Zoom MS70-CDR MIDI specification.
by C. Feddema aka sixeight (www.vguitarforums.com)

After doing the MIDI implementation for the Zoom G3, I bought a MS70-CDR, which also supports midi.

I wanted to be able to read patchnames, switch effects on and off and also be able to read the effects type and state. And i have managed to "crack" the code of the Zoom MS70CDR.

More information about the VController can be found here:
http://www.vguitarforums.com/smf/index.php?topic=15154.msg130947#msg130947

To understand this guide you must have a good understanding of MIDI, binary and hexadecimal numbers

This is by no means a complete guide, but may serve as a starting point to control the MS70-CDR from an Arduino based foot controller. Using a similar approach as i have used with the G3, I found the following things.

Program change:
the MS70-CDR responds to Program change messages.

Control change:
AFAIK the MS70-CDR does not respond to CC messages

System exclusive:

MIDI consists of command bytes and data bytes. All command bytes have a value >=128. This means the 8th bit is 1. All data bytes have a value <=127. This means the 8th byte is 0. A sysex command consists of two command bytes (F0 and F7) and all the other bytes are data bits.

ZOOM MS70cdr sysex messages have the following structure (all numbers are hexadecimal for sysex messages in this document):

F0 52 00 61 33 F7

F0: start of sysex message
52: Zoom manufacturing ID
00: Device ID
61: Model number of the MS70cdr
33: A command
F7: End of sysex message

1) To detect the MS70-CDR I send a MIDI identity request message: F0 7E 00 06 01 F7. This is a message that all MIDI devices should respond to. The Zoom responds with F0 7E **00** 06 02 **52 61** 00   00 00 32 2E 31 30 F7.

i have highlighted:

00 The device ID
52 The manufacturing ID
61 The model number for the MS70CDR

2) Sending F0 52 00 61 50 F7 will put the MS70-CDR into "editor mode". This means that the MS70-CDR will send sysex messages when changing effects and parameters on the unit itself..

3) Sending F0 52 00 61 51 F7   will switch "editor mode" off.

4) If the editor sends F0 52 00 61 33 F7, the MS70-CDR responds with the current Bank number (CC00 and CC32) and the current Program number (PC)

5) If the editor sends F0 52 00 61 29 F7, the MS70-CDR responds with the current patch in 146 bytes with command number 28:

```
00   F0 52 00 61 28 00 21 01    00 4C 00 60 00 00 19 40   | R a( !   L `      @|
10   46 06 00 00 00 00 00 00    00 00 20 01 00 00 4C 00   |F                    L |
20   60 00 19 40 46 00 06 00    00 00 00 00 00 00 00 21   |`   @F               !|
30   01 00 4C 00 60 00 00 19    40 46 06 00 00 00 00 00   |   L `    @F          |
40   00 00 00 21 01 00 00 4C    00 60 00 19 40 00 46 06   |    !   L `   @ F |
50   00 00 00 00 00 00 00 00    21 01 00 4C 00 00 60 00   |           !   L  `|
60   19 40 46 06 00 00 00 00    00 00 00 00 21 00 01 00   | @F               !   |
70   4C 00 60 00 19 00 40 46    06 00 00 00 00 00 00 00   |L `    @F          |
80   00 00 18 0F 45 00 6D 70    74 79 20 20 20 00 20 20   |      E mpty          |
90   00 F7
```

And certain positions there are extra bytes. I will call these "overflow" bytes. These bytes are bold above. It took me a long time to figure out what these were for. But sysex data bytes can only have value 0-127. This gives 7 bits. But the Zoom wants to send bytes of eight bits. So it will put all the 8th bits of the following 7 bytes into one new data byte.

