

INTERFACING EXPANSION SOFTWARE TO THE MODEL 64 SEQUENCER

LOADING EXPANSION SOFTWARE:

The Model 64 Sequencer's "expansion software" option allows new programs to be loaded from diskette, and complete access to the cartridge hardware.

The user selects the expansion software option by going to the main sequence or main song menu and pressing the letter "E". You then answer "Y" (yes) to:

"OK to load expansion software?"

The next question is:

"Are you using a disk drive?"

When you press "Y" for yes, the program turns on the disk drive, and begins looking for a file whose first character is an inverse "E" (ASCII character #05). Upon finding it, the file is loaded in the Commodore starting at memory location \$8000. Code should be assembled at address \$8000 and then stored to disk from that address. This will assure its being reloaded at \$8000. The first two bytes (at \$8000 and \$8001) must be a vector (lo byte, hi byte order) to the start address of the expansion software. For example, your code can begin at address \$8005, in which case the first two bytes of the file will be \$05, \$08. The next three bytes (at \$8002, \$8003, and \$8004) must be #\$4C, #\$50, and #\$53.

The space reserved for expansion software is \$8000-\$9000. If the expansion software needs more space, it must load itself into other RAM. Additional RAM available is \$801 thru \$7FFF, \$9000 thru \$9FFF, and \$C000 thru \$CFFF.

The simplest way to exit is to POWER DOWN since all sequence data is lost anyway.

ACCESSING THE 64 SEQUENCER CARTRIDGE HARDWARE:

There are two integrated circuits in the 64 Sequencer that are responsible for interface to the outside world:

-MIDI IN and MIDI OUT are handled by a 6850 ACIA.

-The footswitch, drum input, LEDs, and slide switches are handled by a 6520 PIA.

Both the ACIA and PIA are initialized for proper operation upon power up of the 64 Sequencer, so unless you want to reconfigure the system, there's no need to reinitialize the ICs.

6850-Asynchronous Communications Interface Adapter (ACIA)
Made by Motorola, AMI, Fairchild

--To initialize the 6850 for proper MIDI transmission you must use the ACIA "control register" (\$DE00).

Example:

```
LDA #$03 ;Master reset
STA $DE00
LDA #$15 ;Setup for 1 start bit+8 bits+1 stop bit and
STA $DE00 ; 31.25 Kbaud
```

--To send MIDI OUT:

To transmit MIDI data, you must store the desired MIDI byte into the ACIA's "transmit data register" (\$DE01). But before you can do that you make sure the ACIA is finished transmitting the previous byte. To do this, you must read the ACIA's "status register" (\$DE02). Bit 1 of the status register is the "transmit data register empty" flag. If this bit is "1", then the transmit register is empty and is ready to transmit another byte. If it's "0", then it's not ready, and you must wait.

It takes about 320 microseconds to transmit (or receive) one MIDI byte.

Example:

```
LOOP LDA $DE02      ;Read ACIA status register  
      AND #$02      ;Check transmit empty flag  
      BEQ LOOP      ;Loop until empty  
      LDA MIDIBYTE  ;Load in byte to send  
      STA $DE01      ;Send MIDI OUT
```

--To receive MIDI IN:

Incoming MIDI data is accessed by reading the ACIA's "receive status register" (\$DE03). But before you can do that you must make sure the ACIA has fully read in the byte. To do this, you must read the ACIA's "status register" (\$DE02) and check bit 0, which is the "receive data register full" flag. If this bit is "1", the receive register is full and ready to read. If it's "0", you must wait.

Example;

```
LOOP LDA $DE02      ;Read ACIA status register  
      LSR          ;Check receive full flag  
      BCC LOOP      ;Loop until ready  
      LDA $DE03      ;Read MIDI IN
```

Additionally, the IRQ output pin of the ACIA is connected to the IRQ input of the Commodore 64, so that the MIDI handler can be interrupt driven. Consult a 6850 data sheet for details on interrupt operation.

