MD-1 card D - Pedal MIDI converter

This page last revised June 12, 2007

The hardware of all four MD-1 cards is identical, but they have slightly different software. On a typical organ, the Pedal interface card (card D) has the basic software, version 1.1. Click here to go to the MD-1 hardware page that describes the actual hardware of the board.

The standard software used on MD-1 card D is listed below, but note that any of the other versions would work just as well since they contain enhancements that are simply not needed or used in the pedal division. (Note: my own organ uses a different software version - see the bottom of this page.)

```
* MD-1 Keyboard scanner program
* Copyright (C) 2002, 2006 by Peter A. Stark

* This version (a) uses the DIP switch to set the lowest note
*              (b) does not allow changing channel or program numbers
*                  from the keyboard

* Version 1.1 10/24/2006 - mod to reset COP timer
* Version 1.0 3/13/2005 - minor edits
* Version 0.5 7/22/2002 - original

* I/O bits:
*   Inputs
* PA0: Serial input from shift register
* PC7-0: 1st note DIP switch
* PD0: Serial RS232 input
* PE5-2: Channel DIP switch
* PE0: Jumper to ground for Buffalo, but usable elsewhere too
*   Outputs
* PA7: H=RS232 (default), L=MIDI out
* PB0: LOAD/SHIFT' selection line
* PB1: Shift register clock line
* PD1: Serial out to RS232 or MIDI


*************************
*   GENERAL SYSTEM EQUATES
*************************

RAM     EQU   $0000     START OF RAM
REG     EQU   $1000     START OF REGISTERS
EEPROM  EQU   $B600     START OF EEPROM
ENDEEP  EQU   $B7FF     END OF EEPROM
PORTA   EQU   $1000     PORT A DATA & TIMER - B7=BI, B6-3=O,B2-1=I
PACTL   EQU   $1026     PORT A CONTROL
PORTB   EQU   $1004     PORT B DATA - OUTPUT ONLY
PORTC   EQU   $1003     PORT C DATA - BI
DDRC    EQU   $1005     PORT C DIRECTION
PORTD   EQU   $1008     PORT D DATA - 6 BITS BI & SCI/SPI
DDRD    EQU   $1009     PORT D DIRECTION
```

```
PORTE   EQU   $100A     PORT E - INPUT ONLY & A/D
BAUD    EQU   $102B     SCI BAUD REG
SCCR1   EQU   $102C     SCI CONTROL 1 REG
SCCR2   EQU   $102D     SCI CONTROL 2 REG
SCSR    EQU   $102E     SCI STATUS REG
SCDAT   EQU   $102F     SCI DATA REG
COPRST  EQU   $103A

*****************
*   RAM LOCATIONS
*****************

        ORG   RAM

CHANNL  RMB   1         CHANNEL NUMBER
NOTE    RMB   1         NOTE NUMBER
LOWNOT  RMB   1         FIRST (LOWEST) NOTE
DATA    RMB   64        KEYBOARD MEMORY
*                       bit 7 = 1 if now playing, =0 if silent
*                       bit 0: latest key status, 1=down
*                       bits 1-6: previous 6 key statuses

STACK   EQU   $00FF

****************
* Start and Initialize ports
****************

        ORG   EEPROM

COLDST  LDS   #STACK
        CLR   PORTB     SHIFT, NOT LOAD & HIGH CLOCK
        CLR   DDRC      PORT C IS ALL INPUT
        CLR   DDRD      PORT D ALL INPUT (EXC SERIAL)
        CLR   PORTE     PORT E IS ALL INPUT

**********
*   Initialize the SCI serial port
*        use $30 for 9600 baud during testing
*        use $20 for 31.25K baud for final version
**********
        LDX   #REG      POINT TO REGISTERS
        BCLR  PORTA-$1000,X $80 PA7=L FOR MIDI
        BSET  PACTL-$1000,X $80 ... AND OUTPUT
        LDAA  #$20      SET 31.25K BAUD INTO...
        STAA  BAUD      BAUD REGISTER
        LDAA  #$00      SET 8X1, NO WAKEUP
        STAA  SCCR1
        LDAA  #$0C
        STAA  SCCR2     ENABLE

**********
*  WARMST - WARM START
**********
```

```
WARMST  LDAA  PORTD     GET CHANNEL NUMBER
        LSRA            SHIFT RIGHT INTO PLACE
        LSRA
        STAA  CHANNL

