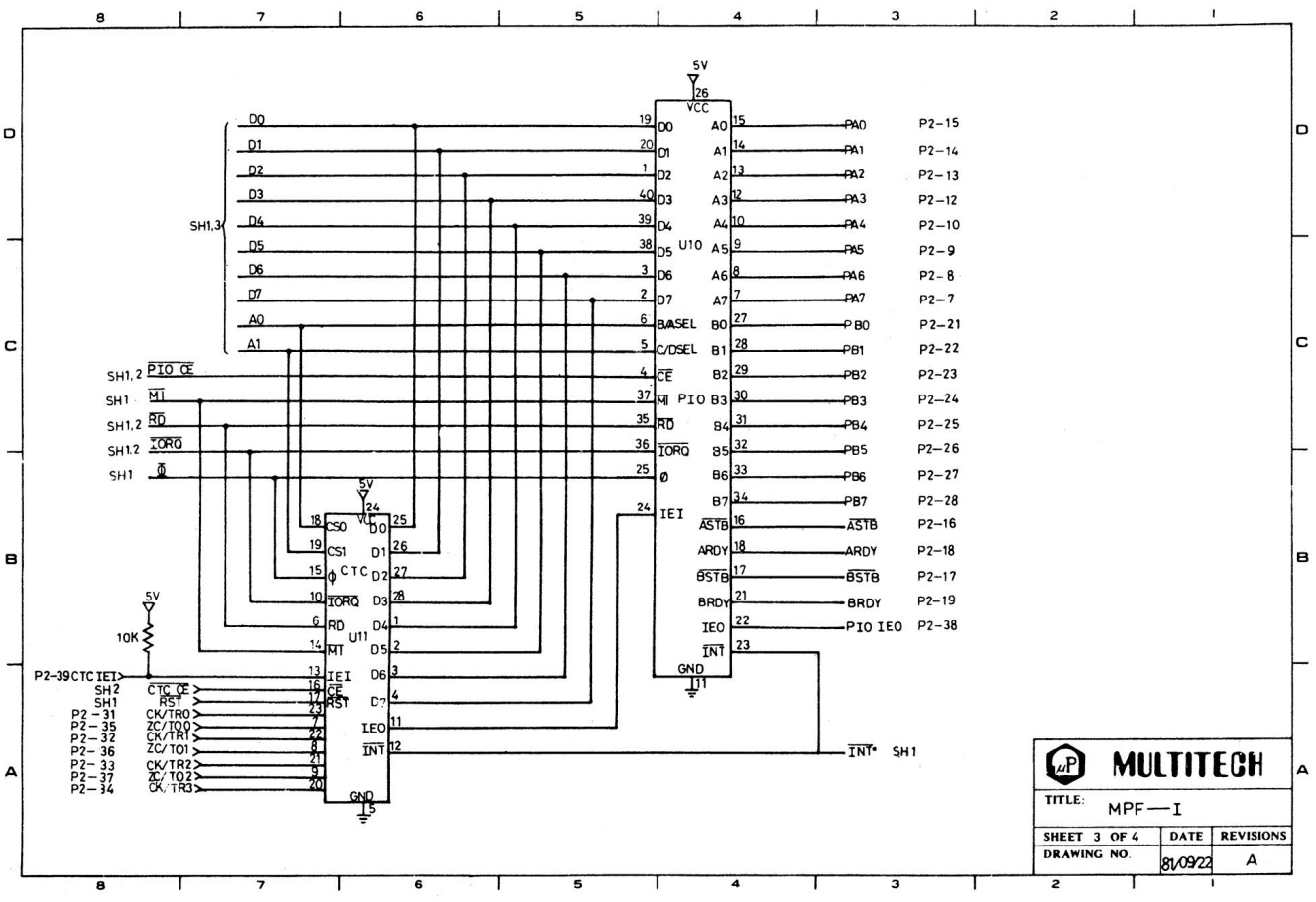
Clock, CPU, Memory



Display, Keyboard, Speaker, Tape I/O



CTC and PIO



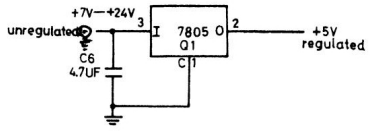
Power Supply, P1/P2 Connectors

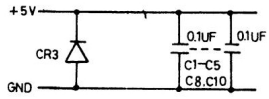
P1 PIN FUNCTION

PIN NO	SIGNAL	PIN NO	SIGNAL
1	A11	21	A10
2	A12	22	A9
3	A13	23	A8
4	A14	24	A7
5	A15	25	A6
6	$\overline{+5}$	26	A5
7	D4	27	A4
8	D3	28	A3
9	D5	29	A2
10	D6	30	A1
11	NC	31	A0
12	D2	32	GND
13	D7	33	$\overline{RFSH}$
14	D0	34	$\overline{MI}$
15	D1	35	$\overline{RESET}$
16	$\overline{INT}$	36	$\overline{BUSRQ}$
17	$\overline{NMI}$	37	$\overline{WAIT}$
18	$\overline{HALT}$	38	$\overline{BUSAK}$
