ALLAFORD RESEARCH

COCO & MIDI =

MUSIC

THE COCO MIDI INTERFACE

Technical Information on using the CoCo Midi Interface with your own programs.

By Allen C. Huffman

-

Introduction:

This document was created for those of you who want to use the CoCo Midi Interface with your own programs. We will attempt to describe methods of using the pack through Basic and Assembly Language.

Background:

The CoCo Midi Interface uses a Motorola 685Ø ACIA to control I/O between the computer and a Midi device. It is programmed in the same manner as the Tandy RS232 Program Pak or Speech/Sound Pak. When the pack is plugged in, two memory locations control everything:

$FF6E (6539Ø) - Status Register. This location is used for initializing the pack for MIDI transmissions, and for checking to see if the pack is ready to send to receive a byte of data.

$FF6F (65391) - Data Register. This location is where all incoming and outgoing data bytes go.

Using the Pack:

Before the pack is able to send or receive Midi data, it must be initialized. This is done by sending a $3 then a $15 to the Status Register. In Basic it looks like this:

```
1Ø POKE &HFF6E,&H3 : POKE &HFF6E,&H15
```

...or for all you decimal-only programmers...

```
1Ø POKE 6539Ø,3 : POKE 6539Ø,21
```

The Assembly Language version looks like this:

```
SETUP  LDA   #$3
       STA   $FF6E
       LDA   #$15
       STA   $FF6E
       RTS
```

After this, the pack is ready for Midi transmissions. To send a byte, simply store it at $FF6F. To receive a byte, simply get it from the same location. In order to tell if $FF6F is empty so you can send, you must check bit 2 of the status register. In Basic, it looks like this:

```
1500 REM Send A to pack
1505 IF (PEEK(&HFF6E) AND 2)=0 THEN 1505
1510 POKE &HFF6F,A
1515 RETURN
```

The Assembly Language version looks like this:

```
SENDAT  LDB   $FF6E      Load accumulator B with Status.
        BITB  #2         Check bit #2.
        BEQ   SENDAT     If 0, buffer full.  Go back.
        STA   $FF6F      Otherwise, send A.
        RTS              Return from subroutine.
```

Receiving data from the pack is just as simple, except you check bit 1. Here it is in Basic:

```
1000 REM Get byte from pack
1005 IF (PEEK(&HFF6E) AND 1)=0 THEN 1005
1010 A=PEEK(&HFF6F)
1015 RETURN
```

The Assembly Language version looks like this:

```
GETDAT  LDA   $FF6E      Load accumulator A with Status.
        BITA  #1         Check bit #1.
        BEQ   GETDAT     If 0, nothing received.  Go back.
        LDA   $FF6F      Otherwise, load A with byte received.
        RTS              Return from subroutine.
```

NOTE: Receiving Midi data in Basic doesn't work too well. By the time you get one byte of data, many other bytes could have already been sent and lost. Basic is just too slow to handle Midi input. Assembly, on the other hand, works great.

A Routine:

The following page contains a short machine language routine that can be used to write custom Midi programs such as librarians. It contains two main routines.

The first will receive bytes from the pack and store them, starting at $7100, until it receives a byte $F0 or greater, such as the EOX byte ($F7).

The other routine does the opposite. It starts sending all data at $7100 until it encounters $F0 or greater.

The rest is up to you. Please note that while receiving data, if no end byte ($F0 or greater) is encountered, the routine will continue to store incoming bytes in memory until it writes over the I/O portion of memory which will probably cause a system crash.

Assembly Language Listing of Midi Get/Send Routine:

```
00100
00110 * MIDI ROUTINE for CoCo Midi Interface
00120 * Version 1.0
00130
00140 MSTAT   EQU    $FF6E          From here on, MSTAT refers to $FF6E
00150 MDATA   EQU    $FF6F                       and MDATA refers to $FF6F
00160
00170         ORG    $7000          Program starts at $7000
00180
00190 START1  BRA    GET            GET routine at $7000
00200 START2  BRA    SEND           SEND routine at $7002
00210
00220 GET     BSR    INIT           Jump to INIT routine
00230         LDX    #$7100         Make X point to start of "buffer"
00240 GLOOP   BSR    GETDAT         Loop.  Jump and get byte from pack
00250         STA    ,X+            Store byte at X, increment X
00260         CMPA   #$F0           Compare A to $F0
00270         BLS    GLOOP          If A < $F0, return to GLOOP
00280         RTS                   Return from subroutine
00290
00300 SEND    BSR    INIT           Jump to INIT routine
00310         LDX    #$7100         Make X point to start of "buffer"
00320 SLOOP   LDA    ,X+            Loop.  Load A from X, increment X
00330         BSR    SENDAT         Jump and send A to pack
00340         CMPA   #$F0           Compare A to $F0
00350         BLS    SLOOP          If A < $F0, return to SLOOP
00360         RTS                   Return from subroutine
00370
00380 INIT    LDA    #$3            Load A with $3
00390         STA    MSTAT          Store it at MSTAT
00400         LDA    #$15           Load A with $21
00410         STA    MSTAT          Store it at MSTAT
00420         RTS                   Return from subroutine
00430
00440 GETDAT  LDA    MSTAT          Load A with Status
00450         BITA   #1             Test bit 1
00460         BEQ    GETDAT         If it's 0, go back to GETDAT
00470         LDA    MDATA          Byte received, load it in A
00480         RTS                   Return from subroutine
00490
00500 SENDAT  LDB    MSTAT          Load B with Status
00510         BITB   #2             Test bit 2
00520         BEQ    SENDAT         It it's 0, go back to SENDAT
00530         STA    MDATA          Pack clear, send A
00540 THEEND  RTS                   Return from subroutine
00550
00560         END                   That's it!
```