        LDAA  PORTC     GET LOWEST (FIRST) NOTE
        ANDA  #$7F      MAX IS 127
        STAA  LOWNOT

        LDAB  #64       ERASE THE DATA ARRAY
        LDX   #DATA
LOOP1   CLR   0,X
        INX
        DECB
        BNE   LOOP1

        LDAB  #127      ISSUE 128 NOTE OFFS
        STAB  NOTE      PUT NOTE NUMBER IN NOTE
LOOP2   BSR   NOTEOFF
        DEC   NOTE
        BPL   LOOP2

********
*** outer loop to scan entire keyboard once
********

OULOOP  LDAA  #$55
        STAA  COPRST    RESET COP TIMER
        LDAA  #$AA
        STAA  COPRST
        LDAA  #$01
        STAA  PORTB     LOAD SHIFT REGISTER
        CLR   PORTB     TURN OFF LOAD
        LDX   #DATA+64  POINT PAST DATA ARRAY
        LDAA  LOWNOT
        ADDA  #64
        STAA  NOTE      CURRENT NOTE BEING WORKED ON+1

********
*** inner loop to process one key
********

INLOOP  DEX             POINT TO NEXT ARRAY ELEMENT
        CPX   #DATA     FINISHED?
        BMI   OULOOP    YES, GO RELOAD SHIFT REG
        DEC   NOTE      CURRENT NOTE BEING WORKED ON
        BRSET 0,X $80 PLAYING BRANCH IF NOTE IS NOW PLAYING

**********
* if note is silent
**********

SILENT  LDAA  PORTA     CHECK IF KEY IS UP OR DOWN
        ANDA  #$01
```

```
        BNE   SILEUP    BRANCH IF SILENT AND KEY UP

* if note is not currently playing but key is down
* then immediately play the note

SILEDN  LDAA  0,X       GET THE BYTE
        LSLA            SHIFT IT LEFT
        ORA   #$81      SET BITS 7 AND 0
        STAA  0,X
        BSR   NOTEON    PLAY THE NOTE

SHIFTIT LDAB  #$02      SEND OUT SHIFT PULSE
        STAB  PORTB     TO SHIFT REG
        CLR   PORTB
        BRA   INLOOP    AND REPEAT FOR NEXT KEY

* if note is silent and key is up, then just store it

SILEUP  LDAA  0,X       GET THE BYTE
        LSLA            SHIFTIT LEFT
        ANDA  #$7F      CLEAR BIT 7
        STAA  0,X
        BRA   SHIFTIT   SHIFT REGISTER AND REPEAT

**********
* if note is playing
**********

PLAYING LDAA  PORTA     CHECK IF KEY IS UP OR DOWN
        ANDA  #$01
        BNE   PLAYUP    BRANCH IF PLAYING AND KEY UP

* if note is currently playing and key is down
* then just store it

PLAYDN  LDAA  0,X       GET THE BYTE
        LSLA            SHIFT IT LEFT
        ORA   #$81      SET BITS 7 AND 1
        STAA  0,X
        BRA   SHIFTIT   SHIFT REG AND REPEAT

* if note is playing and key is up, then debounce

PLAYUP  LDAA  0,X       GET THE BYTE
        LSLA            SHIFTIT LEFT
        BEQ   SHUTOFF   IF IT'S ALL 0, THEN DO NOTE OFF
        ORA   #$80      ELSE SET BIT 7
        STAA  0,X       STORE IT
        BRA   SHIFTIT   SHIFT REGISTER AND REPEAT
SHUTOFF STAA  0,X       STORE IT
        BSR   NOTEOFF
        BRA   SHIFTIT   SHIFT REGISTER AND REPEAT

********
```

```
*   NOTEOFF - OUTPUT ONE NOTE OFF MESSAGE
*     USING CHANNL, NOTE, VEL
********
NOTEOFF LDAA  #$80     NOTEOFF CODE
        ADDA  CHANNL
        BSR   OUTEEE    NOTEOFF + CHANNEL
        LDAA  NOTE
        BSR   OUTEEE    NOTE NUMBER
        CLRA
        BRA   OUTEEE    VELOCITY

********
*   NOTEON - OUTPUT ONE NOTE ON MESSAGE
*     USING CHANNL, NOTE, VEL
********
NOTEON  LDAA  #$90      NOTEON CODE
        ADDA  CHANNL
        BSR   OUTEEE    NOTEON + CHANNEL
        LDAA  NOTE
        BSR   OUTEEE    NOTE NUMBER
        LDAA  #$70
        BRA   OUTEEE    VELOCITY

********
*   OUTEEE - ROUTINE TO OUTPUT ONE CHARACTER THRU SERIAL PORT
********

OUTEEE  LDAB  SCSR      READ STATUS
        BITB  #$80
        BEQ   OUTEEE    LOOP UNTIL TDRE=1
        STAA  SCDAT     SEND CHARACTER
        RTS

COPRIT  FCC 'COPYRIGHT (c) 2002, 2006 BY '
        FCC 'PETER A. STARK'

        END
```