19	$\overline{MREQ}$	39	$\overline{WR}$
20	$\overline{IORQ}$	40	$\overline{RD}$

P2 PIN FUNCTION

PIN NO	SIGNAL	PIN NO	SIGNAL
1	NC	21	PB0
2	NC	22	PB1
3	NC	23	PB2
4	NC	24	PB3
5	NC	25	PB4
6	NC	26	PB5
7	PA7	27	PB6
8	PA6	28	PB7
9	PA5	29	NC
10	PA4	30	GND
11	GND	31	CK/TR0
12	PA3	32	CK/TR1
13	PA2	33	CK/TR2
14	PA1	34	CK/TR3
15	PA0	35	ZC/TO0
16	$\overline{ASTB}$	36	ZC/TO1
17	$\overline{BSTB}$	37	ZC/TO2
18	ARDY	38	PIO IEO
19	BRDY	39	CTC IEI
20	NC	40	NC





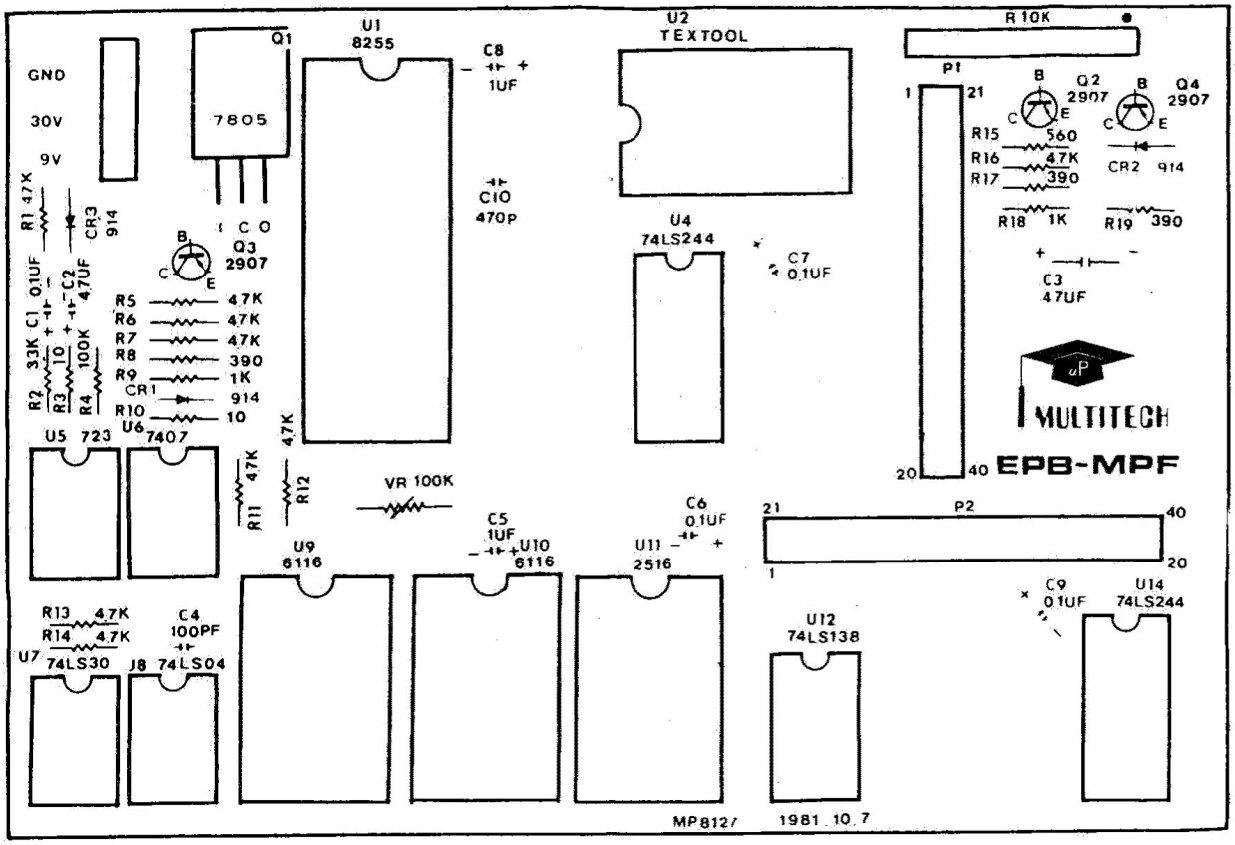
NOTE: 1. U7 is optional, it may be 2516, 2716, 2532, 2732, 6116
2. Jumper 1 is used for selecting certain memory type

