MIDI interface™

USERS MANUAL

PASSPORT

"The Music Software Source"

-*-*-*-*- WARNING -*-*-*-*-

THIS EQUIPMENT GENERATES AND USES RADIO FREQUENCY ENERGY AND IF NOT INSTALLED AND USED PROPERLY, THAT IS, IN STRICT ACCORDANCE WITH THE MANUFACTURER'S INSTRUCTIONS, MAY CAUSE INTERFERENCE TO RADIO AND TELEVISION RECEPTION. IT HAS BEEN TYPE TESTED AND FOUND TO COMPLY WITH THE LIMITS FOR A CLASS B COMPUTING DEVICE IN ACCORDANCE WITH THE SPECIFICATIONS IN SUBPART J OF PART 15 OF THE FCC RULES, WHICH ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST SUCH INTERFERENCE IN A RESIDENTIAL INSTALLATION. HOWEVER, THERE IS NO GUARANTEE THAT INTERFERENCE WILL NOT OCCUR IN A PARTICULAR INSTALLATION. IF THIS EQUIPMENT DOES CAUSE INTERFERENCE TO RADIO OR TELEVISION RECEPTION, WHICH CAN BE DETERMINED BY TURNING EQUIPMENT OFF AND ON, THE USER IS ENCOURAGED TO TRY TO CORRECT THE INTERFERENCE BY ONE OR MORE OF THE FOLLOWING MEASURES:

* REORIENT THE RECEIVING ANTENNA.
* RELOCATE THE COMPUTER WITH RESPECT TO THE RECEIVER.
* MOVE THE COMPUTER AWAY FROM THE RECEIVER.
* PLUG THE COMPUTER INTO A DIFFERENT OUTLET SO THAT THE COMPUTER AND RECEIVER ARE ON DIFFERENT BRANCH CIRCUITS.
* IF NECESSARY, THE USER SHOULD CONSULT THE DEALER OR AN EXPERIENCED RADIO/TELEVISION TECHNICIAN FOR ADDITIONAL SUGGESTIONS.

THE USER MAY FIND THE FOLLOWING BOOKLET PREPARED BY THE FEDERAL COMMUNICATIONS COMMISSION: "HOW TO IDENTIFY AND RESOLVE RADIO-TV INTERFERENCE PROBLEMS".
THIS BOOKLET IS AVAILABLE FROM THE U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402, STOCK No. 004-000-00345-4.

-*-*-*- CAUTION -*-*-*-

THIS EQUIPMENT MUST ONLY USE SHIELDED CABLES THAT ARE SUPPLIED BY PASSPORT DESIGNS, INC. THESE CABLES, AS SUPPLIED WITH THE UNIT, ARE DESIGNED TO REDUCE INTERFERENCE AS REQUIRED BY PROPER ENGINEERING AND MANUFACTURING METHODS AND BY REGULATIONS ENFORCED BY THE FEDERAL COMMUNICATIONS COMMISSION. PRIOR TO USING ANY INTER-CONNECTING CABLES WHICH WERE NOT SPECIFICALLY SUPPLIED WITH THE UNIT, CONTACT PASSPORT DESIGNS, INC. FOR INSTRUCTIONS.

1.0 INTRODUCTION

Passport's Musical Instrument Digital Interface (MIDI) card is a simple, reliable, linking device allowing data communication between MIDI-equipped musical instruments and Apple II, II+, //e, or Commodore 64 microcomputers.

This document is divided into four sections:

1.0 is introductory material;

2.0 covers installation and system configuration, and is "must" reading for all users;

3.0 covers common MIDI codes and beginning-level sample applications;

4.0 covers hardware control and addressing. It is intended primarily as a resource for experienced programmers.

Please note that the sections on MIDI codes and applications in this document are intended as introductory material only. For a thorough study of MIDI command codes and data types, we advise obtaining a copy of the MIDI SPECIFICATION from the International MIDI Association (see pg.36).

2.0 SYSTEM REQUIREMENTS & SETUP

The following hardware is required in order to use the MIDI interface:

- One Apple II-type computer with at least 48K RAM, video monitor, and one disk drive with disk controller card.

- OR -

- One Commodore 64 computer with monitor or television, and one disk drive.