Here is a Basic loader for the above listing:

```
10 CLEAR 200,&H7000
15 X=0:FORA=&H7000 TO&H7040:READA$:B=VAL("&H"+A$):POKE A,B:X=X+B:NEXT:PRINTX
20 DATA 20,2,20,E,8D,1A,8E,71,0,8D,20,A7,E0,81,F0,23,F8,39,8D,C,8E,71,0,A6,80,
        8D,1B,81,F0,23,F8,39,86,3,B7,FF,6E,86,15,B7,FF,6E,39,B6,FF,6E,85,1,27,
        F9,B6,FF,6F,39,F6,FF,6E,C5,2,27,F9,B7,FF,6F,39
```

EXEC &H7000 - Execute GET routine
EXEC &H7002 - Execute SEND routine

When you run the Basic version of the program, it should print 8077. That is the checksum of the data statements. If you get anything else, recheck your typing.

Acknowledgements:

I would like to thank Lester Hands for sending me the initial information on using the interface, and Mark Thomas for digging up the technical specs on the MC6850. Thanks also to Cecil Houk for allowing me to present this information, and finally, thanks to you for supporting the future of music on the Color Computer!

RULAFORD RESEARCH PRODUCT LIST
6-27-89

MUSICA	$24.95	A 4 part software music program. Prints music too. Any model COCO - 64K and up.
MUSIC LYBRARY	$ 5.ØØ	Ready to play MUSICA files. Over 9ØØ titles to choose from; 47 disks. List on disk.
MUSICA MIDI (JUKEBOXM)	$29.95	A program to play MUSICA files via midi. Includes COCO to midi cable. Plays tapes too.
LYRA CONVERT	$ 9.95	A program to convert your MUSICA disk files to LYRA format.
LYRA/LRYAPRINT	$59.95	An 8 part midi music editor. Enter the music with a joystick or mouse. Prints music too.
LYRA COMPANION	$ 9.95	A 1ØØ+ page book with lots of info on music, LYRA, and midi. A must if you're new to midi.
LYRA UPGRADE	$25.ØØ	Upgrade to the latest version. Includes a 12 month update warranty. (return your original disk with order.)
LYRA LYBRARY	$14.95 per disk	Ready to play LYRA music files for midi. 376 titles on 18 disks (as of 6-27-89) $125.ØØ for all! (see Dec '88 RAINBOW)
COCO MIDI 2*	$129.95	A 16 track midi sequencer/editor. Includes COCO MIDI INTERFACE.
COCO MIDI 3*	$15Ø.ØØ	A 1Ø track midi sequencer/editor that's very easy to use. Includes COCO MIDI INTERFACE.
COCO MIDI 3**	$6Ø.ØØ	Software only.
COCO MIDI* INTERFACE	$1ØØ.ØØ	Hardware interface required by COCO MIDI 2/3 and some editors and librarians.
opt 1	$1Ø.ØØ	Adds midi "thru"
opt 2	$1Ø.ØØ	Adds 2nd, switched, midi "in"
opt 3	$2Ø.ØØ	Opt 1 & 2 (opt 1 becomes switched out - thru add $5.ØØ to have each option installed in your existing interface.
"Y" CABLE	$24.95	If you don't have a Multipack Interface...
FBØ1CALC	$19.95	Configuration editor for the Yamaha FB-Ø1.
FBEDIT**	$29.95	Voice editor for Yamaha FB-Ø1 (COCO 3 ONLY!)
CZ LIBRARIAN**	$29.95	Librarian for Casio CZ-1Ø1, CZ-1ØØØ/5ØØØ.
DX LIBRARIAN**	$29.95	Librarian for Yamaha DX-7, DX-27/1ØØ.
K1 LIBRARIAN**	$29.95	Librarian for Kawai K1.

Except for FBEDIT, ALL programs run on COCO's 1, 2 or 3 - 64K and up.

* REQUIRES MULTIPAK OR "Y" CABLE
** REQUIRES COCO MIDI INTERFACE (which requires Y cable or Multipak)