As of June 10, 2007, I plan to change this software to a new version 1.1PT. The code that follows is the beta version of this new code -- it may change if any bugs are discovered. The new software will let this board output MIDI messages on two channels.

My Schober organ has 48 stop tabs plus 11 combination buttons (ten plus a general cancel) mounted below the accomp keyboard, which all connect to MD-1 board C, for a total of 59 used inputs. That MD-1 board has room for 64 inputs, so there is a total of 5 left over unused. But the Virginia WurtliTzer organ for Hauptwerk 2 has many more stops, pistons, toy counter, and other controls than that, so I need more inputs.

Each MD-1 board has room for 64 inputs, but the pedal board only uses 25 of those for the Schober Theater console, or 32 for a full AGO console. The other 32 could be used for stops etc., but they need to output a different channel number so they don't play notes.

The new software thus supports two channels: One channel for the lowest 32 inputs (normally used for pedals, and selected by the 4-bit DIP switch on the board, as usual), and a second channel for the top 32 inputs (used for other inputs, and selected by the least significant four bits

of the 8-bit DIP switch on the board). The pedal outputs will start at note 36 (hex 24) as is normal; the combo outputs will start at note 0, so they can overlap with other control signals on the same channel.

At this time, I have an extra set of 15 buttons mounted below the solo keyboard, as shown in this view.

There may be more in the future as well, such as a small box with indicators and buttons on the cheekblock. I am planning ahead by changing the software. Since each set of 32 inputs has its own 34-pin ribbon cable and connector, future additions will thus be possible without any other software changes.