- One MIDI-equipped synthesizer.
- At least two MIDI 5-pin DIN cables (supplied with unit).
- Amplifier, headphones, or other audio monitoring system.
- OPTIONAL - Programmable drum machine.
- OPTIONAL - MIDI Drum Sync Cable Kit. (DCK-1)
  Available from your local dealer, or directly from Passport Designs.
- OPTIONAL - Two cables with male RCA-type plugs, for use with TAPE SYNC. **(FOR APPLE MODEL MH-02A ONLY)**

In the Passport system, all instruments are connected in daisy-chain format. That is, the "first" instrument connects directly to the computer via MIDI IN and MIDI OUT. (Position in the chain has no relationship to CHANNEL NUMBERS of the receiving instruments.)

MIDI THRU on the "first" instrument connects to MIDI IN on the "second" instrument; MIDI THRU on the "second" instrument to MIDI IN on the "third" instrument; and so on down the chain.

Note that in this system, only the "first" instrument transmits to the computer. NO instruments transmit directly to each other. However, ALL instruments in the system receive from the computer, and thereby from the "first" instrument.

2.1 INSTALLATION: APPLE // VERSION

With the power off (pulling the plug is even better), open the top of your computer and locate slot #2. When facing the keyboard end of your computer, this is the second slot from the left side in Apple //e's, or the third slot from the left side in Apple II's and II+'s. Insert the MIDI interface card into slot #2; it can only fit one way. After the card is in the slot:

1.) Feed your MIDI 5-pin DIN cables through the back of your computer. (Apple //e owners: The cables will fit through hole #12 in the backplate of the computer.)

2.) Connect the Computer MIDI OUT to Synthesizer MIDI IN.

3.) Connect the Synthesizer MIDI OUT to Computer MIDI IN.

4.) OPTIONAL - If you have a non-MIDI drum machine, connect one end of the DCK-1 Drum Cable Kit to DRUM on the interface card, and turn to Section 2.4. IF YOU HAVE A MIDI DRUM MACHINE, DO NOT USE THIS CONNECTION!

5.) OPTIONAL (FOR MODEL MH-02A ONLY) - If you have a multi-track tape recorder and wish to use TAPE SYNC, connect RCA-type cables to the RCA IN and OUT jacks on the interface, and turn to Sec. 2.5.

6.) Close the top of your computer and reconnect power.

If you have multiple instruments in your system, only one synthesizer is connected directly to the computer. Additional instruments tie into the system via cables from MIDI THRU on one instrument to MIDI IN on the next.

2.2 INSTALLATION: COMMODORE 64 VERSION

Installing your MIDI interface is simple: With the power off, plug the MIDI interface label side up into the expansion slot on the right rear corner of your Commodore 64. Then:

1.) Connect the Computer MIDI OUT to Synthesizer MIDI IN.

2.) Connect the Synthesizer MIDI OUT to Computer MIDI IN.

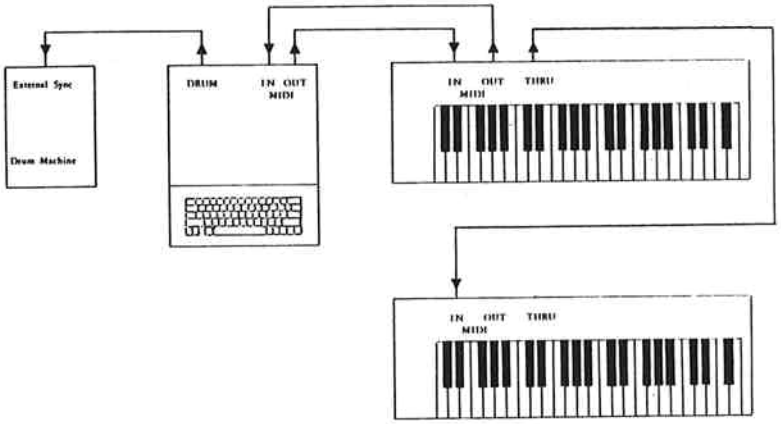
3.) OPTIONAL - If you have a non-MIDI drum machine, connect one end of the DCK-1 Drum Cable Kit to DRUM on the interface card, and turn to Section 2.4. IF YOU HAVE A MIDI DRUM MACHINE, DO NOT USE THIS CONNECTION!

If you have multiple keyboards in your MIDI system, only one synthesizer is connected directly to the computer. Additional instruments tie into the system via cables from MIDI THRU on one instrument to MIDI IN on the next.

