SAC-2K Midi-Implementation V1.06

Date: 14th Apr 2001, Update in Progress 2nd October 2002

Firmware-Version overview:
Firmware < 1.50: Interface V 0.00
Firmware < 2.00: Interface V 1.00
Firmware >= 2.00: Interface V 1.02
Firmware >= 2.21: Interface V 1.03
Firmware >= 2.24: Interface V 1.04
Firmware >= 2.34: Interface V 1.05
Firmware >= 2.40: Interface V 1.06

Color-Legend:
Blue: Corrections to initial Release
Red: Future Extensions
Orange: Extensions to be implemented for V1.02
Green: New Extensions already implemented

SYSTEM EXCLUSIVE

Identity Request:
F0h 7Eh dv 06h 01h F7h

Reply from SAC-2K:
F0h 7Eh dv 06h 02h 00h 01h 36h 2Ah 00h 00h 00h vv vv vv vv F7h

Note: *dv* = system channel (0..15 or 7Fh=respond always)
vv vv vv vv = Version number (ASCII character, i.e. '1.00')

System Exclusive format:

byte 1:	*F0h*	= Sysex Start
byte 2..4:	*00h 01h 36h*	= Radikal Technologies Manufacturers ID
byte 5:	*2Ah*	= SAC-2K Model ID
byte 6:	*dv*	= System channel (Depending on Mode, use 7Fh in Request)
byte 7:	*cm*	= Command: 44h=Dump, 52h=Request Data
byte 8..10:	*ah am al*	= Adress
byte 11...	*dt dt dt ...*	= Data
byte xx:	*F7h*	

Command-Overview:

41h	'A'	Register Application	
43h	'C'	Define LC Character	
44h	'D'	Dump	
47h	'G'	Request System Dump	
49h	'I'	Instrument-Editor	
52h	'R'	Request Data	(V0.00 or V1.04+)
53h	'S'	System Dump	
59h	'Y'	Switch Application	(Used by USB-Driver)
79h	'y'	System-Key	(V1.03, Used by USB-Driver)

Request System Parameter:
F0h 00h 01h 36h 2Ah dv 52h ah am al nb F7h

nb = Number of Bytes to query

Global System Configuration:
F0h 00h 01h 36h 2Ah dv 44h 00h 00h 00h md [ch] [tr] [mo] F7h

Note: *md* = system mode
bit 0..1: time display mode
0 = mtc mode
1 = reserved
2 = ascii mode
3 = jlc3000 compatible mode
bit 2: fader motor off (silence mode)
0 = motor on
1 = silence mode (motor off)
bit 3: fader touch mode
0 = hard touch (only hardware controlled)
1 = soft touch (also software controlled)
bit 4: fader resolution
0 = 7 bit fader (no lsb)
1 = 8 bit fader (send lsb)
bit 5..6: global mode (superseded by *mo* parameter)
These bits define bits 0 and 1 of the current mode the SAC-2k is working. bit 2 is encoded in the optional *ch* byte.
See the description of the preferred *mo* byte below for encoding.

ch = system channel, global mode MSB (sending is optional)
bit 0..3: system channel
bit 6: bit 2 of global mode
tr = touch response (sending is optional)
mo = global mode (V1.00)
0 = Generic Active mode, SAC-2x memory active
1 = (active) EMAGIC Logic Audio object bang mode
2 = Generic Slave mode, SAC-2x memory inactive
3 = (slave) Digidesign Protools (HUI) compatible mode
4 = (active) pulsar mode
5 = (active) Steinberg Reason mode
6 = (slave) EMAGIC Logic Control (V1.05)
7 = (slave) Steinberg Houston (V1.05)
8 = (slave) Mackie Control (V1.05)
7fh = no change

Unique Device ID (V1.04):
F0h 00h 01h 36h 2Ah dv 44h 01h 00h 00h id F7h

Note: *id* = Unique Device ID set by the User

Motor Parameters (V1.04):
F0h 00h 01h 36h 2Ah dv 44h 02h 00h 00h Tmo Scl MinH MinL MaxH MaxL F7h

Note: *Tmo* = Timeout Value 0..31
Scl = Scale 0..15
MinH, MinL = Minimum Speed Bit8 in MinH, Bit7..0 in MinL
MaxH, MaxL = Maximum Speed Bit8 in MaxH, Bit7..0 in MaxL

