Multipurpose Interface Board 238 (MIB238)

Contents

List of Figures

Multipurpose Interface Board 238 (MIB238)

Installation and Operating Instructions

The MIB238 board is a redesign of the Micro Innovations MIB3 board that only requires 5 volts using modern MAX238 transceivers. The MIB238, like the MIB3, adds a BOOT PROM socket, two RS-232 ports, a parallel port, and a memory expansion board addressor port to your ADAM computer.

This document contains all the information you will need to get the MIB238 up and running. It's as simple as plugging it in and booting the supplied software. The detailed instructions contained herein will take you through the installation and checkout process.

We know you are anxious to get started. But first please read all the way through the instructions so you'll be somewhat familiar with the process before you actually start the installation.

A WORD OF WARNING ABOUT STATIC ELECTRICITY!

Before you get started, just a word of warning about static electricity. The integrated circuit chips on the MIB238 can be destroyed by static charges. If you notice that you get shocks when you touch metal appliances after walking around the room, then you should take precautions to prevent static discharges when handling it. There are a couple of common precautions you can take if you suspect static electricity is a problem in your installation environment.

One precaution you can take is to discharge yourself each time before you touch the MIB238. You can do this by performing your installation near an appliance you can touch to discharge the static electricity just prior to handling it. Another way is to connect a wire to a water pipe or the metal frame of a grounded appliance (like a refrigerator). AC power outlets in modern homes and businesses also can be a grounded source (you can pick up the ground from the screw that holds the cover plate onto the receptacle). Wrap the other end of this wire around your wrist or a finger. Make sure you are using the frame of a grounded appliance.

By the way, the MIB238 chips are not particularly sensitive to static electricity, but like all normal 74LS series integrated circuits, they can be destroyed if hit with a big enough discharge.

INSTALLATION

As mentioned earlier, the MIB238 has two RS-232 ports, a parallel port, and a memory expander board addressor port on it. Which cables you'll install will depend on how many MIB238 compatible devices you have. The following steps will take you through the installation process for all the MIB238 cables. If you do not have a particular device to connect to the MIB238 please disregard the installation instructions for that cable.

Parallel Printer Cable

Let's start with the parallel printer cable. This cable has a 26-pin socket connector on one end and a 36-pin Centronics connector on the other. The socket connector plugs in to MIB238 connector J5. Use the MIB238 layout drawing included as Figure 1 to locate its position. It is recommended you key the connector so that it plugs in to J5 in only one orientation (See Appendix B Parallel Cable Construction). The board connector J5 is labelled "Printer". Gently push the socket onto J5. For now, leave the other end unconnected. If you have too many pieces of equipment connected at first, you may have difficulty making the cables reach as you are installing the MIB238 into your computer.

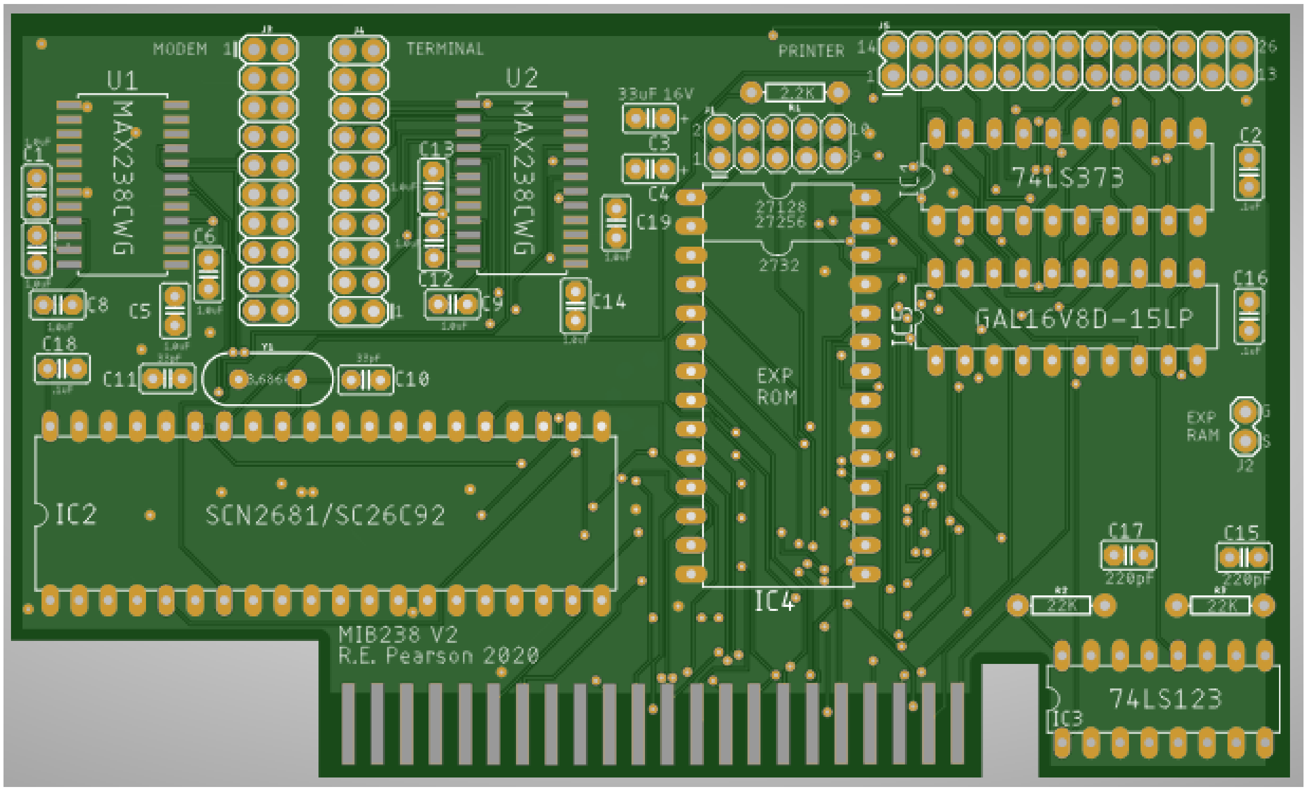


Figure 1 MIB238 Layout

Serial Port Cables

Now for the Serial ports. Locate the cable(s) with a 20-pin socket connector on one end and a DB-25 male connector on the other. Now locate the serial connectors J3 and J4 on the circuit board. Connector J3 is for an external modem and J4 is for an 80 column CRT terminal. Align the 20-pin socket connector with the pins on J3 or J4 and push it on to the header in one orientation. As before, leave the other ends of both cables unconnected for now. Again, it is recommended you use keyed cables so you can only plug the cable in to the connector in one orientation (See Appendix A Serial Cable Construction).

Boot Prom

If you have a BOOT PROM for the MIB238 you should install it now. The PROM goes into the socket IC4 with its notch facing the top of the board and its other end flush with the bottom end of the socket. An IDE BOOT PROM should have two empty socket positions at the top on each side of the socket when installed correctly. Now place shorting jumpers on the 10-pin connector (labelled J1) according to the type of PROM installed. The following table will help you make those choices:

	J1 Jumpers
1-2	Disables Printer port
3-4	2764 or 27128 PROM
5-6	27256 PROM
7-8	2732 PROM (use this selection for a 4K IDE BOOT PROM)
9-10	27128 or 27256 PROM

Okay, we are ready to plug the MIB238 into the computer! Pick up the MIB238 with its dangling cables and take it to the computer. Note that the board is keyed to ONLY fit into slot #2, the center slot, and ONLY with the components facing towards the right side of the computer. (However, it is possible to insert the board backwards if you have the upper case removed or have cut out your plastic keying barriers.) Note that when the board is installed properly, the cables must be wrapped back over the top of the board to have them exit the computer on the left side.

If you have a board in the center slot, it will have to be removed. If you do have a board in slot #2, the center slot, it is most likely a parallel printer interface or memory addressor board. The MIB238 will replace all the functions of the board you have to remove, so don't worry that you will lose your printer port or memory expansion board addressing capability. You can take the removed board and sell it to someone who does not have a MIB238!