6520-Peripheral Interface Adapter (PIA)

Made by Synertek, Rockwell

--To initialize the 6520 for proper operation:

Example:

```
LDA #$00      ;Use 6520's control register to set up  
STA $DF01      ;access to data direction registers  
STA $DF03
```

```
LDA #$C0      ;Set 10 LED lines to act as outputs,  
STA $DF00      ;and 3 top slide switches as inputs  
LDA #$FF  
STA $DF02
```

```
LDA MODE      ;Use control registers to set up  
STA $DF01      ;access to peripheral interface lines,  
STA $DF03      ;and Interrupt mode. See explanation of  
                ;MODE in next section.
```

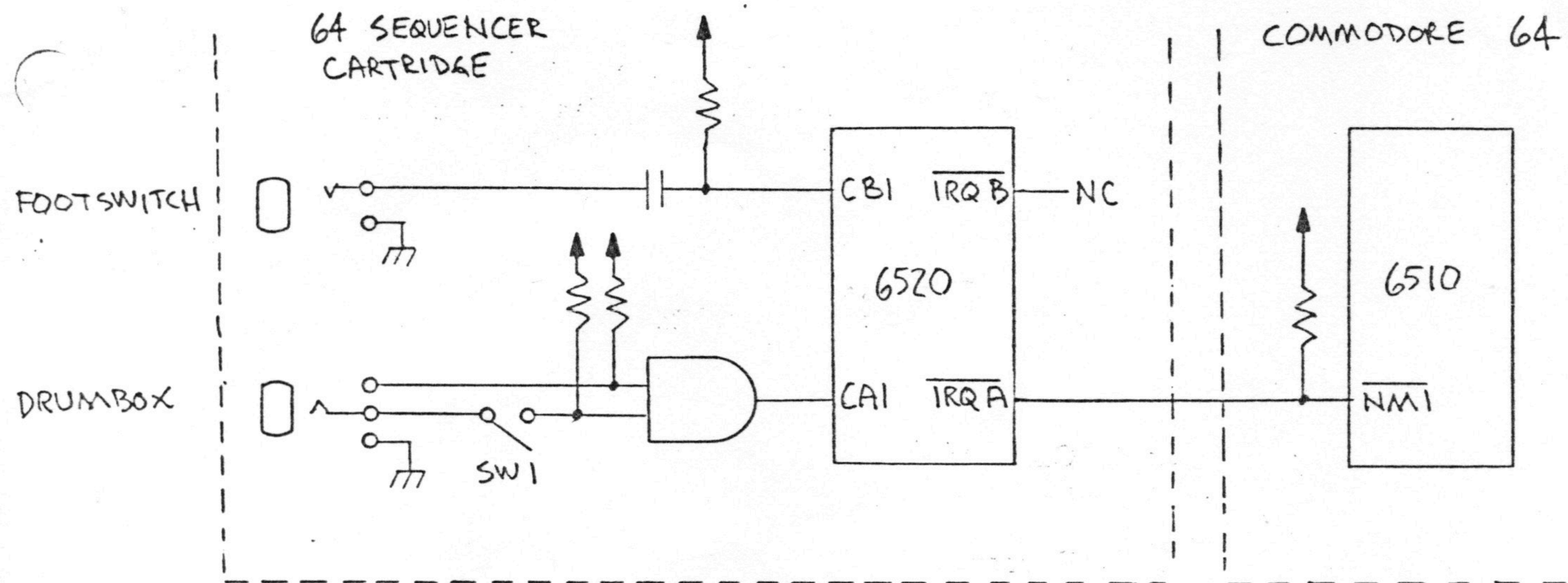


Fig. 1

To read Drum Input:

The drumbox jack on the cartridge is a stereo jack to accomodate all kinds of drumboxes. The switch (SW1 in Fig. 1) is the bottom of the four slide switches. It is open when in the left position and closed in the right. Most drumboxes (including the SCI Model 400) use a mono cable. For these systems the switch must be open (moved to the left) otherwise the bottom input of the AND gate will be grounded, inhibiting the AND gate. Drumboxes using stereo cables must have the switch closed.

An input from a drumbox can be detected in two ways:

(1) Direct interrupt (NMI) of the processor. For this, MODE=#07 in the initialization routine. A positive transition of the drumbox will cause a NMI. (If MODE=#05, a negative transition will cause a NMI).

(2) Poll the 6520 periodically to check if a drum pulse has occured. For this, MODE=#06. A positive transition will cause an interrupt flag bit to be set within the 6520. This is bit 7 of the control register, \$DF01. (If MODE=04, a negative transition will cause bit 7 to be set). Bit 7 is reset by a read of the peripheral interface register, \$DF00.

Example:

LDA \$DF01	;Read control register
BPL EXIT	;Branch if no drum click yet
	;Falls thru if interrupted by drum click
LDA \$DF00	;Clear the interrupt flag bit

To read Footswitch Input:

The footswitch can only be read by polling the other control register, \$DF03. Bit 7 is set on a footswitch transition (ground the tip), and reset by a read of the other peripheral interface register, \$DF02.