Input Parameters (V1.04):
F0h 00h 01h 36h 2Ah dv 44h 03h 00h 00h DtH DtL Dmax JtH JtL Jmax FtH FtL F7h

Note: DtH, DtL = Dial Timeout in ms Bit8 in DtH, Bit7..0 in DtL
Dmax = Maximum Delta sent from Dials
JtH, JtL = Jog-Wheel Timeout in ms Bit8 in JtH, Bit7..0 in JtL
Jmax = Maximum Delta sent from JogWheel
FtH, FtL = Fader Timeout in ms Bit8 in FtH, Bit7..0 in FtL

Display Parameters (V1.04):
F0h 00h 01h 36h 2Ah dv 44h 04h 00h 00h Vtmo Blink Fill Help F7h

Note: Vtmo = Value Timeout 0..127 in 32ms Increments
Blink = Blink-Rate
Fill = Show filled Encoders
Help = Display Online-Help

Use global mode 2 (slave) because the active mode not finished until now. The descriptions of the control changes below belongs to this mode! To set this mode send (i.e.): *F0h 00h 01h 36h 2Ah 0Fh 44h 00h 00h 00h 40h F7h*

Send Tracknames (used only in active and logic mode):
F0h 00h 01h 36h 2Ah dv 44h 1xh tt 00h dt... F7h

Note: x = track categorie number
0 = Audio tracks
1 = Midi tracks
2 = Input tracks
3 = Instrument tracks
4 = Bus tracks
5 = user defined track groups
tt = track number (0..31 = track 1..32)
dt... = text (ASCII character, 10 byte)

Send Instrumenteditor Number **(Use Select Instrument-Editor version beyond!)**:
F0h 00h 01h 36h 2Ah dv 44h 20h 00h tt n1 n2... F7h

Note: tt = first track number (0..15 = track 1..16)
no = syntheditor number (possibility to send one or multiple numbers)

Send Display Text:
F0h 00h 01h 36h 2Ah dv 44h 70h dn cp dt... F7h

Note: *dn* = display number (0..2)
cp = cursor position from 00 through 27h for the top line,
40h through 67h for the bottom line.
dt... = text (ASCII character)
If the 40th character is written in the first line the cursor is automatically repositioned into the second line.
You may also send these special characters (V1.00):
0Dh (CR) Clears display and set Cursor into Home-Position
0Ah (NL) Position Cursor to home position in the second line

Switch Application (V1.02)
F0h 00h 01h 36h 2Ah dv 59h app F7h

Note: app = Application-Number (Digit)
This Message is sent, when the user depresses a Digit-Key with the System-Key held down. Since this Message is interpreted by the USB-Driver, Applications should not use it.

System-Function pressed (V1.03)
F0h 00h 01h 36h 2Ah dv 79h key F7h

Note: Key pressed along with System-Key.
Key=7Fh: System-Button down
Key=7Eh: System-Button up

Select Instrument Editor (V1.02):
F0h 00h 01h 36h 2Ah dv 49h ii pp mm id id id id F7h
ii = Editor-Index (0..31)
pp = Port-Index. 0=System, 1=USB3, 2=USB4, 3=USB5
mm = MIDI Channel (>15 = Off)
id = 4-Byte Editor-Identifier:

'SPr5' Pro-Five
'SPr2' Pro 52
'SWv2' Wave 2.V
'NiB4' Native Instruments B4
'SMdE' Model E
'NP2k' NI Premium Library SH-2K
'NMMo' NI Many Mood
'NPTo' NI Premium Library Three Osc
'NPMe' NI Premium Library Me2SalEM
'Ex24' Emagic EXS 24 Extreme Sampler
'EEs1' Emagic ES 1 Synthesizer
'TMc1' TC-Works Mercury 1
'CEd8' Pulsar EDS8i
'CESy' Pulsar EZ Synth
'CInf' Pulsar Inferno
'CMSc' Pulsar MiniScope
'CBSy' Pulsar BlueSynth
'CVoc' Pulsar Vocoder
'Cknw' Pulsar U-Know 007
'PSbt' Reason Subtractor
'PDRx' Reason Dr. Rex
'PN19' Reason NN-19
'PRdr' Reason Redrum Mixer
'PVct' Pulsar Vectron
'COr2' Zarg Orion V2.0
'CDst' Zarg Dark Star