Expansion RAM Addressor

If you have a memory expander board installed in slot #3 (the right slot), you'll have one or two wires running from it to the board you are removing. The MIB238 provides the same signal (and its return wire – a ground connection) to your memory expander board, so detach the wire(s) at the printer board end.

Locate connector J2 (labelled 'EXP RAM') on the MIB238 (see Figure 1 to locate it). It consists of two pins – pin 'S' is for the bank switch signal to the memory board and pin 'G' is for a ground line. If you experience erratic operation of your memory board, you'll have to use both pins and twisted wire between the boards. Twisted wire will reduce the amount of noise that the signal wire will pick up. If you're using a single wire from the memory expansion board, connect it to J2 pin 'S'. The most common color codes for the wires are red or white for the signal wire and black for the ground wire.

Connect the memory board wire (if needed) to the MIB238 and insert it into the center expansion slot. Make sure that the MIB238 sits straight up in the center slot (the cables can pull the MIB238 so adjust their tension so that the board sits up straight in the slot). The top cover can be put back in place but won't close all the way unless you cut a narrow slot to allow the cables to exit.

POWERING UP

Turn on your ADAM with no disks or tapes in your drives. SmartWriter should launch as usual. If it does, skip the next paragraph. If you have a hard disk BOOT PROM installed, you'll see the Micro Innovations splash screen, followed by the SmartWriter screen (with the hard disk interface board out of the ADAM).

If SmartWriter does not start up, turn of the power and remove the MIB238. Try it again with the MIB238 removed. If SmartWriter comes up fine without the MIB238 installed, it is likely that your MIB238 is defective. Incorrect construction of the MIB238 or static electricity are two possibilities.

BOOTING UP

Insert the distribution diskette or tape into the appropriate disk or tape drive. Pull the computer reset switch. TDOS 4.5 will boot up and sign on. You can what programs are provided on the distribution disk or tape by typing "DIR" and hitting the RETURN key. The exact collection of files on the distribution media may var but should include at least the following:

40MIB458.COM	40 column TDOS Install program

80MIB458.COM	80 column TDOS Install program
CLONE21.COM	The utility program used to copy an IMG file from a TDOS media to an EOS media
DRIVES12.COM	Used to identify all disk and tape drives attached to your ADAM
IOBYTE12.COM	Utility to set the TDOS IOBYTE
BASPATCH.IMG	Utility to patch the MIB238 parallel port driver into EOS and boot SmartBasic
PARPATCH.BAS	Utility to patch the MIB238 paralley port driver into EOS
IMP-MIB3.COM	The IMP communication program patched for the MIB238
MEX-MIB3.COM	The MEX communication program patched for the MIB238
UNCR.COM	File un-Crunch utility to expand compressed MIB238 documentation files to normal text
MEX.HZP	Crunched MEX help file
IMP.DZC	Crunched IMP documentation file

INSTALLING AN OPERATING SYSTEM

The MIB238 distribution tape or disk comes with the TDOS operating system installed. To boot it, you need only to hit the RESET switch with the distribution media installed. If you wish to reconfigure your system, you must select the version you want to install (the 40 column for the ADAM screen or the 80 column for an external terminal) and execute the appropriate TDOS install program by typing its file name (without the extension) – 40MIB458 Or 80MIB458. The install program will prompt you for the information about your system, configure TDOS for you, and then install it on a boot diskette or tape. Hitting the RESET switch will then cause TDOS to be booted.

INSTALLING TDOS

If you want to reinstall TDOS, you must first choose which version you need. Choose 80MIB458 If you have an external terminal or an 80-column add-on unit, or 40MIB458 To use the ADAM monitor. Execute the appropriate version by typing its name and hit the RETURN key. For example, if the names of the TDOS install programs provided on the distribution media are 40MIB45L.COM and 80MIB45L.COM, then type its full name, 40MIB45L or 80MIB45L, followed by the RETURN key to execute the appropriate version.

When the TDOS install program signs on, the TDOS release number will be shown on the top line. The first screen asks you specify which ADAM disk or tape drive to write the operating system to. It checks immediately after your selection to see if the device exists on your system. If it doesn't, it gives you an error message and lets you try again. You can get out of the installation program at any prompt by typing a CONTROL-C (that is holding down the CONTROL key and hitting the 'C' key).

The following tells you what your TDOS drive assignments are. TDOS assigns the drive letters to all the devices it finds when it is installed. It always assigns the drive letters starting with the fastest first (you may choose to have your RAM disk first or last, however). For example, if you have a single disk drive and two tape drives (no memory expander), the disk drive will be drive A and the tape drives will be B and C.

The next screens ask you to specify the size of the floppy disk drives – one screen for each drive. The choices are:

1 – 145K Standard Coleco single-sided 40 track format
2 – 254K Medium sized double-sided 40 track format
3 – 304K Full-sized double-sided 40 track format
4 – 356K IBM-sized double-sided 40 track format
5 – 702K Quad density 80 track format
6 – 714K Quad density 80 track format
7 – 1418K High Density 80 track format

Formats 1 through 4 are used for 5 ¼'' floppy diskettes and formats 5, 6, and 7 for 3.5'' diskettes. The exact selection of formats available for your system will depend on the equipment you have. Not all of the alternatives are valid – for example, you can't have a 714K format on a 5 ¼'' floppy disk drive. Some formats may require a special EPROM in your ADAM floppy drive for it to be functional. All formats except the 356K, 714K, and 1.4MB formats are compatible with existing ADAM formats.

The DSKSZ??? program will let you temporarily change your ADAM floppy disk drive definitions so that you can keep your permanent format different than one you might use only for information interchange. To permanently change to a different format, you must re-install TDOS.

After selecting floppy disk formats, the next two screens ask if you'd like to change parameters on the MIB3 serial ports. Serial Port 1 (J3) is wired for direct connection for an external modem and is setup for a default of 2400 baud, no parity, 8 data bits, and one stop bit. You can exit the screen without changing any of the parameters (by hitting a '0'), or you can choose to change any of the parameters if you desire (a '1' to change baud rate, a '2' to change parity, a '3' to change the number of data bits, or a '4' to change number of stop bits). The default settings are normal for a 2440 baud external modem. After exiting the screen, you are asked the same questions about Serial Port 2 (J4), which is wired for direct connection of an external CRT terminal or a serial interface printer. Its defaults are 19200 baud, no parity, 8 data bits, and 1 stop bit – normally the highest speed an external terminal can run.

You are next asked if you would like to change the IOBYTE assignments. CP/M and TDOS use the IOBYTE to know which physical devices to use for each of their five logical devices. The five logical devices are CON: (the system console output), KEY: (the system console keyboard input), RDR: (the reader), PUN: (the punch), and LST: (the system printer).

The reader and punch device names are left over from the days when a paper tape reader or a paper tape punch were common microcomputer peripheral devices. Each of the logical devices can be assigned to any of it's five physical devices, and the physical devices to be selected from can be different from logical device to logical device. The valid assignments for logical devices are shown in the table below:

Logical Device	Permitted Physical Device Assignments			
CON:	CRT:	SR1:	SR2:	UC1:
KEY:	KYB:	SR1:	SR2:	UK1:
RDR:		SR1:	SR2:	
PUN:		SR1:	SR2:	
LST:	LP1:	SR1:	SR2:	PAR:

Definitions for the physical devices are as follows:

For Logical Device CON:, the system console
CRT: ADAM 40 column display
SR1: MIB238 Serial Port #1 Out

SR2:	MIB238 Serial Port #2 Out
UC1:	80 column terminal output

NOTE – on the 80-column version of TDOS, physical device CRT: is the ADAM Serial Port.