2.3 SYSTEM DIAGRAM

Remember, this is just the physical wiring of the system. The actual destination of transmitted data is determined by the MIDI CHANNEL assignment of the transmitting software, and the CHANNEL and MODE controls of the receiving instruments in the system; not by the interface.

Also remember, in multiple-instrument systems the instrument connected directly to the computer is the recording keyboard. All other instruments are playback-only.



2.4 CONNECTING DRUM MACHINES (OPTIONAL)

Using the Passport DCK-1 Drum Sync Cable Kit, most brands of drum machine can be integrated into your MIDI system. With Drum Sync, your MIDI system generates the CLOCK and START/STOP signals for the drum machine. Your drum machine must be capable of accepting external CLOCK signals for this to work.

The Drum Sync Cable Kit consists of two parts: 1) a male-to-male DIN 5-pin cable, and 2) a female DIN "Y" cable terminating with 2 quarter-inch phone plugs.

MIDI-equipped Drum Machines

The signals available through the Drum Sync jack are analog, not MIDI information. DO NOT CONNECT DRUM SYNC DIRECTLY TO MIDI IN of MIDI-equipped drum machines.

Drum machines equipped with MIDI are connected via MIDI OUT, IN, and THRU jacks, just like any other synthesizer.

Roland or Korg Drum Machines

If you use a Roland, Korg, or other non-MIDI drum machine equipped with a 5-pin DIN socket for external sync:

1) Connect the male-to-male DIN cable from the DRUM output of the MIDI interface to the external sync input on the drum machine;

2) Set the SYNC INPUT/OUTPUT switch to INPUT.

Linn, Oberheim, Drumulator, and others

If you're using a drum machine without a DIN sync connector, you must use both cables in the Drum Sync Cable set.

1) Connect the male-to-male DIN cable to the DRUM output of the MIDI interface;

2) Attach the female DIN "Y" cord to the male DIN cable;

3) Insert the straight phone plug into the CLOCK input of the drum machines, and the right angle plug into the START/STOP (or footswitch) input of the drum machine.

Drum Sync Clock Rate

The MIDI standard for timing pulses is 24 CLOCK pulses per quarter note beat. Different non-MIDI drum machines use different clock rates. Older E-mu Drumulators use 12 clocks/quarter note; new Drumulators and Rolands use 24 clocks per quarter note. Korg uses 48 clocks per quarter note; Oberheim uses 96 clocks/quarter note.

When running commercial software, you will need to allow for these differences in programming the drum machine for synchronization with the MIDI system.

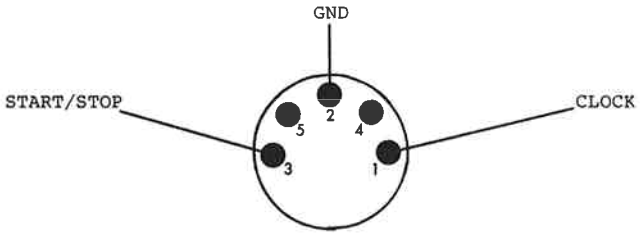
Drum synchronization is not automatic. If you are writing your own software for this interface, you will need to make provision for generating drum sync signals.

Drum Sync Pinout

When operating under software control the Drum Sync connector outputs two signals:

START/STOP - Normally high, pulses low (& returns high) to toggle running state of drum machine;

CLOCK - - - +5V at a rate of 24 clocks/qtr. note. Normally off, turns on when START is sent.



2.5 CONNECTING & USING TAPE SYNC (OPTIONAL)

(FOR APPLE MODEL MH-02A ONLY)

One of the most significant features of the MH-02 Interface is TAPE SYNC. When used with the appropriate software, this interface will WRITE sync information onto one track of a multi-track tape deck. The tape can then be played back and the sync track READ by the interface, producing perfect synchronization of multiple sequencer and/or drum machine tracks, regardless of when they're recorded.

The first crucial part of the process is appropriate software. TAPE SYNC is not automatic; you must have a sequencer program which supports it. At this time, MIDI/4 PLUS and MIDI/8 PLUS will make use of the TAPE SYNC features of the MH-02 Interface; owners of older versions of MIDI/4 are advised to contact Passport Customer Service and ask about the Software Update Program.