Define LC Character (V1.03):
F0h 00h 01h 36h 2Ah dv 43h di ch r0 r1 r2 r3 r4 r5 r6 r7 F7h
di = Display Number (0..2)
ch = Character Number 0..7

r0..r7 = Eight rows of Pixel-Definitions
Row-Order is from top to bottom
Bit 4 is leftmost, Bit 0 is rightmost

Note: If your Application requires special Characters to appear inside the Display, you can specify up to eigth special Characters per Display. These definable Characters have the Character-Codes 0x00..0x07 and can be addresses like standard ASCII Characters.
As these special Characters are internally used to create the Level-Meters, you can only use either.
Since these Characters are defined per Display, you can use special Characters in the third Display and VU-Meters in the first and second simultaneously.

Register Application (required to use V1.02):
F0h 00h 01h 36h 2Ah dv 41h Vend Vend Vend VerMaj VerMin SacIfVer State Id Id Id Id F7h
Vend Vend Vend = Official 3 Byte Vendor ID given by the MMA. If no Vendor ID is available, choose a three Alpha-Letter Pseudonym for your Company.
VerMaj VerMin Major in Minor Version Number. If you use only one Digit for the Minor Version Number, multiply this Value by 10.
So Version 2.5 will be encoded as: 02h 32h (32h = 50d)
SacIfVer SAC Interface Version encoded as two Digits (V0.00..V7.15). Prior Implementation are considered V0.xx, newer Implementations should use the Version stated at the Top of this Document (i.e. V1.02 = 12h).
State = Application State. 0=Shutdown, 1=Startup
Id Id Id Id = A four Byte Identifier for your Application. This should be constant through all Software-Revisions.
Applications directly supporting the SAC should register themselves with this message. Future Versions of the SAC Firmware may interpret the settings and fine-adjust its parameters depending on these informations.

The following specifications are preliminary informations and should not yet be used for implementations (currently they're are not even implemented!).

Inter-Device Communitaion (preliminary)
F0h 00h 01h 36h 2Ah dv 5Ah dd... F7h

Note: dd = Internal used Device-Data
This Message is used for Communication between the Main-Module and the Fader Expansion-Units and completely interpreted by the USB-Driver.

Request Complete System Dump (preliminary):
F0h 00h 01h 36h 2Ah dv 47h what F7h
What = a sequence of types (see below) requested or 0x2A ('*') to get all valid fields

Transfer System Dump (preliminary):
F0h 00h 01h 36h 2Ah dv 53h vv mo ie subpacket ... F7h
vv – Interface Version-Number. Current: 10h = V1.0
mo – requested mode or 7Fh to keep current mode
ie - Currently Selected Instrument-Editor.
7Fh=Keep current, 00h=No Editor, 1..32=Set Active Editor

Subpacket-Format:

Each subpacket contains a Type describing the kind of data encoded, a Width in Bytes of each Item, an Offset to the first Item transmitted and a Count of transmitted Items.
Each of these Counts are encoded as 6-Bit-Values stored in the low 6 Bit of a Byte. If Bit 6 is set, then the current Byte encodes the most significant 6 Bits and the next Byte(s) the less significant Bits. So an arbitrary Bit-Length is encodable and does not impact efficiency on low numbers.
The decoding can be implemented using a similiar Code-Fragment:

```
int Value;
byte MidiByte;
Value = 0;
do {
  MidiByte = GetMidiByte();
  Value = (Value * 64) + (MidiByte AND 0x3F);
  }
while((MidiByte AND 0x40) != 0);
```

<type> Width Offset Count data ...
Width of one field in Bytes (0 = Zero Terminated for Strings)
Offset describes at which offset the data starts (not necessary a real index)
Count specifies the number of fields transmitted.