For logical device KEY:, the keyboard

KYB:	ADAM keyboard
SR1:	MIB238 Serial Port #1 input
SR2:	MIB238 Serial Port #2 input
UC1:	80 column terminal input

For logical device RDR:, the reader

SR1:	MIB238 Serial Port #1 input
SR2:	MIB238 Serial Port #2 input

For logical device PUN:, the punch

SR1:	MIB238 Serial Port #1 output
SR2:	MIB238 Serial Port #2 output

For logical device LST:, the printer

LPT:	The ADAM printer
SR1:	MIB238 Serial Port #1 output
SR2:	MIB238 Serial Port #2 output
UL1:	MIB238 Parallel Printer Port

Note that is possible during the installation process to define where you want your printer output to go and what device you want to use for the system console. The reader and punch logical devices are not used by many programs. About the only way we know of is the PIP (Peripheral Interchange Program) program supplied with CP/M. You can use PIP to copy files in and out the assigned physical devices (for example – between computers) but no error checking protocol is used. You will be much better of to use one of the modem programs. Two are provided on the distribution media already patched for the MIB238 serial ports. All of the modem programs are designed to talk directly to the physical devices and purposely bypass the reader and punch logical devices.

The default IOBYTE assignments are:

CON:	CRT: (the ADAM display)
KEY:	KBD: (the ADAM keyboard)
RDR:	SR1: (MIB238 serial port #1 in)
PUN:	SR2: (MIB238 serial port #1 out)
LST:	UL1: (MIB238 parallel printer port)

NOTE – The default system console (CON:) for the 80-column version is the ADAM Serial Port.

After you've selected your IOBYTE assignments or chosen not to change them, the installation program asks if you'd like to change the function key definitions. This is a rather long and technical operation so if you're even marginally satisfied with the function key translations, avoid this part of the process. By the way, the default function key definitions match normal Wordstar definitions.

The next screen asks if you would like to change the SMART key strings. These are the character strings that are sent to the operating system whenever you hit a SMART key. The default settings are:

I	COPY
II	REN (to rename a file)
III	DEL (to delete a file)
IV	LIST (to print a file)
V	TYPE (to display a file on the console
VI	DIR (to display a directory listing on the console)

The last screen asks you to insert a tape or disk for the boot block to be written on. After you hit the RETURN key, the installation program writes the operating system to the diskette or tape. TDOS installation is now complete.

NOTE: The 80-column version asks three additional questions before it prompts you to insert a tape or disk. It asks you if you are using an ADAM keyboard for the console, if you want the SMART key definitions displayed on line 25 of your 80 column display (the display must have a command set compatible with the Heathkit H-19 or Zenith Z-19 terminal, which is what the ADAM uses) and whether or not ADAM Serial Port 2 is configured for an EVE 80 column display.

INSTALLING EOS PATCHES

Two patch programs (PARPATCH.IMG and BASPATCH.IMG) are provided with the MIB238 to allow its parallel printer port to operate with EOS. To utilize either of them, you must copy them to tape or disk using the CLONE21.COM program. The CLONE program is executed by typing:

```
          “CLONE21 PROGNAME.IMG X:”
```

and hitting the RETURN key. The X: portion of the command line is the letter of the floppy or tape drive you are copying to.

NOTE: TDOS assigns the drive letters in order to all storage devices it finds when it is installed. It always assigns the drive letters starting with the fastest drives first (you may choose to have your RAM disk first or last, however). If you have a single Disk drive and two tape drives (no memory expander), the disk drive will be drive A and the tape drives will be B and C.

Let’s run through an example of “clone”ing with the BASPATCH.IMG program. The program is supplied on the MIB238 Distribution Diskette and therefore resides on a TDOS media. To be able to use it, we must clone it to a newly formatted EOS media (disk or tape). We’ll assume that you have one disk drive. Therefore, you’ll have to clone it to a tape. Knowing that you have at least one tape drive, we’ll assume that you have you’re newly formatted tape in tape drive #1. Since Disk #1 is drive A, and you don’t have a second disk drive, Tape #1 will be drive B. The command you’ll enter to clone the program is:

```
CLONE21 BASPATCH.IMG B:
          <RETURN>
```

To utilize the BASPATCH program, you must copy SmartBasic onto the EOS media that BASPATCH was cloned to (using a file copy program, such as AJM Software’s File Manager), and pull the RESET switch. The BASPATCH program will boot and patch EOS for the parallel printer port on the MIB238. It will then load and execute SmartBasic. You can now print onto a dot matrix printer attached to the MIB238 parallel printer port.

The process is the same to clone the PARPATCH program.

APPENDIX A – Serial Port

MIB238 Serial Port Information

This section describes the register addresses and pin assignments for the MIB238's Serial ports, connectors J3 and J4. Both serial ports are provided by a single integrated circuit, a Signetics 2681, which the manufacturer calls a DUART (DUal Asynchronous Receiver/Transmitter). The combined driver/receiver chips used are Maxim MAX238s. All of the other currently available ADAM serial port products utilize the Sygnetics 2651, which is a single serial port IC. The register addresses and bit assignments within the registers are different between the 2651 and 2681. Therefore, software written for the 2651 will not function with 2681 without modification. Provided are patched version of IMP and MEX communications programs so the user will not have to modify those programs. In addition, ADAMlink 5 and other communication programs will work with the MIB238.

However, for those users who wish to utilize some other communication package or would like to talk directly to the ports from programs they have written, the I/O address information is given below. All addresses are in hexadecimal. Bit assignments within registers are in accordance with the 1983 Signetics MOS Microprocessor Data Manual.

```
;SIGNETICS 2681 DUART I/O PORT EQUATES
;NOTE: Port A is Serial Port 2, Port B is Serial Port 1
;
S2681      EQU  10H        ;S2681 DUART BASE ADDRESS
MRA        EQU  S2681      ;MODE REGISTERS 1A AND 2A
SRA        EQU  S2681+1    ;STATUS REGISTER A
CSRA       EQU  S2681+1    ;CLOCK SELECT REGISTER A
CRA        EQU  S2681+2    ;COMMAND REGISTER A
RHRA       EQU  S2681+3    ;RX HOLDING REGISTER A (RX  DATA)
THRA       EQU  S2681+3    ;TX HOLDING REGISTER A (TX DATA)
IPCR       EQU  S2681+4    ;INPUT PORT CHANGE REGISTER
ACR        EQU  S2681+4    ;AUXILLIARY CONTROL REGISTER
ISR        EQU  S2681+5    ;INTERRUPT STATUS REGISTER
IMR        EQU  S2681+5    ;INTERRUPT MASK REGISTER
CTU        EQU  S2681+6    ;COUNTER/TIMER UPPER
CTUR       EQU  S2681+6    ;COUNTER/TIMER UPPER REGISTER
CTL        EQU  S2681+7    ;COUNTER/TIMER LOWER
CTLR       EQU  S2681+7    ;COUNTER/TIMER LOWER REGISTER
MRB        EQU  S2681+8    ;MODE REGISTERS 1B AND 2B
SRB        EQU  S2681+9    ;STATUS REGISTER B
CSRB       EQU  S2681+9    ;CLOCK SELECT REGISTER B
CRB        EQU  S2681+10   ;COMMAND REGISTER B
RHRB       EQU  S2681+11   ;RX HOLDING REGISTER B (RX DATA)
THRB       EQU  S2681+11   ;TX HOLDING REGISTER B (TX DATA)
IP         EQU  S2681+13   ;INPUT PORT
OPCR       EQU  S2681+13   ;OUTPUT PORT CONFIGURATION REGISTER
STARTC     EQU  S2681+14   ;START COUNTER COMMAND PORT (READ)
SOPB       EQU  S2681+14   ;SET OUTPUT PORT BITS COMMAND PORT
STOPC      EQU  S2681+15   ;STOP COUNTER COMMAND PORT (READ)
ROPB       EQU  S2681+15   ;RESET OUTPUT PORT BITS CMD PORT
```

Detailed Coleco ADAM Computer MIB238 I/O Address Map

Port #	Device	Input	Output	Comments
1	MIB238 RESET line	* Not Used on MIB238 *	Bit 3 = 1 for MIB238 RESET	
10	MIB238 Serial ports	Mode Register A	Mode Register A	1st write is to MR1A, second write is to MR2A,requires reset to go back to MR1A
11	MIB238 Serial ports	Status Register A	Clock Select Reg A	BAUD RATE SELECT
12	MIB238 Serial ports	* DO NOT USE *	Command Register A	
13	MIB238 Serial ports	RX Holding Register A	TX Holding Reg A	
14	MIB238 Serial ports	Input Port Change Reg	Aux Control Register	
15	MIB238 Serial ports	Interrupt Status Reg	Interrupt Mask Reg	
16	MIB238 Serial ports	Read Counter Upper	Set C/T Upper Register	
17	MIB238 Serial ports	Read Counter Lower	Set C/T Lower Register	
18	MIB238 Serial ports	Mode Register B	Mode Register B	
19	MIB238 Serial ports	Status Register B	Clock Select Reg B	
1A	MIB238 Serial ports	* DO NOT USE *	Command Register B	
1B	MIB238 Serial ports	RX Holding Register B	TX Holding Register B	
1C	MIB238 Serial ports	* Reserved (note 1) *	MIB238 Serial Port RESET	
1D	MIB238 Serial ports	Read Input Port Bits	Output Port Config Reg	
1E	MIB238 Serial ports	Start Counter Cmd Port	Set Output Port Bits	
1F	MIB238 Serial ports	Stop Counter Cmd Port	Reset Output Port Bits	

1) Reserved ports in serial port map: Input ports 12 and 1A - screw up serial ports if used; Input port 1C doesn't bother anything but the 2681 drives the bus.