Memory TyPe	Memory Address	Cuts	Jumpers
2516 2716	2000-27FF	NIL	NIL
2532	2000-2FFF	NIL	NIL
2732	2000-2FFF	PIN1 TO PIN2 PIN3 TO PIN4 PIN5 TO PIN6	PIN2 TO PIN3 PIN4 TO PIN5 PIN6 TO PIN7
6116	2000-27FF	PIN3TOPIN4	PIN4TO PIN5

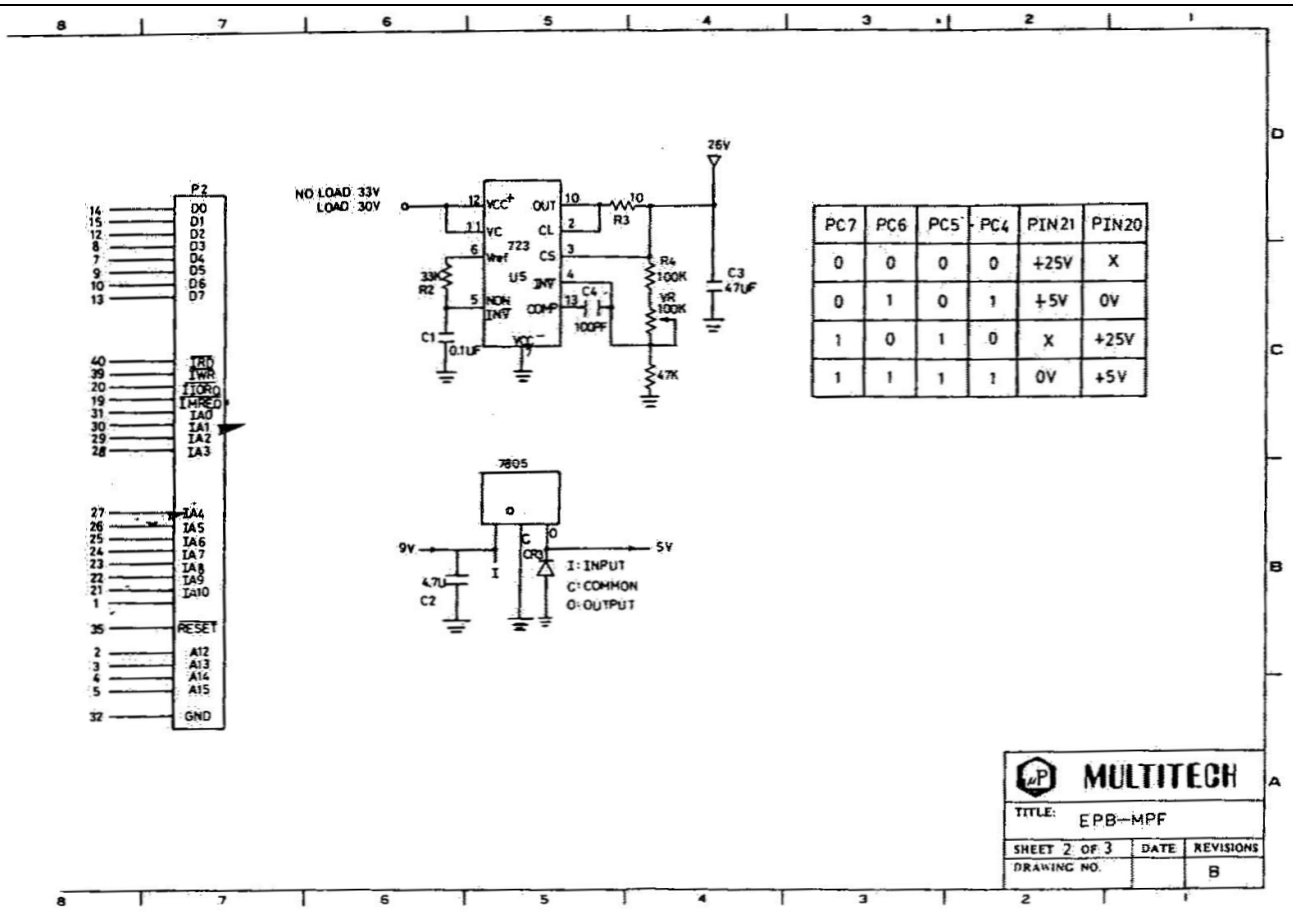
MULTITECH
TITLE: PF—I
SHEET 4 OF 4 | DATE | REVISIONS
DRAWING NO. | 8/09/22 | A

EPB-MPF Schematics

Board Layout