For example: say the Zoom wants to send the following 7 bytes:
10 00 00 **80** 00 00 **A0**
Sending 80 and A0 is not allowed by sysex MIDI convention.
The Zoom will   put the "extra" bits in a new byte:
Byte 4 and 7 are too large. This makes B00001001 = HEX 09. So the G5 will send:
09 10 00 00 **00** 00 00 **20**

80 HEX with the 8th bit 0 becomes 00
A0 HEX with the 8th bit 0 becomes 20

Patchname:
Byte 0x84 - 0x8F contain the patch name. with an overflow byte at position 0x85 and one at position 0x8D The bytes for the patch name are underlined and bold.

Effect type and state:
The are a number of bytes underlined above. These bytes together with the overflow

bytes contain the effect type and number and whether the effect is on or off. In the example above I highlighted 0x21 0x01 and 0x4C

It took a long time to make sense of this data. First I found a number of effect categories. These are stored in the last five bits of the last byte. So we have to split off the correct bits first by doing 0x4C & 0x1F = 0x1C

The effect categories used in the MS70-CDR are:
  0x02   (2): Zoom noise reduction (ZNR)
  0x04   (4): Eq effects
  0x0C (12): Chorus effects
  0x0E (14): Roto effects
  0x10 (16): Delay effects
  0x12 (18): Reverb effects

(The numbers between brackets are decimal)

Extensive searching of the other two bytes gave me the following "formula" to decipher the other two bytes and get a unique number for the effect type:

(byte1 & 0xF8) + (byte2 & 0x0F)

In byte1 the relevant overflow bit has to be included!!!
Bit 4 from byte1 is included, to differentiate between HD Hall and HD reverb.

Using this code you can search for the effect in the table at the end of this document.

6) It is also possible to read a specific patch by sending :
  F0 52 00 61 09 00 00 **00** F7.
The last 00 in this command is the patch number with goes from 00 to 63 (0 - 100 decimal)
The MS70-CDR responds with 156 bytes with command number 08.

```
00   F0 52 00 61 08 00 00 28   7A 00 40 41 01 00 04 00   | R a    (z @A      |
10   00 00 00 00 00 00 00 00   40 06 04 00 00 00 00 41   |          @        A|
20   01 00 00 04 00 00 00 00   00 00 00 00 00 40 06 00   |                @   |
30   00 00 20 00 41 01 00 04   00 00 00 00 00 00 00 00   |    A              |
40   00 00 02 00 00 00 00 00   41 01 00 00 04 00 00 00   |         A         |
50   00 00 00 00 00 00 00 00   00 00 10 00 00 41 01 00   |              A    |
60   04 00 00 00 00 00 00 00   00 00 01 00 00 00 00 00   |                   |
70   00 41 00 01 00 04 00 00   00 00 00 00 00 00 00 00   | A                 |
80   00 00 00 00 00 00 00 18   0F 50 00 69 6E 6B 46 20   |            P inkF  |
90   20 20 00 20 20 00 06 6F   2A 3A 03 F7               |          o*:   |
```

The structure is similar to the data of the current patch. The first highlighted 28 is the patch name. Overflow bytes have been highlighted again. Patchname and effect type/state bytes have been underlined.

7) Switch effect on/off:

Switch on effect 1: F0 52 00 61 31 00 00 01 (on) 00 F7,
Switch off effect 1: F0 52 00 61 31 00 00 00 (off) 00 F7

Switch on effect 2: F0 52 00 61 31 01 00 01 (on) 00 F7
Switch off effect 2: F0 52 00 61 31 01 00 00 (off) 00 F7

Switch on effect 3: F0 52 00 61 31 02 00 01 (on) 00 F7
Switch off effect 3: F0 52 00 61 31 02 00 00 (off) 00 F7

This only works for the first three effects. Effect 4 to 6 can not be switched using this command. In my VController I flip the relevant bit by changing it in the patch data. After that I send the full patch data back to the MS70CDR.