Handshaking Lines

The handshaking lines use the S2681’s general purpose input and output ports and are assigned as follows:

J3 – Serial Port 1 (wired for direct connection of a modem) –

Signal Line Name	Bit Number	Interface Board Pin	RS-232 Pin
Ground		J3 Pin 1	1
TX		J3 Pin 3	2
RX		J3 Pin 5	3
Request To Send	Output Bit 1	J3 Pin 7	4
Clear To Send	Input Bit 1	J3 Pin 9	5
Data Set Ready	Input Bit 3	J3 Pin 11	6
Data Terminal Ready	Output Bit 3	J3 Pin 14	20
Carrier Detect	Input Bit 5	J3 Pin 15	8

J4 – Serial Port 2 (wired for direct connection of a terminal or a serial printer)

Signal Line Name	Bit Number	Interface Board Pin	RS-232 Pin
Ground		J4 Pin 1	1
RX		J4 Pin 3	2
TX		J4 Pin 5	3
Clear To Send	Input Bit 0	J4 Pin 7	4
Request To Send	Output Bit 0	J4 Pin 9	5
Data Terminal Ready	Output Bit 2	J4 Pin 11	6
Data Set Ready	Input Bit 2	J4 Pin 14	20
Carrier Detect	Output Bit 4	J4 Pin 15	8

Serial Cable Construction

Follow the illustration below to construct your MIB238 Serial Cables to connect to Serial Port 1 or 2. Select a length of ribbon cable with 20 conductors. Cable length should be 24 inches. On the RS-232 side, line the cable up with pin 1. The RS-232 connector will have 5 pins with no wire connection.

If you have a multimeter, I recommend checking continuity between Pin 1 on the 20-pin socket and Pin 1 on the 25-pin RS-232 connector.

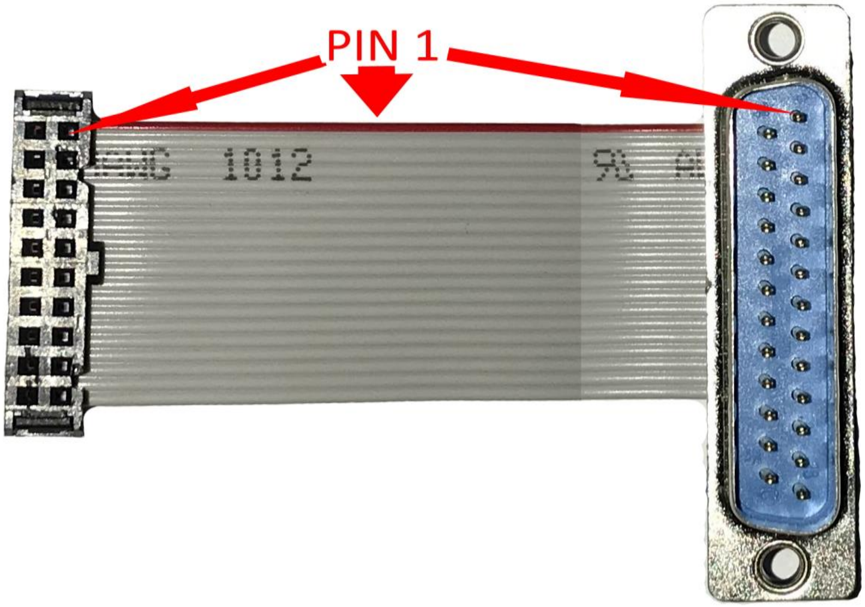


Figure 2 Serial Cable Diagram

It is recommended you "key" your Serial port cables and connectors so they can only connect in the correct orientation.

Instructions for Keying:

1. On connector J3 and J4 cut pin number 16 (or during construction of the MIB238 remove pin 16).
2. On the cable that will connect to connector J3 block the plug for pin number 16 with a small amount of epoxy or super glue. See Figure 3.
3. Repeat steps 2 and 3 for the cable that will plug in to connector J4.

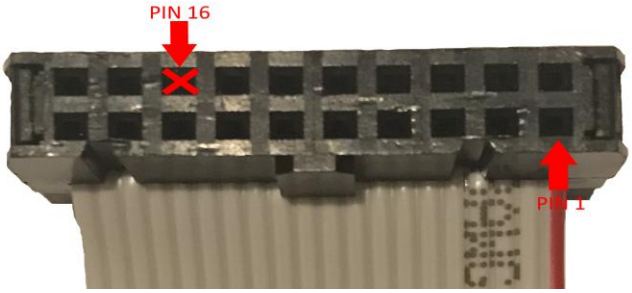


Figure 3 Serial Cable Keying

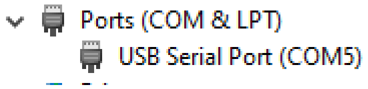
Using a PC as a Terminal Monitor

You can connect the MIB238 terminal port J4 to a Windows PC to display 80 column terminal output from programs like ADAMLink V or operating systems like T-DOS. To do this you will need the following:

1. A DB25 female to DB9 female Serial adapter
2. A USB to DB9 Serial port convertor
3. A Windows based terminal program, I recommend Putty. You can download Putty here: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

For the connection you would first plug the cable from the MIB238 terminal port into the DB25 to DB9 adapter. Then plug the DB25 to DB9 adapter into the USB to DB9 converter. Finally plug the USB to DB9 converter into the USB port on your Windows PC.

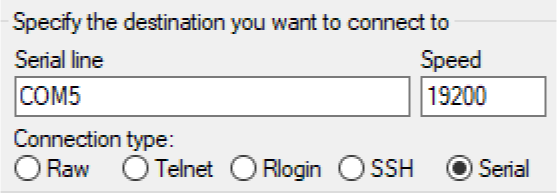
To determine which COMPORT the USB to DB9 converter is using on your PC launch Device Manager and check under 'Ports'. See Figure 4.



Ports (COM & LPT)
USB Serial Port (COM5)

Figure 4 Device Manager Ports

Launch Putty on your PC and create a new connection using the COMPORT for the USB to DB9 converter found in Device Manager, use a speed of 19200 and the connection type should be Serial. See Figure 5.



Specify the destination you want to connect to

Serial line: COM5
Speed: 19200

Connection type:
○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Figure 5 Putty Connection Configuration

Click 'Open' in Putty to launch the terminal window.

Then just launch the appropriate application on the ADAM computer and ensure the terminal output is configured to use the MIB serial port number 2 and a speed of 19200.