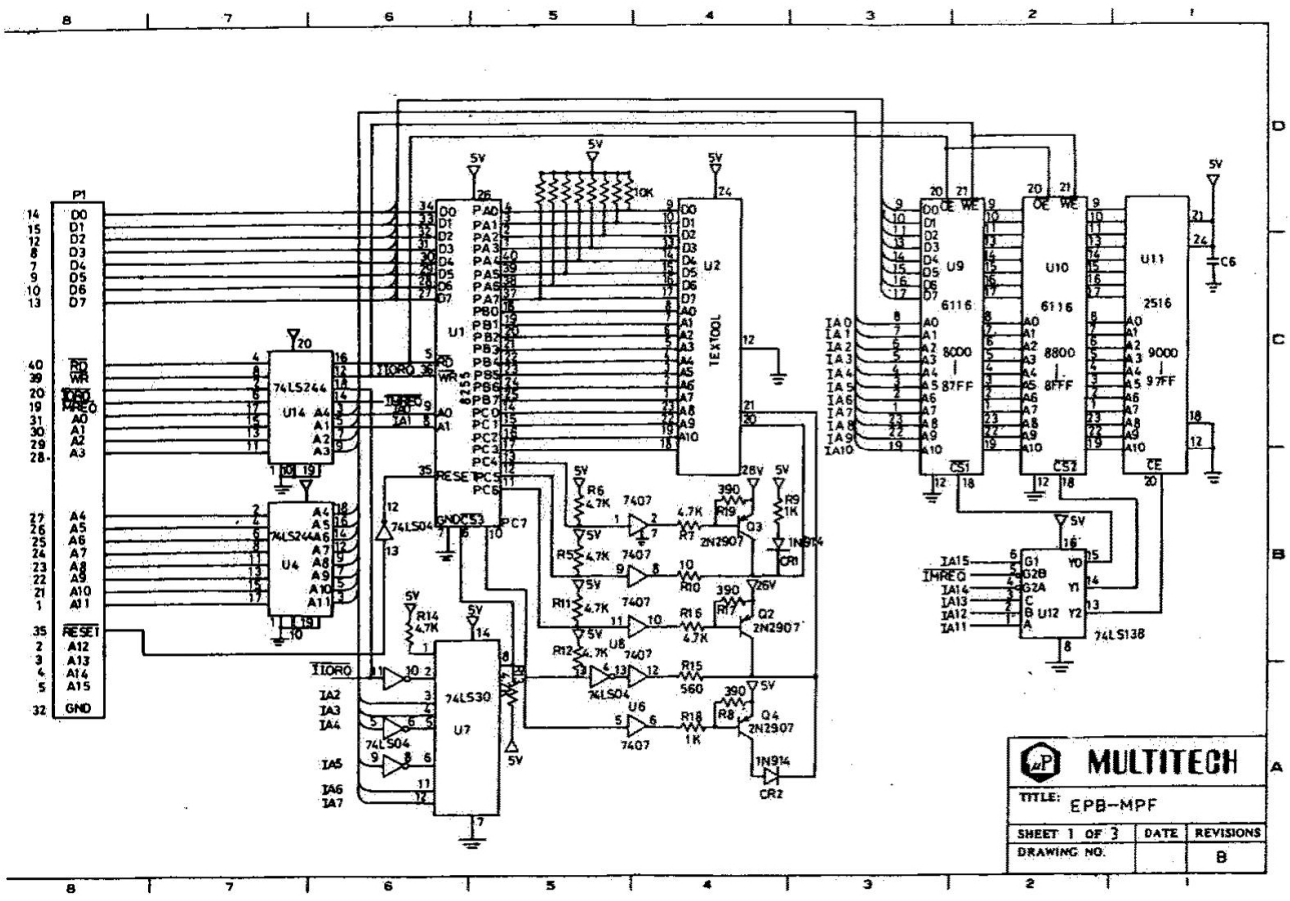


Power Supply and EPROM Control Signals



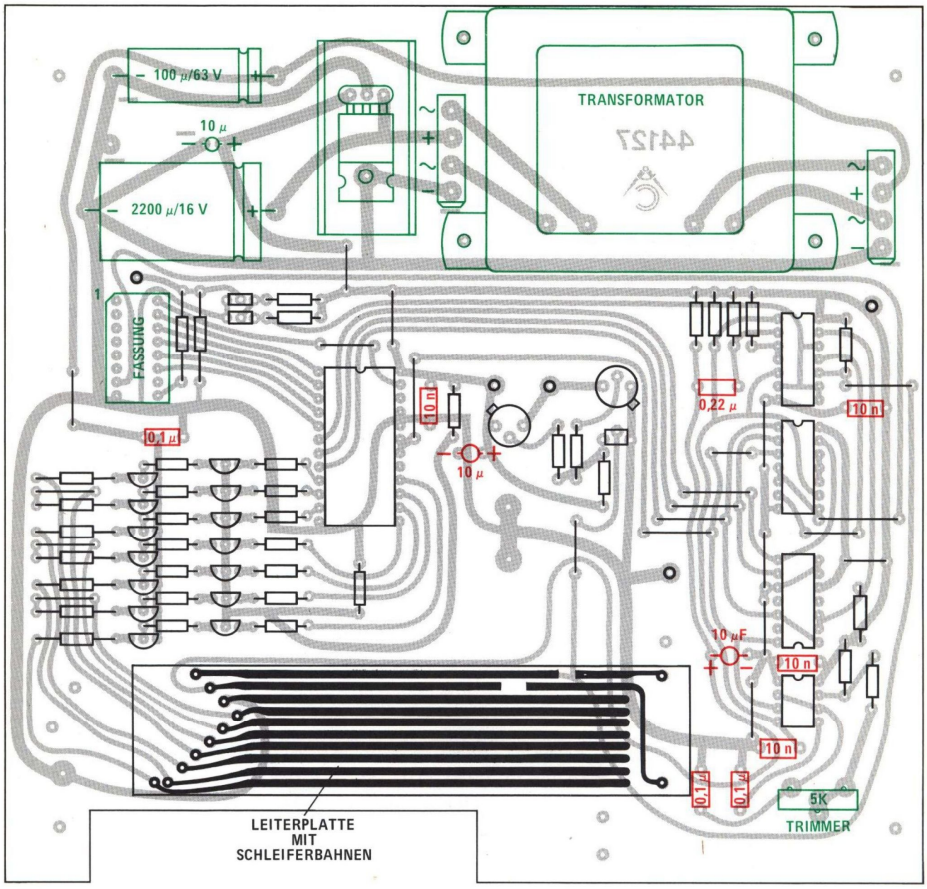
PC7	PC6	PC5	PC4	PIN21	PIN20
0	0	0	0	+25V	X
0	1	0	1	+5V	0V
1	0	1	0	X	+25V
1	1	1	1	0V	+5V