The new code follows:

```
* MD-1 Keyboard scanner program
* Special version PT for the pedal and comb. buttons
* Copyright (C) 2002, 2007 by Peter A. Stark

* This version (a) does NOT use the DIP switch to set the lowest note -
*                  defaults to low note = 36 (two oct below middle C)
*                  for the pedal channel only
*               (b) does not allow changing channel or program numbers
*                  from the keyboard
*               (c) Outputs on two different channels:
*                  (1) The bottom 32 notes (mainly pedal) output on the
*                      channel set by the DIP-4 switch
*                  (2) The top 32 notes (comb. buttons etc.) output on
*                      channel set by the low 4 bits of the DIP-8 switch
*                      They also output note numbers 0 through 31

* Version 1.1PT revised 6/12/2007 for note number change
* Version 1.1PT 3/29/2007 - mod for pedal/comb button use
* Version 1.1 10/24/2006 - mod to reset COP timer
* Version 1.0 3/13/2005 - minor edits
* Version 0.5 7/22/2002 - original

* I/O bits:
```

```
*   Inputs
* PA0: Serial input from shift register
* PC7-0: former 1st note DIP switch, now 2nd channel
* PD0: Serial RS232 input
* PE5-2: Channel DIP switch
* PE0: Jumper to ground for Buffalo, but usable elsewhere too
*   Outputs
* PA7: H=RS232 (default), L=MIDI out
* PB0: LOAD/SHIFT' selection line
* PB1: Shift register clock line
* PD1: Serial out to RS232 or MIDI


*************************
*   GENERAL SYSTEM EQUATES
*************************

RAM     EQU   $0000     START OF RAM
REG     EQU   $1000     START OF REGISTERS
EEPROM  EQU   $B600     START OF EEPROM
ENDEEP  EQU   $B7FF     END OF EEPROM
PORTA   EQU   $1000     PORT A DATA & TIMER - B7=BI, B6-3=O,B2-1=I
PACTL   EQU   $1026     PORT A CONTROL
PORTB   EQU   $1004     PORT B DATA - OUTPUT ONLY
PORTC   EQU   $1003     PORT C DATA - BI
DDRC    EQU   $1005     PORT C DIRECTION
PORTD   EQU   $1008     PORT D DATA - 6 BITS BI & SCI/SPI
DDRD    EQU   $1009     PORT D DIRECTION
PORTE   EQU   $100A     PORT E - INPUT ONLY & A/D
BAUD    EQU   $102B     SCI BAUD REG
SCCR1   EQU   $102C     SCI CONTROL 1 REG
SCCR2   EQU   $102D     SCI CONTROL 2 REG
SCSR    EQU   $102E     SCI STATUS REG
SCDAT   EQU   $102F     SCI DATA REG
COPRST  EQU   $103A
LOWNOT  EQU   36        LOWEST NOTE ON KBD

*****************
*   RAM LOCATIONS
*****************

        ORG   RAM

PCHANL  RMB   1         PEDAL CHANNEL NUMBER
TCHANL  RMB   1         COMB BUTTON CHANNEL NUMBER
CHANNL  RMB   1         CURRENT CHANNEL TO USE
NOTE    RMB   1         KEYBOARD NOTE NUMBER
DATA    RMB   64        KEYBOARD MEMORY
*                       bit 7 = 1 if now playing, =0 if silent
*                       bit 0: latest key status, 1=down
*                       bits 1-6: previous 6 key statuses

STACK   EQU   $00FF
```

```
****************
* Start and Initialize ports
****************

        ORG   EEPROM

COLDST  LDS   #STACK
        CLR   PORTB     SHIFT, NOT LOAD & HIGH CLOCK
        CLR   DDRC      PORT C IS ALL INPUT
        CLR   DDRD      PORT D ALL INPUT (EXC SERIAL)
        CLR   PORTE     PORT E IS ALL INPUT

**********
*   Initialize the SCI serial port
*        use $30 for 9600 baud during testing
*        use $20 for 31.25K baud for final version
**********
        LDX   #REG      POINT TO REGISTERS
        BCLR  PORTA-$1000,X $80 PA7=L FOR MIDI
        BSET  PACTL-$1000,X $80 ... AND OUTPUT
        LDAA  #$20      SET 31.25K BAUD INTO...
        STAA  BAUD      BAUD REGISTER
        LDAA  #$00      SET 8X1, NO WAKEUP
        STAA  SCCR1
        LDAA  #$0C
        STAA  SCCR2     ENABLE

**********
*  WARMST - WARM START
**********

WARMST  LDAA  PORTD     GET PEDAL CHANNEL NUMBER
        LSRA            SHIFT RIGHT INTO PLACE
        LSRA
        STAA  PCHANL

        LDAA  PORTC     GET COMBO CHANNEL NO
        ANDA  #$0F      USE LOW FOUR BITS
        STAA  TCHANL

        LDAB  #64       ERASE THE DATA ARRAY
        LDX   #DATA
LOOP1   CLR   0,X