Example Z80 Code

An example of practical Z80 code for the serial ports is given below:

```
;MIB2/3/238 MODEM LIBRARY
;
DATABIT:    DB     00
PARITY:     DB     00
STOPBIT:    DB     00
BAUDRATE:   DB     00
BAUDRATES:              ;0=300, 1=1200, 2=2400, 3=4800, 4=9600, 5=19200
      DB    44h         ;300
      DB    66h         ;1200
      DB    88h         ;2400
      DB    99h         ;4800
      DB    0BBh        ;9600
      DB    0CCh        ;19200
DATABITS:               ;0=5 bits per character, 1=6 bit, 2=7 bits, 3=8
bits
      DB    00h
      DB    01h
      DB    02h
      DB    03h
PARITYS:                ;1=odd parity,2=no parity,3=even parity
      DB    10h
      DB    04h
      DB    00h         ;3=even parity
STOPBITS:               ;0-2=1 stop bit
      DB    07h
      DB    07h
      DB    07h
;
;##################################################################
OFFSET:
; Entering - offset in A register
;            HL starting address to add offset to
; ld A register with value at HL+E offset
;##################################################################
      ld      e,a
      ld      d,00h
      add     hl,de
      ld      a,(hl)
      ret
;
;##################################################################
MIB238RESET:
;reset mib
;##################################################################
      ld      a,0ffh   ;set bit 3 1111 1111
      out     (01h),a  ;mib238 reset line
      ld      a,0f7h   ;unset bit 3 1111 0111
      out     (01h),a  ;mib238 reset line
      ret
;
;###################################################################
COMMANDREG:
;###################################################################
      out     (c),a
      inc     c
      ld      a,05h
      out     (c),a
      ld      a,0f0h
      out     (14h),a ;Aux Control Register
      ld      a,0ffh
      out     (1eh),a  ;Set Output Port Bits
      ret
;
;###################################################################
SENDBYTE:
; send a byte through register A
;###################################################################
      push    af
.LOOP:
      in      a,(19h) ;Status Register B
      ;in  a,(11h)    ;Status Register A
      bit     3,a     ;check TxEMT
      jr      z,.LOOP
      pop     af
      out     (1Bh),a ;TX Holding Reg B
      ;out     (13h),a;TX Holding Reg A
      Ret
;
;###################################################################
RECEIVEBYTE:
; receive a byte
;###################################################################
      ex      af,af'
      in      a,(19h) ;Status Register B
      ;in      a,(11h);Status Register A
      bit     0,a     ;check RxRDY
      jr      nz,.SKIP
      ex      af,af'
      ret
.SKIP:
      in      a,(1Bh) ;RX Holding Register B
      ;in      a,(13h);RX Holding Register A
      call    STORE   ;store received character in buffer
      ex      af,af'
      ret
;
;###################################################################
INITBUFFER:
;###################################################################
```

```
      push    hl
      ld      hl,0000h
      ld      (7a9eh),hl    ;new received character position
      ld      (7aa0h),hl    ;next character to pop position
      pop     hl
      ret
;
;##################################################################
STORE:
;store received character in buffer
;##################################################################
      push    hl
      push    bc
      push    af
      ld      bc,(7a9eh) ;get buffer end
      ld      hl,8902h
      add     hl,bc      ;add buffer end to buffer beginning
      ld      (hl),a     ;store character in buffer
      ld      hl,(7a9eh) ;get buffer end put in HL
      call    CHECKBUFFER
      ld      (7a9eh),hl ;save new buffer end
      ld      bc,(7aa0h)
      or      a
      sbc     hl,bc
      jr      nz,.SKIP
      ld      hl,(7aa0h)
      call    CHECKBUFFER
      ld      (7aa0h),hl
.SKIP:
      pop     af
      pop     bc
      pop     hl
      ret
;
;##################################################################
CHECKBUFFER:
;checks if buffer position has exceeded buffer and if has
;resets position
;##################################################################
      push    bc
      push    af
      inc     hl
      push    hl
      ld      bc,13ffh
      or      a
      sbc     hl,bc
      pop     hl
      jr      nz,.SKIP
      ld      hl,0000h;reset buffer end to 0
.SKIP:
      pop     af
      pop     bc
      ret
;
;##################################################################
CONFIGMIB238:
; sets serial data bits, parity, stop bits and baud rate
; set variables DATABIT, PARITY, STOPBIT, BAUDRATE before calling
;##################################################################
      ld      c,18h         ;set C to port $18 serial port 1, port $10
                            ;if serial port 2
      push    bc
      call    MIB238RESET   ;mib238 reset
      ld      hl,DATABITS
      ld      a,(DATABIT)   ;serial data bits
      call    OFFSET        ; ld A register with value at HL+E offset
      push    af
      ld      hl,PARITYS
      ld      a,(PARITY)    ;serial parity
      call    OFFSET        ; ld A register with value at HL+E offset
      pop     bc
      or      b
      pop     bc
      out     (c),a         ;set data bits and parity
      ld      hl,STOPBITS
      ld      a,(STOPBIT)   ;serial stop bits
      call    OFFSET        ; ld A register with value at HL+E offset
      out     (c),a
      inc     c       ;set C to port $11 clock select register A or $19
clock select register B
      ld      hl,BAUDRATES
      ld      a,(BAUDRATE)  ;serial baud
      call    OFFSET        ; ld A register with value at HL+E offset
      call    COMMANDREG    ; send to port $12 command register A or
$1A command register B and send $05 to port $13 or port $1B
      ld      a,0c3h
      ld      (0030h),a
      ld      hl,RECEIVEBYTE     ;set HL to receive character routine
      ld      (0031h),hl    ;set interrupt 31 to automatically receive
characters at be28h
      ld      a,0c3h
      ld      (2e68h),a
      ld      hl,0be38h
      ld      (2e69h),hl
      ;call    0b818h
      ret
;
;##################################################################
CHKRECBTIMES:
;checks B times if a character has been received
;##################################################################
.LOOP:
      ld      de,0b250h
.LOOP2:
      in      a,(19h)       ;Status Register B
      ;in      a,(11h)      ;Status Register A
      bit     0,a           ;check RxRDY
      jr      nz,GETBYTE
      in      a,(19h)       ;Status Register B
      ;in      a,(11h)      ;Status Register A
      bit     0,a           ;check RxRDY
      jr      nz,GETBYTE
      dec     e
      jr      nz,GETBYTE
      dec     d
      jr      nz,.LOOP2
      djnz    .LOOP
      pop     de
      scf
      ret
;
;##################################################################
GETBYTE:
;gets character from receive holding register
;##################################################################
      in      a,(1Bh)       ;RX holding register B
      ;in      a,(13h)      ;RX holding register A
      or      a
      pop     de
      ret
;
;##################################################################
CHKRECEIVE:
;checks if character has been received
;##################################################################
      in      a,(19h)       ;Status Register B
      ;in      a,(11h)      ;Status Register A
      bit     0,a           ;check RxRDY
      ret
;
;##################################################################
CHKGETBYTE:
;check receive status register and gets character if one has been
;received from receive holding register
;##################################################################
      call    CHKRECEIVE
      jr      z,CHKGETBYTE
      in      a,(1Bh)       ;RX Holding Register B
      ;in      a,(13h)      ;RX Holding Register A
      or      a
      ret
;
;##################################################################
```

```
CHKTRANSSEND:
;checks transmit register and wait for transmit to be clear to
;send, transmits character when clear
;A = byte to send
;##################################################################
      ld      b,01h
      push    de
      push    af
.RSTRT:
      ld      de,0b250h
.LOOP:
      in      a,(11h)       ;Status Register A
      bit     3,a           ; check TxEMT
      jr      nz,SENDIT     ;if value is not 0 then jump and transmit
                            ;character in transmit holding register
      in      a,(11h)       ;Status Register A
      bit     3,a           ; check TxEMT
      jr      nz,SENDIT     ;if value is not 0 then jump and transmit
                            ;character in transmit holding register
      dec     e
      jr      nz,.LOOP      ;if e not 0 then check again
      dec     d
      jr      nz,.LOOP      ;if d not 0 then check again
      djnz    .RSTRT
      pop     af
      pop     de
      scf
      ret
;
;##################################################################
SENDIT:
;send character, character to send in A on stack
;##################################################################
      pop     af
      out     (1Bh),a       ;TX Holding Reg B
      ;out     (13h),a      ;TX Holding Reg A
      or      a
      pop     de
      ret
```