8255 PPI and ZIF Socket

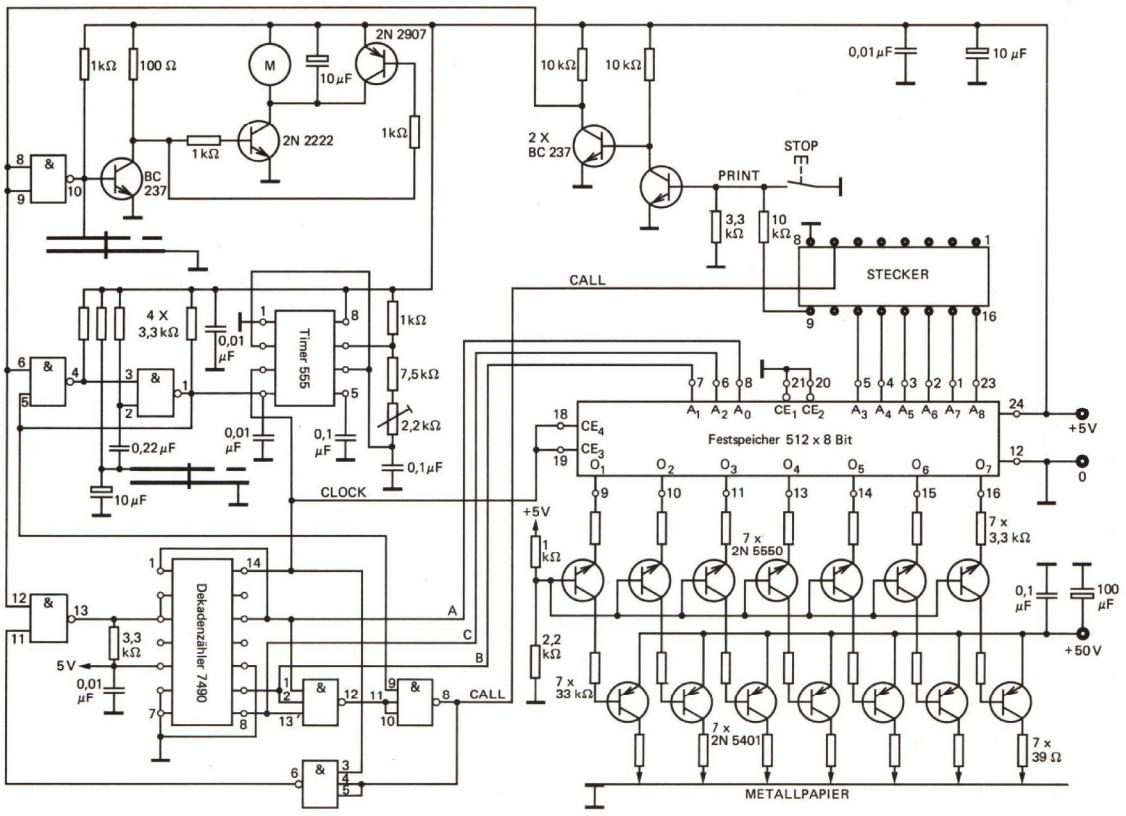


Christiani Printer and Cables

Board Layout



Schematics



PPI and PIO Cables