While the operation of TAPE SYNC is described in detail in our software manuals, we'd like to offer a few pointers here:

CONNECTIONS: The TAPE SYNC jacks are 2 RCA-type female jacks on the MH-02 Interface. For best results, bypass your mixer and connect TAPE SYNC OUT directly to LINE IN of the tape deck channel you'll be using for the sync track. Connect LINE OUT of the tape deck to TAPE SYNC IN on the interface. This will provide a cleaner signal. ABOVE ALL, NEVER USE DOLBY OR DBX NOISE REDUCTION ON THE SYNC TRACK!

THE TAPE: The integrity of your tape is crucial. Splices, drop-outs, or wrinkles in the tape can render the recorded sync track worthless.

BEST PROCEDURE: Make sure all tracks are recorded in the sequencer before going to tape. No sequencer track can be longer than the SYNC TRACK on the tape, and the sync code is only written onto the tape while the sequencer is running. Therefore, the

sync track is only as long as longest track in the sequencer at the time the sync track is recorded.

a.) Don´t turn on the TAPE SYNC WRITE in the software until you´re ready to record the SYNC TRACK.

b.) Watch your VU meters. TAPE SYNC works best when the sync track is recorded and played back at -3 dB.

c.) TAPE SYNC works best if you record the SYNC TRACK first, with the sequencer running, but not recording the music tracks. Record just the WRITE TAPE SYNC output. Then set the sync track to normal playback mode and begin laying down the sequencer and drum machine tracks while READING the sync track off the tape.

We recommend this procedure because we´ve found that many low-priced multi-track tape decks -- particularly integrated mixer/cassette units -- have relatively poor audio quality in "simul-sync" playback mode and the sync track data can get lost in the noise.

d.) Sync track data consists of a "ready" code, followed by a "running" code. Both codes must be present for tape sync to work correctly. This means, when recording the sync track, start the tape rolling before you start sequencer playback. When overdubbing later tracks, never "cue up" the tape to the first beat of the song; always start at least 1 beat before the beginning of the song in order to catch the "ready" code on the sync track.

e.) There is no TEMPO information in the sync track. It´s up to you to set the TEMPO of your sequencer correctly.

f.) Provided they are connected to your MH-02 Interface and the "utilities" in your MIDI/4 PLUS or MIDI/8 PLUS program are set correctly, external MIDI sequencers and both MIDI and non-MIDI drum machines will start and run in sync with the tape track.

3.0 INTRODUCTION TO PROGRAMMING THE **MIDI** INTERFACE

While we haven't the space to thoroughly cover MIDI programming in this manual, we will touch briefly on the subject for those users who decide to write their own programs.

Why? Passport is a software publisher, and like most publishers we are interested in good independently written programs to publish. That's our business. If you'd like to know more, write us and request our submissions guidelines.

The **MIDI** interface is easily accessed from assembly language, BASIC, and other programming languages that allow direct manipulation of Input/Output (I/O) addresses. As "music" is an extremely time-sensitive application, most professional programming is done in assembly language. However, even beginning BASIC programmers can write entertaining and enjoyable MIDI music programs. (Please note: While Applesoft or Commodore BASIC programs can play MIDI musical instruments, they are not able to record.)

The program that follows is a simple demonstration of MIDI playback under Applesoft BASIC control. When typed into your Apple and RUN it "plays" a two octave note sequence at a steady tempo, pauses to allow entry of a transposition value, then replays the sequence in transposition.

This program, with slight modification, can also be typed in and RUN on a Commodore 64. Most importantly, you must change the definitions of ACIA CONTROL and DATA register addresses in lines 3060 and 3070 to read:

```
3060  AC = 56840
3070  AD = 56841
```