Type-Descriptors:
Fader Values 'F':
0x46 0x01 Offset Count <Value> ...
Offset 0..32, 32 = Master-Fader
Active-Modes: All 33 Faders are supported
Slave-Modes: Only Values 0..7,32 are supported

Encoder Values 'E':
0x45 0x02 Offset Count <Value> <Mode> ...
Offset 0..11
Only available in Slave-Modes

Display-Contents 'D':
0x44 0x00 FirstPosition 0x01 <String>
First Position contains the Line-Number in the bottom 4 Bits and the Display-Number in Bits 4..7. So for SAC-2k this is the range 0x00..0x21
String is the Zero-Terminated String written to the given Display-Position. Any Newline (0x0A) jumps to the next Line of the current Display, a Tab (0x09) jumps to the next Display. A Carriage-Return (0x0D) clears the current Display.
Only available in Slave-Modes

Led-Encoding 'L':
0x4C 0x01 Offset Count <LedState>
LedState: This is a 4-Bit Blink-Pattern, these are some combinations:
0x00 Led stays off (LEDMODE_OFF)
0x03 Led Blinks (LEDMODE_BLINK)
0x06 Led Blinks with a different Phase (LEDMODE_BLINK2)
0x0C Led Blinks with another Phase (LEDMODE_BLINK3)
0x05 Led Blinks faster (LEDMODE_QBLINK)
0x02 Led Flashes (LEDMODE_FLASH)
0x07 Led Flashes inverted (LEDMODE_UNFLASH)
0x0F Led stays on (LEDMODE_ON)

The Led-Order is System-Dependant
Only available in Slave-Modes

Instrument Editor 'I':
0x49 0x06 ii 0x01 pp mm pg id id id id
ii = Editor-Index (0..31)
pp = Port-Index. 0=System, 1=USB3, 2=USB4, 3=USB5
mm = MIDI Channel (>15 = Off)
pg = Selected Page
id = 4-Byte Editor-Identifier
The Editor-Index must be stored for upcoming Page-Data (Type 0x69)

Instrument Editor Page Data 'i':
0x69 0x0C FirstPage Count 12xData
FirstPage 0..12
Count Number of Page-Parameters transferred
Data Always multiples of 12 Values per Page

Mixer Data 'M':
0x6D 0x18 FirstTrack Count 28xData
FirstTrack 0..32
Only available in Active-Modes
Data Bytes 0..23 contain the 24 Mixer-Values for each track
Byte 0: Mute (0..1)
Byte 1: Solo (0..1)
Byte 2: Volume (0..127)
Byte 3: Pan (0..127, Center at 64)
Byte 4: Gain High (0..127)
Byte 5: Send 1 (0..127)
Byte 6: Gain HighMid (0..127)
Byte 7: Send 2 (0..127)
Byte 8: Gain LowMid (0..127)
Byte 9: Send 3 (0..127)
Byte 10: Gain Low (0..127)
Byte 11: Send 4 (0..127)
Byte 12: Insert 1 (0..127)
Byte 13: Insert 2 (0..127)
Byte 14: Insert 3 (0..127)
Byte 15: Insert 4 (0..127)
Byte 16: Freq Low (0..127)
Byte 17: Q Low (0..127)
Byte 18: Freq LowMid (0..127)
Byte 19: Q LowMid (0..127)
Byte 20: Freq HighMid (0..127)
Byte 21: Q HighMid (0..127)
Byte 22: Freq High (0..127)
Byte 23: Q High (0..127)
Data Bytes 24..27 contain the 24 Bypass-Bits for each Mixer-Value
Bit0 of Byte 24 is the Bypass-Bit for Mixer-Value 0,
Bit1 of Byte 24 is the Bypass-Bit for Mixer-Value 1,
Bit6 of Byte 24 is the Bypass-Bit for Mixer-Value 6,
Bit0 of Byte 25 is the Bypass-Bit for Mixer-Value 7 etc.

Track-Mode Names 'T':

0x54 0x00 FirstParameter Count 24xData
FirstParameter 0..191. Each of the 32 Tracks has 6 Modes (Audio, Midi, Input, Inst, Bus, Group). With this Entry any of these assignment can be changed.
Only available in Active-Modes

Internal Status 's':
0x73 0x01 0x00 0x0x Trk Grp Mode
Trk: Currently selected Track (0..127, top 2 Bits specify Mixer-Mode)
Grp: Selected Track-Group
Mod: Selected Track-Mode
Only available in Active-Modes

[internal scratch
- Version-Number
- System-Dump depends on currently selected mode
=> All Active-Modes:
- Track-Names
- Mixer-Values
- Current Selections (Track)
=> Other Modes:
- Displays
- Fader&Encoder-Positions, Button States
internal scratch]

CONTROL CHANGES (for slave mode only)

The SAC sends these Messages, when the corresponding Button changes.
The Parameter dr is encoded as follows:
00h Button Released
01h Button Depressed
02h Button Depressed with Shift held