Example:

LDA \$DF03	;Read control register
BPL EXIT	;Branch if no footswitch press
LDA \$DF02	;Clear interrupt flag

--To turn on or off LEDs:

There is one bit assigned to each of the 10 LEDs. A "1" bit turns the LED on, and a "0" turns it off. The LED-controlling bits are the 8 bits written to \$DF02, and the top two bits when writing to \$DF00.

Example: Turn on all LEDs:

```
LDA #$C0  
STA $DF00      ;Turns on 2 LEDs  
LDA #$FF  
STA $DF02      ;Turns on 8 LEDs
```

LED assignments:

\$DF00:

Bit 7 - Cassette
Bit 6 - #4

\$DF02:

Bit 7 - Record
Bit 6 - Playback
Bit 5 - Overdub
Bit 4 - #3
Bit 3 - #1
Bit 2 - #2
Bit 1 - Song
Bit 0 - Sequence

LED physical layout on cartridge:


Sequence	#1
Song	#2
Record	#3
Playback	#4
Overdub	Cassette

-To read the 3 top slide switches:

Read the peripheral interface register, \$DF00. Bits 3,4, and 5 are the statuses of the switches, bit 3 being the top switch. A "1" bit means the switch is in the left position.

Sequential Circuits, Inc.
3051 North First Street
San Jose, CA 95134-2093
U.S.A.
408/946-5240
TELEX: 364412 INTR 706

Sequential Circuits, Inc.
Nijverheidsweg 11c
3641 RP Mijdrecht
Netherlands
02979-6211
TELEX: 12721 SQNTL NL



MODEL 64 OPERATION NOTE

The following information relates to software release "SEQ64_0_2."

Assigning Sequencer Tracks to MIDI

The "MO" Mode Select option which is discussed on pages 34 and 35 of the operation manual (CM64A) is no longer included.

Instead, there is now a simple way to assign a track to any MIDI channel you like:

In sequence-edit mode, select a track then press "C" on the Commodore 64.

A prompt will ask for the desired "Channel #?".

(If you change your mind, you can escape now by pressing RETURN.)

Enter desired channel number. (Press return.)

The sequencer always receives data in all channels. The channel select option allows you to specify in which channel the data is transmitted. If a track's channel is not edited, then the data is transmitted in the same channel in which it was received.

(over)

CN64-2 page 1
July, 1984

Autocorrect

Rather than delete note beginnings, this autocorrect version adjusts note beginnings so that the true, recorded note duration is retained.

Model 900

Before entering the Model 900 Expansion software, all sequences and songs must be saved to tape or diskette. Otherwise they will be lost.

When the Model 900 Expansion software is used, it will not be possible to exit back to the Model 64 main program by pressing "X". Instead, switch power off then back on.

64 SEQUENCER

64/47 Original software is V0104.

64/65 Software version SEQ64_0_1 is required for proper operation with Yamaha DX-9.

Schematic Current schematic is on reverse.

MODEL 64 OPERATION NOTE

The following information relates to software release "SEQ64_0_2."

Assigning Sequencer Tracks to MIDI

The "MO" Mode Select option which is discussed on pages 34 and 35 of the operation manual (CM64A) is no longer included.

Instead, there is now a simple way to assign a track to any MIDI channel you like:

In sequence-edit mode, select a track then press "C" on the Commodore 64.

A prompt will ask for the desired "Channel #?".

(If you change your mind, you can escape now by pressing RETURN.)

Enter desired channel number. (Press return.)

If this channel option is not used, the sequencer defaults to Omni Mode (Receive all Channels, Transmit Channel 1). For Poly mode, you would set all tracks to the same MIDI channel. For Mono Mode, set each track to a different channel.

(over)

CN64-2 * page 1
July, 1984

Autocorrect

Rather than delete note beginnings, this autocorrect version adjusts note beginnings so that the true, recorded note duration is retained.

Cassette/Disk Transfer

This version allows you to switch the configuration to allow loading from tape and saving to disk, or vice-versa.

Model 900

Before entering the Model 900 Expansion software, all sequences and songs must be saved to tape or diskette. Otherwise they will be lost.

When the Model 900 Expansion software is used, it will not be possible to exit back to the Model 64 main program by pressing "X". Instead, switch power off then back on.