It is also necessary to omit lines 1240, 1310, 2050, 2080, and 2110, as Commodore BASIC doesn't accept HTAB and VTAB commands.

```
100  REM   ------------------------------
110  REM  -   MIDI DATA PLAYBACK DEMO  -
120  REM  -        APPLESOFT BASIC     -
130  REM  -(C)1984 PASSPORT DESIGNS INC-
140  REM  -     ALL RIGHTS RESERVED    -
150  REM  -                            -
160  REM  - VERSION 1.1          9/6/84 -
170  REM  - PROGRAMMER:   BRUCE BETHKE -
200  REM  ------------------------------
210  REM
220  GOSUB 2000: REM  DRAW SCREEN
230  GOSUB 3000: REM  DEFINE VARIABLES
235  GOTO 1000: REM  BEGIN PLAYBACK
240  REM
250  REM
260  REM  ------------------------------
270  REM  -          SUBROUTINES        -
280  REM  ------------------------------
290  REM
300  REM  --- WAIT FOR ACIA TO CLEAR ---
310  FOR TIM = 1 TO 4: NEXT : RETURN
320  REM
330  REM
400  REM  -- WAIT FOR NOTE'S DURATION --
410  FOR TIM = 1 TO DUR: NEXT : RETURN
420  REM
430  REM
500  REM   ------TURN OFF OLD NOTE ------
510  POKE AD,OFFKY: GOSUB 310
520  POKE AD,OLDKY: GOSUB 310
530  POKE AD,VOFF: GOSUB 310
540  RETURN
550  REM
560  REM
600  REM  ----- TURN ON NEW NOTE  -----
610  POKE AD,KEYDN: GOSUB 310
620  POKE AD,NEXKY: GOSUB 310
630  POKE AD,VMAX: GOSUB 310
640  RETURN
650  REM
660  REM
1000  REM  -------------------------
1010  REM  -     BEGIN PLAYBACK     -
1020  REM  -------------------------
1030  REM
1040  FOR N = 1 TO 24
1050  READ NEXKY
1060 NEXKY = NEXKY + T: REM   TRANSPOSE
1070  GOSUB 510: REM  TURN OFF PREVIOUS NOTE
1080  GOSUB 610: REM  PLAY CURRENT NOTE
1090  GOSUB 410: REM  WAIT ONE NOTE'S DURATION
1100 OLDKY = NEXKY: REM    CURRENT NOTE BECOMES PREVIOUS NOTE
1110  NEXT N: REM  GET NEXT NOTE
1120  REM
```

```
1130  REM  ----------------------------
1140  REM  - AT END OF DATA, TURN OFF -
1150  REM  - LAST NOTE PLAYED         -
1160  REM  ----------------------------
1175  GOSUB 410
1180  POKE AD,OFFKY: GOSUB 310
1190  POKE AD,OLDKY: GOSUB 310
1195  POKE AD,VOFF: GOSUB 310
1197  REM
1200  REM   ----------------------------
1210  REM  -  EXIT / TRANSPOSE PROMPT -
1220  REM  ----------------------------
1240  VTAB 12: HTAB 10
1250  PRINT "CONTINUE (Y/N)";: INPUT A$
1260  IF (A$ = "Y" OR A$ = "y") THEN 1300
1270  HOME : END
1290  REM
1300  REM  ---- CONTINUE PLAYBACK ----
1310  VTAB 14: HTAB 3
1320  PRINT "TRANSPOSE VALUE (+/- 0..36)";: INPUT A$
1330 T =  VAL (A$)
1340  IF (T >  =  - 36) AND (T <  = 36) THEN 1400
1350  HOME : GOSUB 2000: GOTO 1310
1360  REM
1400  RESTORE : GOTO 1040
1410  REM  ======================
1500  REM
1900  REM
2000  REM  -------------------------
2010  REM  -      DRAW SCREEN      -
2020  REM  -------------------------
2040  HOME : PRINT  CHR$ (7): REM  BEEP!
2050  VTAB 1: HTAB 7: INVERSE
2060  PRINT "MIDI PLAYBACK DEMONSTRATION"
2070  NORMAL
2080  VTAB 3: HTAB 5: INVERSE
2090  PRINT "(C) 1984 PASSPORT DESIGNS, INC."
2100  NORMAL
2110  VTAB 5: HTAB 5: PRINT "********************************"
2120  RETURN
2130  REM
2140  REM
3000  REM  ----------------------------
3010  REM  -      DEFINE VARIABLES     -
3020  REM  ----------------------------
3030  REM
3040  REM  ACIA CONTROL AND DATA REGISTERS
3050  REM  WITH MIDI CARD IN SLOT#2

3060 AC = 49320: REM  ACIA CONTROL
3070 AD = 49321: REM  ACIA DATA
3080 OFFKY = 128: REM  KEY OFF STATUS BYTE
3090 KEYDN = 144: REM       KEY ON STATUS BYTE
3100 VOFF = 0: REM  KEY VELOCITY = 0
3110 VMAX = 127: REM  KEY VELOCITY = 127
3120 OLDKY = 0: REM  "OLD" KEY ID NUMBER
3130 NEXKY = 0: REM   "NEW" KEY ID NUMBER
3140 DUR = 50: REM  NOTE DURATION
3150 T = 0: REM  TRANSPOSITION VALUE
3160 A$ = "": REM  INPUT STRING VARIABLE
```

```
3180  REM
3200  REM  ------------------------------
3210  REM  -  RESET & CONFIGURE ACIA    -
3220  REM  ------------------------------
3230  REM
3240  POKE AC,19: POKE AC,17
3250  RETURN
3300  REM
3400  REM
4000  REM  ------------------------------
4010  REM  -  KEY ID (NOTE) DATA TABLE -
4020  REM  ------------------------------
4030  REM
4040  DATA  60,61,62,63,64,65
4050  DATA  66,67,68,69,70,71
4060  DATA  72,71,69,68,67,66
4070  DATA  65,64,63,62,61,60
4080  REM
4090  REM
```