        INX
        DECB
        BNE   LOOP1
        LDAA  PCHANL
        JSR   ALLOFF    ALL NOTES OFF FOR PEDAL
        LDAA  TCHANL
        JSR   ALLOFF    ALL NOTES OFF FOR COMBO

********
*** outer loop to scan entire keyboard once
********
```

```
OULOOP  LDAA  #$55
        STAA  COPRST    RESET COP TIMER
        LDAA  #$AA
        STAA  COPRST
        LDAA  #$01
        STAA  PORTB     LOAD SHIFT REGISTER
        CLR   PORTB     TURN OFF LOAD
        LDX   #DATA+64  POINT PAST DATA ARRAY
        LDAA  #32       DOING COMBO BUTTONS FIRST
        STAA  NOTE      CURRENT NOTE BEING WORKED ON+1
        LDAA  TCHANL    TOP NOTES ARE COMBO BUTTONS
        STAA  CHANNL

********
*** inner loop to process one key
********

INLOOP  DEX             POINT TO NEXT ARRAY ELEMENT
        CPX   #DATA     FINISHED?
        BMI   OULOOP    YES, GO RELOAD SHIFT REG
        DEC   NOTE      CURRENT NOTE BEING WORKED ON
        BPL   CONTNU    CONTINUE UNLESS NEGATIVE
        LDAA  PCHANL    IF -1 SWITCH TO PEDAL CHANNEL
        STAA  CHANNL
        LDAA  #LOWNOT
        ADDA  #31       SWITCH TO PEDAL NOTES
        STAA  NOTE
CONTNU  BRSET 0,X $80 PLAYING BRANCH IF NOTE IS NOW PLAYING

**********
* if note is silent
**********

SILENT  LDAA  PORTA     CHECK IF KEY IS UP OR DOWN
        ANDA  #$01
        BNE   SILEUP    BRANCH IF SILENT AND KEY UP

* if note is not currently playing but key is down
* then immediately play the note

SILEDN  LDAA  0,X       GET THE BYTE
        LSLA            SHIFT IT LEFT
        ORA   #$81      SET BITS 7 AND 0
        STAA  0,X
        BSR   NOTEON    PLAY THE NOTE

SHIFTIT LDAB  #$02      SEND OUT SHIFT PULSE
        STAB  PORTB     TO SHIFT REG
        CLR   PORTB
        BRA   INLOOP    AND REPEAT FOR NEXT KEY

* if note is silent and key is up, then just store it

SILEUP  LDAA  0,X       GET THE BYTE
```

```
        LSLA            SHIFTIT LEFT
        ANDA  #$7F      CLEAR BIT 7
        STAA  0,X
        BRA   SHIFTIT   SHIFT REGISTER AND REPEAT

**********
* if note is playing
**********

PLAYING LDAA  PORTA     CHECK IF KEY IS UP OR DOWN
        ANDA  #$01
        BNE   PLAYUP    BRANCH IF PLAYING AND KEY UP

* if note is currently playing and key is down
* then just store it

PLAYDN  LDAA  0,X       GET THE BYTE
        LSLA            SHIFT IT LEFT
        ORA   #$81      SET BITS 7 AND 1
        STAA  0,X
        BRA   SHIFTIT   SHIFT REG AND REPEAT

* if note is playing and key is up, then debounce

PLAYUP  LDAA  0,X       GET THE BYTE
        LSLA            SHIFTIT LEFT
        BEQ   SHUTOFF   IF IT'S ALL 0, THEN DO NOTE OFF
        ORA   #$80      ELSE SET BIT 7
        STAA  0,X       STORE IT
        BRA   SHIFTIT   SHIFT REGISTER AND REPEAT
SHUTOFF STAA  0,X       STORE IT
        BSR   NOTEOFF
        BRA   SHIFTIT   SHIFT REGISTER AND REPEAT

********
*   NOTEOFF - OUTPUT ONE NOTE OFF MESSAGE
*     USING CHANNL, NOTE, VEL
********
NOTEOFF LDAA  #$80      NOTEOFF CODE
        ADDA  CHANNL
        BSR   OUTEEE    NOTEOFF + CHANNEL
        LDAA  NOTE
        BSR   OUTEEE    NOTE NUMBER
        CLRA
        BRA   OUTEEE    VELOCITY = 0

********
*   NOTEON - OUTPUT ONE NOTE ON MESSAGE
*     USING CHANNL, NOTE, VEL
********
NOTEON  LDAA  #$90      NOTEON CODE
        ADDA  CHANNL
        BSR   OUTEEE    NOTEON + CHANNEL
        LDAA  NOTE
```

```
        BSR   OUTEEE    NOTE NUMBER
        LDAA  #$70      DEFAULT
        BRA   OUTEEE    VELOCITY

********
*    ALLOFF - ISSUE AN ALL NOTES OFF MSG - ENTER WITH CHANNEL IN A
********
ALLOFF  ADDA  #$B0      CONTROLLER CODE
        BSR   OUTEEE    CTRLR + CHANNEL
        LDAA  #123
        BSR   OUTEEE    ALL NOTE OFF CODE
        CLRA
        BRA   OUTEEE    LAST BYTE = 0

********
*   OUTEEE - ROUTINE TO OUTPUT ONE CHARACTER THRU SERIAL PORT
********

OUTEEE  LDAB  SCSR      READ STATUS
        BITB  #$80
        BEQ   OUTEEE    LOOP UNTIL TDRE=1
        STAA  SCDAT     SEND CHARACTER
        RTS

COPRIT  FCC 'COPYRIGHT (c) 2002-2007 BY '
        FCC 'PETER A. STARK'

        END
```