APPENDIX B – Parallel Port

Parallel Printer Port Signals For Multipurpose Interface Board 238

Interface Board Signal Name	Interface Board Pin No.	Centronics Connector Pin No.	Centronics Printer Signal Name
Strobe	1	1	Strobe
Signal Ground	14	19	Strobe Return
D1	2	2	Data Bit 1
Signal Ground	15	20	Data 1 Return
D2	3	3	Data Bit 2
Signal Ground	16	21	Data 2 Return
D3	4	4	Data Bit 3
Signal Ground	17	22	Data 3 Return
D4	5	5	Data Bit 4
Signal Ground	18	23	Data 4 Return
D5	6	6	Data Bit 5
Signal Ground	19	24	Data 5 Return
D6	7	7	Data Bit 6
Signal Ground	20	25	Data 6 Return
D7	8	8	Data Bit 7
Signal Ground	21	26	Data 7 Return
D8	9	9	Data Bit 8
Signal Ground	22	27	Data 8 Return
No Connection	10	10	Acknowledge
Signal Ground	23	28	Ack Return
Printer Busy	11	11	Busy
Signal Ground	24	29	Busy Return
No Connection	12	12	Paper Error
Signal Ground	25	30	Reset Return
No Connection	13	13	Select
No Connection	26	31	Reset

Centronics Pinout

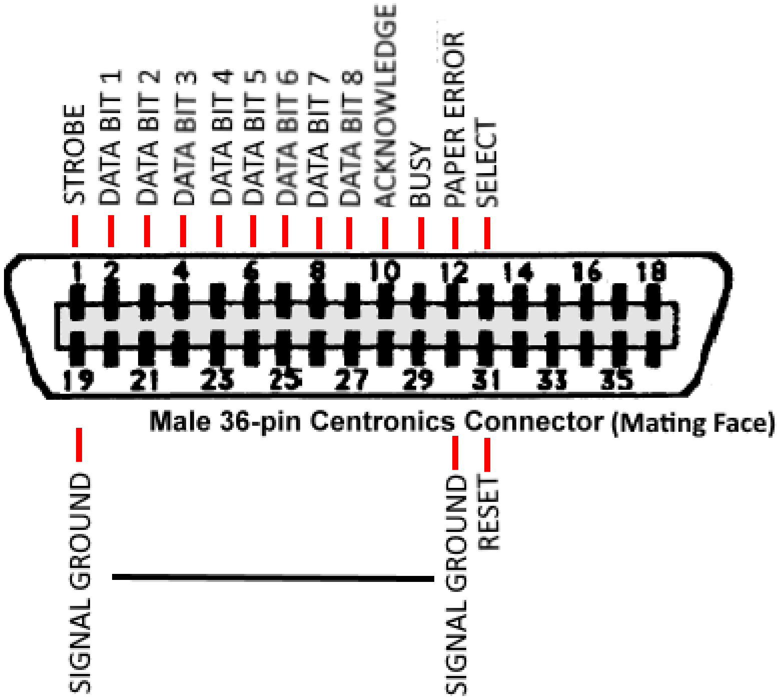


Figure 6 Centronics typical pinout

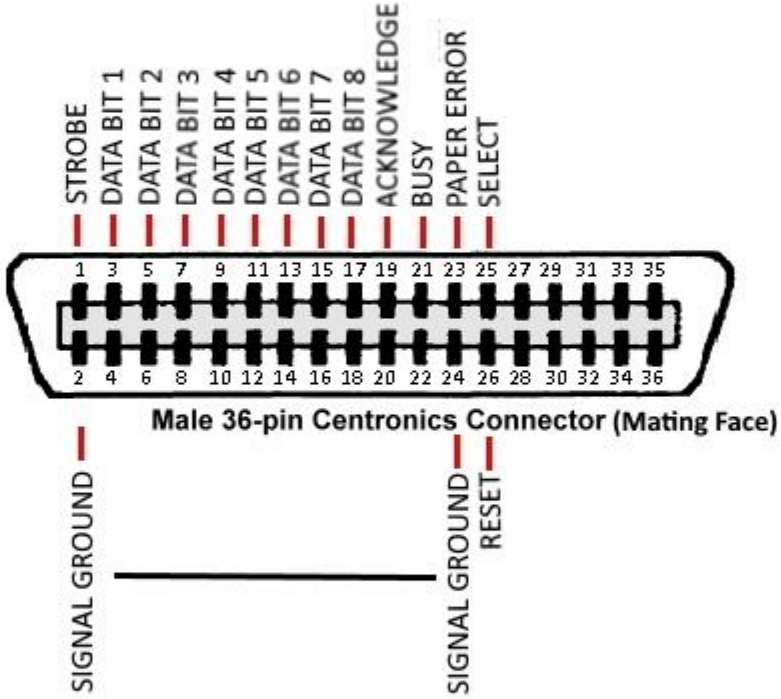


Figure 7 Centronics ribbon cable pinout

The parallel pinout numbering inf Figure 6 can be confusing since it does not illustrate correctly how the ribbon cable actually connects to the parallel connector (Centronics or DB25 parallel). This is shown in Figure 5.

Figure 7 shows the MIB238 parallel port connector ribbon cable pinout. You can see there are only 26 pins so no connection is made at all to the parallel port connector for pins 27 through 36.

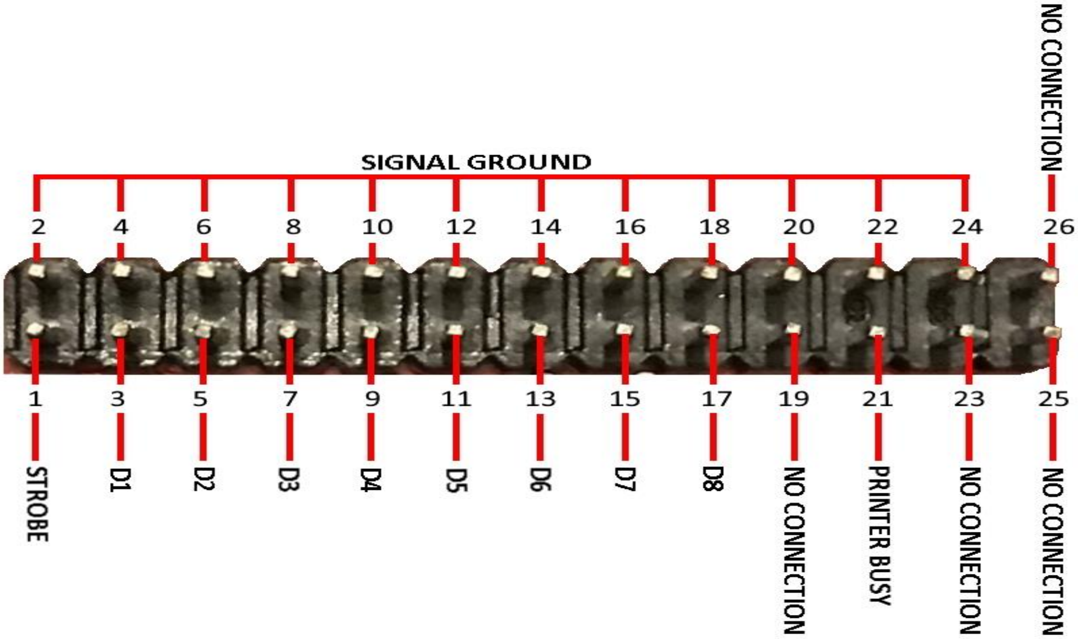


Figure 8 MIB238 Parallel Connector Pinout

Parallel Cable Construction

Follow the illustration below to construct your MIB238 Parallel Printer Cable to connect to the Parallel Printer Connector. Select a length of ribbon cable with 26 conductors. Cable length should be 24 inches. On the Centronics side, line the cable up with pin 1. The Centronics connector will have 10 pins with no wire connection.