3.1 MIDI CODES

The following 6850 ACIA and MIDI codes are used in the example program (the example program does not use the 6840 Timer). MIDI codes shown are valid only if the synthesizer is set to receive on MIDI CHANNEL 01.

For a complete listing of MIDI codes, see the IMA MIDI Specification.

ACIA CODE 19 -- RESET (line 3240)

This value poked into the ACIA control register resets the ACIA. This must be done before any other configuration code can be sent to the ACIA.

ACIA CODE 17 -- MASK INTERRUPTS (line 3240)

Poking this value into the ACIA control register sets the ACIA to generate no interrupts to the Apple.

MIDI CODE 128 -- KEY OFF (line 3080, 510)

MIDI messages consist of a status byte followed by one or two data bytes (depending on the type of message). Status bytes are always greater than 127; data bytes are always less than 128.

Both status bytes and data bytes are poked into the ACIA data register.

Code 128 (KEY OFF) must be followed by 2 data bytes: a key i.d. number (line 520) indicating which note to turn off, and a key velocity (line 530).

MIDI CODE 144 -- KEY ON (line 3090, 610)

KEY ON also must be followed by 2 data bytes: a key i.d. (line 620) indicating which note to turn on, and a key velocity (line 630). Standard velocity for all notes in this program is defined in line 3110 as 127.

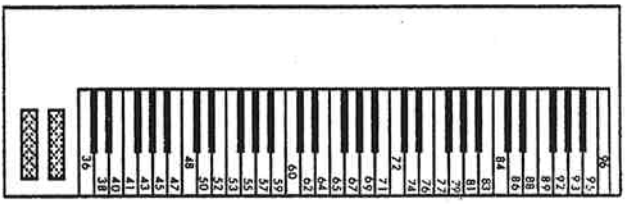
Many instruments without velocity sensitive keyboards ignore the transmitted velocity byte and use a default velocity of 64.

MIDI CODE 192 -- CHANGE PRESET

This code is not used in this program, but is included in case you want to experiment with it. The CHANGE PRESET status byte is followed by 1 data byte indicating the new preset number.

While presets can be changed under MIDI control, you're still limited to the number of presets in the synthesizer.

The valid range of a KEY NUMBER (ON or OFF) data byte is from 0 to 127. All keys are numbered in relation to Middle C, which is 60. On an industry-standard 5-octave keyboard the lowest physical key is C2 (36) and the highest is C7 (96). Depending on the instrument, key numbers outside the instrument's range will either be ignored or transposed into normal range.



4.0 TECHNICAL NOTES

Passport MH-02 Interfaces are "clean and elegant" devices consisting of a 6850 Asynchronous Communications Interface Adapter (ACIA), a 6840 Programmable Timer Module (PTM), and computer bus interface circuitry. The simple design of the interface imposes the fewest possible restrictions on application software. Given proper software, any MIDI-equipped device can be addressed and controlled with this interface.

This section covers specific I/O addressing of the Apple and Commodore versions of the MH-02 Interface, frequently used ACIA and PTM codes, and an introduction to the specifics of the ACIA and PTM.

For a complete description of the operating characteristics of the MC6840 PTM and MC6850 ACIA, see the 6840 and 6850 data sheets (available directly from Motorola).

4.1 APPLE // HARDWARE ADDRESSES

All Input/Output hardware addresses in the Apple II are slot dependent. Memory locations $C000 through $CFFF are allocated for I/O addresses; exact slot addresses are computed as follows:

BASE ADDRESS = $C080 + $n0

Where n equals the slot number (1 thru 7). The addresses given here are based on the assumption that the interface is installed in Slot #2 (base address $C0A0), the "standard" slot used for serial communications devices in the Apple.