8) Setting the tempo does not seem to work using a short command. In my VController I change the tempo in the current patch data. After that I send the full patch data back to the MS70CDR. The tempo data is found in the following index position:

```
Tempo bit 4 - 8 is bit 1-5 from second tempo byte (index 131)
Tempo bit 3 is bit 3 of the overflow byte! (index 125)
Tempo bit 2 and 1 are first byte bit 2 and 3 (index 130)
```

Table: effect types - numbers are decimal:

Here is the table where the effect type can be searched:
The first number is the effect category
The second number is the effect type, using the previously mentioned "formula"

There seem to be many unused numbers. it could be that the effect numbering system is consistent on all Zoom MultiStomp devices, but as I only have the MS70-CDR I can not confirm this.

```
 0, 0,    "---"
12, 195, "CoronaCho"
12, 224, "Chorus"
12, 33,  "VintageCE"
12, 227, "ANA 234Cho"
12, 3,   "CE-Cho5"
12, 35,  "Clone Chorus"
12, 161, "Super Cho"
12, 163, "Mirage Chorus"
12, 65,  "Stereo Cho"
12, 4,   "Corona Tri"
12, 97,  "Ensemble""
12, 131, "Silky Chorus"
12, 1,   "Detune"
12, 240, "Bass Chorus"
12, 17,  "Bass Detune"
12, 113, "Bass Ensemble"
```

```
12, 193, "Flanger"
12, 209, "Bass Flanger"
12, 2,    "Vibrato"
12, 96,   "Phaser"""
12, 128, "The Vibe"
12, 66,   "Pitch SHFT"
12, 130, "HPS"
12, 114, "Bass Pitch"
12, 32,   "Duo-Trem"
14, 160, "Auto-Pan"
14, 192, "Roto Closet"
16, 193, "Duo Digital Dly"""
16, 16,   "Delay"
16, 129, "Stomp Dly"
16, 33,   "Stereo Delay"
16, 225, "Carbon Delay"
16, 96,   "Analog DLY"
16, 32,   "Tape Echo"
16, 161, "Tape Echo3"
16, 2,    "Drive Echo"
16, 34,   "Slapback Delay"""
16, 66,   "Smooth Delay"
16, 160, "Multi Tap Delay"
16, 128, "Reverse Delay"
16, 98,   "Lo-Fi Delay"
16, 130, "Slow Attack Dly"
16, 192, "Dyna Delay"
16, 162, "Tremolo Delay"
16, 64,   "ModDelay"
16, 97,   "Trigger Hold Dly"""
16, 194, "Filter PP-Delay"
16, 224, "Filter Dly"
16, 65,   "Phase Delay"
16, 226, "Auto Pan Delay"
16, 1,    "Pitch Delay"
16, 3,    "ICE Delay"
16, 80,   "ModDelay2"
18, 24,   "HD Reverb"
18, 128, "Spring"""
18, 129, "Spring63"
18, 33,   "Plate"
18, 64,   "Room"
18, 96,   "Tiled Room"
18, 161, "Chamber"
18, 193, "LoFi Reverb"
18, 32,   "Hall"
18, 16,   "HD Hall"
18, 225, "Church"""
18, 160, "Arena Reverb"
18, 2,    "Cave"
```

```
18, 34,   "Ambience"
18, 224, "AIR"
18, 192, "Early Reflection"
18, 66,   "Gate Reverb"
18, 98,   "Revere Reverb"
18, 97,   "Slap Back Reverb"
18, 130, "Echo""
18, 65,   "Mod Reverb"
18, 162, "Tremelo Reverb"
18, 194, "Holy Flerb"
18, 226, "Dynamic Reverb"
18, 3,     "Shimmer Reverb"
18, 35,   "Particle Reverb"
18, 67,   "Space Hole"
18, 99,   "Mangled Space"
18, 131, "Dual Reverb"
2, 160,   "ZNR"
4, 193,   "Stereo Gtr GEQ"
4, 2,       "Stereo Bass GEQ"
4, 16,     "LineSel"
```