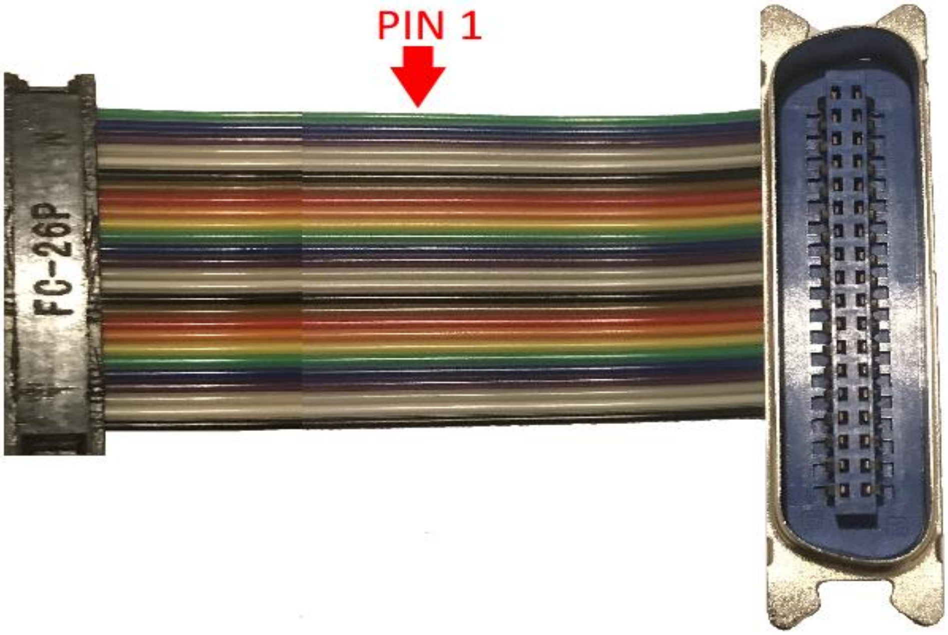


Figure 9 Parallel Cable Diagram

It is recommended you "key" your Parallel port cable and connector so it can only connect in the correct orientation.

Instructions for Keying (See diagrams):

1. On connector J5 cut pin number 26.
2. On the cable that will connect to connector J3 block the plug for pin number 26 with a small amount of epoxy or super glue.

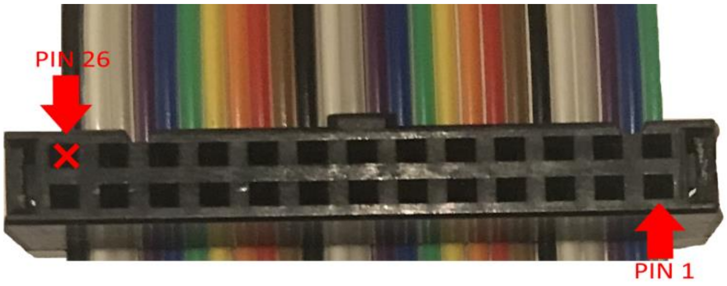


Figure 10 Parallel Cable Keying

Example Z80 Code

An example of practical Z80 code for the parallel port is given below:

```
PRINT
      in      a,(40h)       ;check printer status
      and     01h           ;is printer busy?
      jr      z,PRINT       ;yes, wait
      ld      a,(BUFFER)    ;get character to print from buffer
      out     (40h),a       ;send character to printer
```

APPENDIX C – Bill of Materials

Bill of Materials

PART	**QTY**
Board	1
MC2681	1
GAL16V8-15LP	1
MAX238CWG	2
Crystal 3.6864 MHz 20pf	1
SN74LS123N*	1
SN74LS373N	1
40 pin DIP socket (double wipe)	1
28 pin DIP socket (double wipe)	1
20 pin DIP socket (double wipe)	2
2x40 header connector	1
.1 uF ceramic radial capacitors	3
1 uF ceramic radial capacitors	10
32 pF ceramic capacitors (for crystal)	2
33 uF 16v electrolytic radial capacitors	2
220 pF ceramic disc capacitor	2
22k 1/4 watt resistors	2
2.2k 1/14 resistor	1
OPTIONAL	
2732 eprom	1
Cables	
Serial	
20-pin IDC female connector	2
DB25 IDC male crimp connector	2
Parallel	
26-pin IDC female connector	1
36 Pin Centronics male IDC Ribbon Cable Crimp Connector	1
Ribbon Cable (30-40 conductor) 2 meters	1

* I recommend NOT socketing the SN72LS123N due to clearance with the ADAM metal ground shield.

APPENDIX D - Construction

PCB Fabrication

You can send the Gerber file included with this package to the PCB fabrication house of your choice. Some PCB fabrication companies have slightly different naming conventions for Gerber files so you may need to adjust them. PCBWAY will accept the Gerber files as is. Below are basic parameters you may need to enter when completing your order. NOTE: Gold fingers are an option to improve the durability/longevity of the board but does add significant additional cost. If you decide to go with Gold fingers, I recommend the 'Immersed Gold' option.

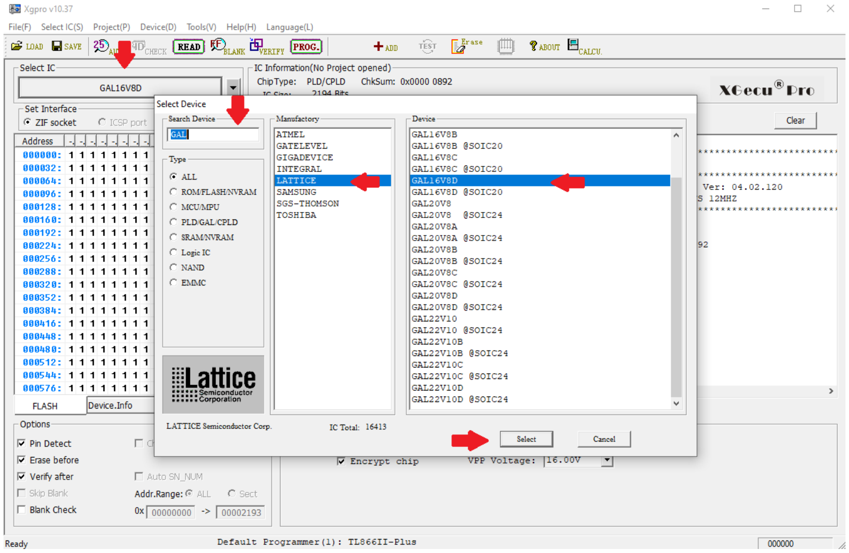
Parameter Information

Board type :	Single pieces
Size :	112 x 67 mm
Quantity :	5
Layer :	2 Layers
Material :	FR-4: TG130
Thickness :	1.6 mm
Min Track/Spacing :	6/6mil
Min Hole Size :	0.3mm
Solder Mask :	Red
Silkscreen :	White
Gold fingers :	No
Surface Finish :	HASL with lead
"HASL" to "ENIG"	No
Via Process :	Tenting vias
Finished Copper :	1 oz Cu

Programming the Lattice GAL18V8D-15LP

Example uses the MiniPro TL866II Plus

1. Connect the MiniPro programmer to your PC with the USB cable.
2. Launch the MiniPro programmer software.
3. Click the 'Select IC' button and in 'Search Device' type 'GAL'. The list of manufacturers will update with your search. Select 'LATTICE' in the 'Manufactory' field and the 'Devices' will update. Select 'GAL16V8D'. Then click the 'Select' button.