DECIMAL	HEX	DESCRIPTION
49312	C0A0	6840 Timer control register 1
49313	C0A1	6840 Timer control register 2
49314	C0A2	High Byte Timer 1 value
49315	C0A3	Low Byte Timer 1 value
49316	C0A4	High Byte Timer 2 value
49317	C0A5	Low Byte Timer 2 value
49318	C0A6	High Byte Timer 3 (RESERVED)
49319	C0A7	Low Byte Timer 3 (RESERVED)
49320	C0A8	6850 control register
49321	C0A9	6850 data register
49326	C0AE	Drum sync SET
49327	C0AF	Drum sync CLEAR

4.2 COMMODORE 64 HARDWARE ADDRESSES

I/O hardware addresses in the Commodore 64 are absolute, as there is only one expansion slot.

DECIMAL	HEX	DESCRIPTION
56832	DE00	6840 Timer control register 1
56833	DE01	6840 Timer control register 2
56834	DE02	High Byte Timer 1 value
56835	DE03	Low Byte Timer 1 value
56836	DE04	High Byte Timer 2 value
56837	DE05	Low Byte Timer 2 value
56838	DE06	High Byte Timer 3 (RESERVED)
56839	DE07	Low Byte Timer 3 (RESERVED)
56840	DE08	6850 control register
56841	DE09	6850 data register
56880	DE30	Drum sync SET
56888	DE38	Drum sync CLEAR

4.3 INITIALIZATION & OPERATION CODES

The following values are used to set up the 6840 PTM and the 6850 ACIA to function in various capacities, and to control their operation.

Values relevant to the 6850 are poked into the 6850 Control and Transmit Data Registers. Values relevant to the 6840 are poked into the 6840 Timer Control Registers.

Timer & ACIA must be reset before configuration.

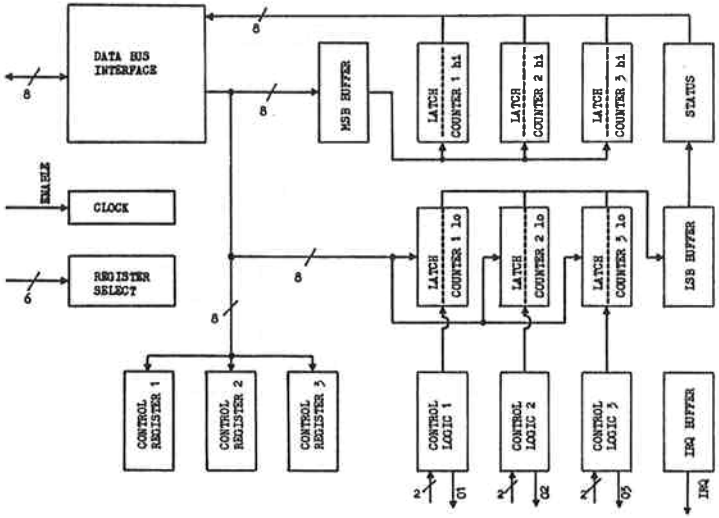
	DECIMAL	HEX	BINARY	DESCRIPTION
6840 PTM	0	00	0000 0000	Let timers run
	1	01	0000 0001	Turn off timers
	67	43	0100 0011	Reset the 6840 PTM
6850 ACIA	17	11	0001 0001	No interrupts generated
	19	13	0001 0011	Reset ACIA
	21	15	0001 0101	Turn off ACIA
	49	31	0011 0001	Generate interrupts during Send only
	145	91	1001 0001	Generate interrupts during Receive only
	177	B1	1011 0001	Generate interrupts during both Send & Receive

4.4 MC6840 PROGRAMMABLE TIMER MODULE (PTM)

The Motorola MC6840 is a versatile timer module that is completely accessible to the programmer, appearing as memory locations in the I/O space. The module contains three timers which are independently programmable, cyclic in operation, controllable by external events or software, and are accessible by the microprocessor at any time.

Typically, a timer is loaded by storing two bytes of time data into its respective Counter Latch, which is subsequently transferred into the Counter during the initialization phase. The counter is decremented every clock period until a predetermined condition causes it to halt or recycle.

SIMPLIFIED BLOCK DIAGRAM OF THE MC6840



CONTROL REGISTERS

There are three Write-Only registers on the 6840 that are used to "control" the operation of the three onboard timers.

Control Register #2 is uniquely mapped in the I/O address space and may be addressed at any time.