On Reception these Messages control the corresponding Button LEDs (if available). Then the Parameter dr is encoded as follows:

00h Led off
01h Led on
02h Led blinking

From Version 1.02 you can specify the Led-State as a 4-Bit sequence in Bits 0..3 when Bit 5 is set. The Led is off in the specific state when the Bit is cleared and on otherwise.
Some Examples (V1.02):
20h Led off
2Fh Led always on
2Ch Led Blink
25h Led fast Blink
2Ah Led fast Blink, alternate Phase
21h Led Flashing

Switches

Solo:	*Bxh dr 00*	Note: x=system channel
Mute#1:	*Bxh dr 01*	
...	...	(physical switch, not bank switched...)

Mute#8:	*Bxh dr 08*	
Select#1:	*Bxh dr 09*	
...	...	
Select#8:	*Bxh dr 10h*	
Master-Select:	*Bxh dr 11h*	receiption of these commands will result
Ins./Send:	*Bxh dr 12h*	in the corresponding LED being turned on/off
Pan:	*Bxh dr 13h*	(LED on if dr=01, LED off if dr=00, LED blink if dr=02)
High:	*Bxh dr 14h*	
Send1:	*Bxh dr 15h*	
HiMid:	*Bxh dr 16h*	
Send2:	*Bxh dr 17h*	
LowMid:	*Bxh dr 18h*	
Send3:	*Bxh dr 19h*	
Low:	*Bxh dr 1Ah*	
Send4:	*Bxh dr 1Bh*	
Audio:	*Bxh dr 1Ch*	
Midi:	*Bxh dr 1Dh*	
Input:	*Bxh dr 1Eh*	
Inst.:	*Bxh dr 1Fh*	
Bus:	*Bxh dr 20h*	
Group:	*Bxh dr 21h*	
Bank#1-8:	*Bxh dr 22h*	
Bank#9-16:	*Bxh dr 23h*	
Bank#17-24:	*Bxh dr 24h*	
Bank#25-32:	*Bxh dr 25h*	
System:	*Bxh dr 26h*	
ChannelStripEQ:	*Bxh dr 27h*	
Insert/Sends:	*Bxh dr 28h*	
Dynamics:	*Bxh dr 29h*	
Midi:	*Bxh dr 2Ah*	
Instrument:	Bxh dr 2Bh	
Recall-Marker:	*Bxh dr 2Ch*	
Jump-To:	*Bxh dr 2Dh*	
Store-Marker:	*Bxh dr 2Eh*	
Jump-From:	*Bxh dr 2Fh*	
Scrub:	*Bxh dr 30h*	
Rewind:	*Bxh dr 31h*	
Forward:	*Bxh dr 32h*	
Stop:	*Bxh dr 33h*	
Play:	*Bxh dr 34h*	
Record:	*Bxh dr 35h*	
Num:	*Bxh dr 36h*	
Enter:	*Bxh dr 37h*	
Shift:	*Bxh dr 38h*	
Key#1:	*Bxh dr 39h*	
...	...	
Key#10:	*Bxh dr 42h*	

<u>Encoder Push Buttons</u>

Encoder#1:	*Bxh dr 43h*	Note: dr=depression/release
...	...	(01/00, if shift pressed 02/00)
Encoder#12:	*Bxh dr 4Eh*	(physical encoder, not bank switched)

Fader Touch Sensors

Fader#1:	*Bxh tr 77h*	Note: tr=touch/release (01/00)
...	...	(physical fader, not bank switched)
Fader#9:	*Bxh tr 7Fh*	

Faders

Fader#1 MSB:	*Bxh 17h vv*	Note: vv = fader position (00..7Fh)
...	...	(physical fader, not bank switched)
Fader#9 MSB:	*Bxh 1Fh vv*	
Fader#1 LSB:	*Bxh 37h vv*	(reserved)
...	...	
Fader#9 LSB:	*Bxh 3Fh vv*	

Encoder Control

Encoder#1:	*Bxh 53h vv*	
...	...	
Encoder#12:	*Bxh 5Eh vv*	
Jog:	*Bxh 5Fh vv*	

SAC sends these messages, when there is a movement detected.
vv=01h..3Fh increment (clockwise movement),
vv=40h..7Fh decrement.