ChipType: PLD/CPLD ChkSum: 0x0000 0892

Search Device: GAL

Type: ● ALL

Manufactory	Device
ATMEL	GAL16V8B
GATELEVEL	GAL16V8B @SOIC20
GIGADEVICE	GAL16V8C
INTEGRAL	GAL16V8C @SOIC20
LATTICE	GAL16V8D
SAMSUNG	GAL16V8D @SOIC20
SGS-THOMSON	GAL20V8
TOSHIBA	GAL20V8 @SOIC24
	GAL20V8A
	GAL20V8A @SOIC24
	GAL20V8B
	GAL20V8B @SOIC24
	GAL20V8C
	GAL20V8C @SOIC24
	GAL20V8D
	GAL20V8D @SOIC24
	GAL22V10
	GAL22V10 @SOIC24
	GAL22V10B
	GAL22V10B @SOIC24
	GAL22V10C
	GAL22V10C @SOIC24
	GAL22V10D
	GAL22V10D @SOIC24

LATTICE Semiconductor Corp. IC Total: 16413

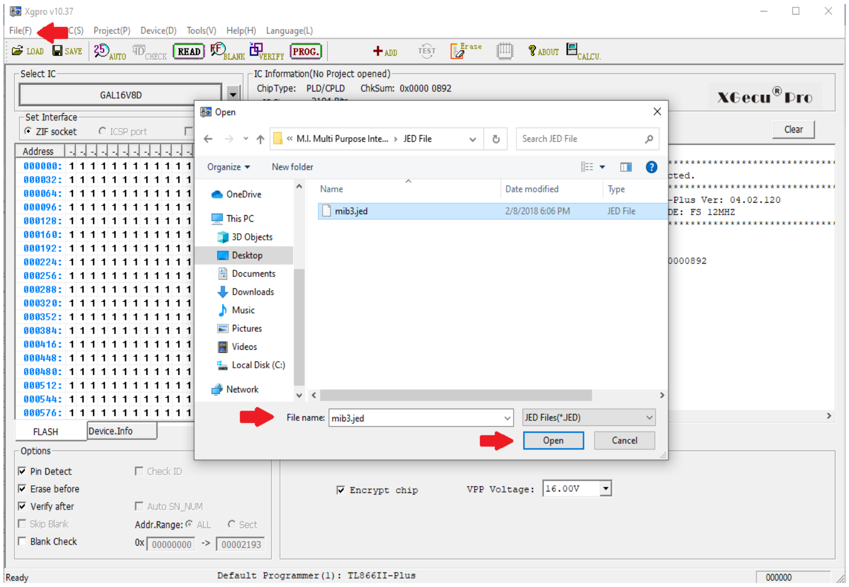
Ver: 04.02.120
S 12MHZ

☑ Pin Detect
☑ Erase before
☑ Verify after
☐ Auto SN_NUM
☐ Skip Blank
Addr.Range: ● ALL ○ Sect
☐ Blank Check
0x 00000000 -> 00002193

☑ Encrypt chip VPP Voltage: 16.00V

Default Programmer(1): TL866II-Plus 000000

4. Go to the 'File (F)' menu and select 'Open'.
5. From the open file dialog box select the 'mib3.jed' file that was included in the Distribution Media of this package.
6. Click the 'Open' button to the load the .JED file in.



ChipType: PLD/CPLD ChkSum: 0x0000 0892

Name	Date modified	Type
mib3.jed	2/8/2018 6:06 PM	JED File

File name: mib3.jed JED Files(*.JED)

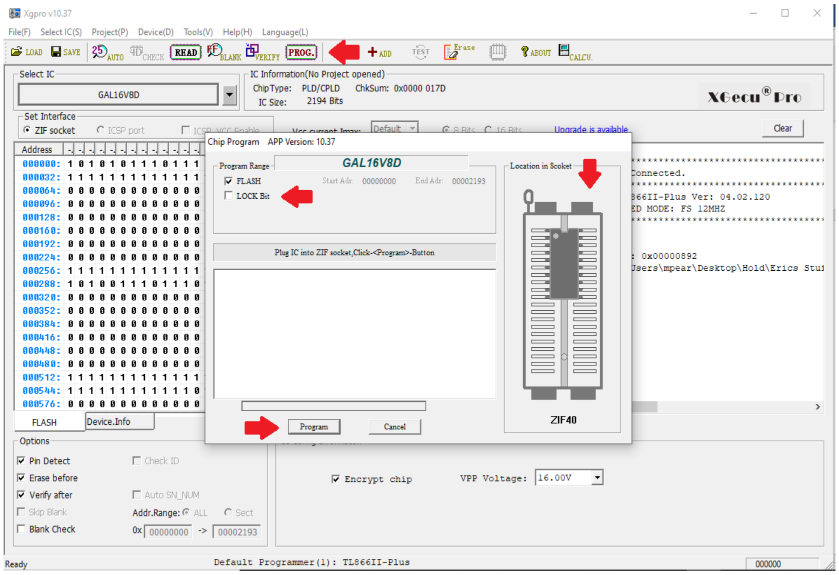
-Plus Ver: 04.02.120
DE: FS 12MHZ
0000892

☑ Pin Detect
☐ Check ID
☑ Erase before
☑ Verify after
☐ Auto SN_NUM
☐ Skip Blank
Addr.Range: ● ALL ○ Sect
☐ Blank Check
0x 00000000 -> 00002193

☑ Encrypt chip VPP Voltage: 16.00V

Default Programmer(1): TL866II-Plus 000000

7. Click the ‘PROG’ button for Program.
8. When the ‘Chip Program’ dialog box opens uncheck the ‘LOCK BIT’ checkbox.
9. Position your GAL16V8D-15LP chip in the chip programmer as indicated by the diagram and lower the locking arm.
10. Click the ‘Program’ button to program the chip.



Select IC: GAL16V8D

IC Information(No Project opened)
ChipType: PLD/CPLD ChkSum: 0x0000 017D
IC Size: 2194 Bits

Set Interface: ● ZIF socket ○ ICSP port □ ICSP_VCC Enable

Vcc current Imax: Default ● 8 Bits ○ 16 Bits

```
Address
000000: 1 0 1 0 1 0 1 1 1 0 1 1 1
000032: 1 1 1 1 1 1 1 1 1 1 1 1 1
000064: 0 0 0 0 0 0 0 0 0 0 0 0 0
000096: 0 0 0 0 0 0 0 0 0 0 0 0 0
000128: 0 0 0 0 0 0 0 0 0 0 0 0 0
000160: 0 0 0 0 0 0 0 0 0 0 0 0 0
000192: 0 0 0 0 0 0 0 0 0 0 0 0 0
000224: 0 0 0 0 0 0 0 0 0 0 0 0 0
000256: 1 1 1 1 1 1 1 1 1 1 1 1 1
000288: 1 0 1 0 0 1 1 1 0 1 1 1 0
000320: 0 0 0 0 0 0 0 0 0 0 0 0 0
000352: 0 0 0 0 0 0 0 0 0 0 0 0 0
000384: 0 0 0 0 0 0 0 0 0 0 0 0 0
000416: 0 0 0 0 0 0 0 0 0 0 0 0 0
000448: 0 0 0 0 0 0 0 0 0 0 0 0 0
000480: 0 0 0 0 0 0 0 0 0 0 0 0 0
000512: 1 1 1 1 1 1 1 1 1 1 1 1 1
000544: 1 1 1 1 1 1 1 1 1 1 1 1 0
000576: 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
*********************************
Connected.
*********************************
866II-Plus Ver: 04.02.120
ED MODE: FS 12MHZ
*********************************

: 0x00000892
Users\mpear\Desktop\Hold\Erics Stu
```

Chip Program APP Version: 10.37

GAL16V8D

Program Range
☑ FLASH Start Adr: 00000000 End Adr: 00002193
□ LOCK Bit

Location in Scoket

Plug IC into ZIF socket,Click-<Program>-Button

Options
☑ Pin Detect □ Check ID
☑ Erase before
☑ Verify after □ Auto SN_NUM
□ Skip Blank Addr.Range: ● ALL ○ Sect
□ Blank Check 0x 00000000 -> 00002193

☑ Encrypt chip VPP Voltage: 16.00V

Ready Default Programmer(1): TL866II-Plus 000000

APPENDIX E - MEDIA

Distribution Media

MIB238.jed – File required to program the GAL16V8-15LP chip.

MIB238 T-DOS Utilities.dsk – TDOS System disk that allows configuration and use of MIB238 ports.

SmartBASIC_MIB3_Patched.DSK – SmartBASIC 1.0 patched for use with the MIB238 parallel printer port.

ADAMLink V – Terminal program that supports all MIB238 ports, serial, terminal and parallel.

SNC2681 Datasheet.pdf – Datasheet for the 2681 Dual asynchronous receiver/transmitter (DUART). Contains detailed information on the SCN2681 pinout and registers including setting baud rate, stop bits, data bits, parity, etc.

IDEBOOTROM.BIN – 4K binary of the IDE BOOT ROM for 2732 EPROM.