The remaining two control registers (#1 & #3) share the same address. They are addressed by setting all of the Register Select inputs, the three least significant bits (lsb) of the address, to logic zero and using bit 0 of Control Register #2 as an additional addressing bit.

Given the Register Select lines are zero, if bit 0 of Control Register #2 is logic one, Control Register #1 is selected. If Control Register #2 bit 0 is logic zero, then Control Register #3 is selected (reserved).

TABLE #1

Register Select Bits 2	1	0	Operations R/$\overline{W}$ = 0	R/$\overline{W}$ = 1
0	0	0	CR2 bit 0=0, Write CR #3 CR2 bit 0=1, Write CR #1	No operation
0	0	1	Write Control Register#2	Read Status register
0	1	0	Write MSB buffer register	Read Timer #1 counter
0	1	1	Write Timer #1 latches	Read LSB buffer register
1	0	0	Write MSB buffer register	Read Timer #2 counter
1	0	1	Write Timer #2 latches	Read LSB buffer register
1	1	0	Write MSB buffer register	Read Timer #3 counter
1	1	1	Write Timer #3 latches	Read LSB buffer register

The lsb of Control Register #1 is used as an Internal Reset bit. Setting this bit to logic one causes all counters to be set with the contents of their associated latches, all counter clocks to be disabled, and all timer outputs and interrupt flags to be reset. Conversely, when bit 0 of Control Register #1 is set to logic zero, all of the timers are allowed to operate in the modes specified by the remaining bits in the respective control registers.

The lsb of Control Register #3 is used as a selector for a divide-by-8 prescaler used with Timer #3 only.

TABLE #2

Control Register 1

Bit 0	Function: Internal Reset bit
0	All timers allowed to operate
1	Hold timers in present state

Control Register 2

Bit 0	Function: Control Register Address bit
0	Control Register #3 may be written
1	Control Register #1 may be written

Control Register 3

Bit 0	Function: Timer #3 Clock Control
0	Timer 3 clock is NOT prescaled
1	Timer 3 clock is prescaled (divide by 8)

The remaining bits (1-7) of each of the Control Registers select common functions with the effect restricted to the timer associated with the respective Control Register.

Bit 1: Selects the clock source for the timer.

Bit 2: Determines whether the binary data contained in the Counter Latches is treated as one single 16-bit word, or as two 8-bit Bytes.

Bits 3, 4, and 5: Select the operating modes of the respective timers: Continuous, Single-shot, Frequency comparison, or Pulse Width Comparison. Continuous (the most commonly used mode) is established by writing zeroes into bits 3 and 5 of the corresponding Control Register.
A Timer Reset (Internal Reset: Control Register #1, bit 0=1; or an External Reset=0) condition results in Counter initialization.

Bit 6: Is an Interrupt Mask and is used in conjunction with the interrupt flag bits of the Status Register It must be set (=1, logic one) if interrupts are to be recognized from the associated timer.

Bit 7: Enables the corresponding Timer Output, (This mode is not used).

TABLE #3

Control Registers 1 through 3

Bit	State	Function
1	0	Timer n* uses external clock source
	1	Timer n uses (Enables) internal clock src
2	0	Timer n 16-bit mode
	1	Timer n dual 8-bit mode
3	-	Timer n Counter Mode & Interrupt Control
4	-	(see Table #4)
5	-	
6	0	Timer n IRQ masked
	1	Timer n IRQ enabled
7	0	Timer n Timer Output masked
	1	Timer n Timer Output enabled

* n equals 1, 2, or 3

TABLE #4

Control Registers 1 through 3

Bits 5	4	3	Function
0	*	0	Continuous
1	*	0	Single-Shot
*	0	1	Frequency Comparison
*	1	1	Pulse Width Comparison

* Defines additional timer functions

STATUS REGISTER (READ ONLY)

The internal Status register contains four Interrupt flags (or bits). The lowest three flags (bits) are assigned to one of the three timers. That is, bit 0 is assigned to Timer #1, bit 1 to Timer #2, and bit #2 to Timer #3. (Bits 3 to 6 are not used.)

Bit 7 is a composite Interrupt flag and will be set if any of the interrupt flags is set and bit 6 of the corresponding control register is also set.

An interrupt flag is cleared by a Timer Reset condition, an External Reset, or an Internal Reset. It will also be reset by a Read Timer Control Command if, and only if, the Status Register has previously been read while the interrupt flag was set.