For Version 0.xx typically this is +-1 (01/7Fh)
Version 1.xx may send any Value between –63..+63

Receiption of these commands will result in
setting the corresponding Led around the Encoder (vv=0..1Fh,7F=OFF)
and Encoder Mode set to 04h (32 Steps)

Enhanced Encoder Control (V1.02)

Encoder #1:	Axh 43h vv	vv=00h..7Fh Encoder Mode
...		
Encoder #12:	Axh 4Eh vv	

Encoder Mode describes, how the Encoder Value is displayed on the Encoder Leds:

00h:	All Leds Off
01h:	128 Steps, Value is in the Range 0..127, Halftone output
02h:	100 Steps, Value is in the Range 0..99, Halftone output
03h:	64 Steps, Value is in the Range 0..63, Halftone output
04h:	32 Steps, Value is in the Range 0..31, Maps to the 31 Leds
05h:	16 Steps, Value is in the Range 0..15, 2 Leds per Step
06h:	8 Steps, Value is in the Range 0..7, 4 Leds per Step
07h:	4 Steps, Value is in the Range 0..3, 8 Leds per Step
08h:	reserved, don't use
09h:	128 Volume Steps, Value is in the Range 0..127, Band-Display
0Ah:	32 Volume Steps, Value is in the Range 0..31, Band-Display
0Bh:	16 Volume Steps, Value is in the Range 0..15, Band-Display
0Ch:	128 Panorama Steps, Value is in the Range 0..127, centered Band
0Dh:	32 Panorama Steps, Value is in the Range 0..31, centered Band
0Eh:	16 Panorama Steps, Value is in the Range 0..15, centered Band
0Fh:	2 Panorama Steps, Value is in the Range 0..1, Left/Right Band
10h:	128 Band Steps, Range 0..127, centered Band (V1.05)

11h: 32 Band Steps, Range 0..31, centered Band (V1.05)
12h: 16 Band Steps, Range 0..15, centered Band (V1.05)
13h: 2 Band Steps, Range 0..1, Center Dot/Full (V1.05)
14h: 128 Damping Steps, Range 0..127, left-turned Band (V1.05)
15h: 32 Damping Steps, Range 0..31, left-turned Band (V1.05)
16h: 16 Damping Steps, Range 0..15, left-turned Band (V1.05)
17h: 2 Damping Steps, Range 0..1, Right Dot/Full (V1.05)

Setting Bit 5+6 makes the Encoder-Leds Blink (V1.05):
20h: Normal Blink
40h: Fast Blink
60h: Flash

Encoder #1:	Axh 53h vv	vv=00h..7Fh Value, Range depends on Mode
...		
Encoder #12:	Axh 5Eh vv	

Time Display

Digit:	*Bxh 67h nn*	Note: nn=digit number, 00=right most, auto-increment
Write Char:	*Bxh 68h nn*	Note: nn=bit mapped as shown below (time display mode 3) or ASCII character (mode 2, bit 4 ist decimal point) with auto cursor increment
Write Dp:	*Bxh 69h nn*	Note: nn=bit mapped decimal point: bit 0=right-most, bit 6=next to left-most (only mode 3)
Write Dp:	*Bxh 6Ah nn*	Note: bit 0 = left-most decimal point (only mode 3)

Time Display Mode 3 bit map:

```
  --0--
6 |   | 5
  --1--
4 |   | 3
  --2--
```

The "Digit" command is analogous to a cursor command. It simply points to the location of the next digit which will receive a Write Char command. If 00, Write Char will write to the right most digit. Subsequent Write Chars will automatically increment, that is, the characters will appear in the display from right to left.

Time Display Mode 0:
time display write by mtc quarter frame commands (*F1h xx*), dots automaticly set.

Levelmeter (in Display)

Display#1:	*Bxh 6Dh nv*	Note: n=position (meter #1..4, left/right justified)
Display#2:	*Bxh 6Eh nv*	v=value (00..0Fh)
Display#3:	*Bxh 6Fh nv*	

Displays

Cursor Display#1:	*Bxh 70h cp*	Note: cp = cursor position from 00 through 27h for the top line, 40h through 67h for the bottom line.
Write Char:	*Bxh 71h cc*	cc = ASCII character with auto cursor increment.
Special:	*Bxh 72h nn*	nn=01 to clear display, nn=0Fh to turn on blinking cursor,

		nn=0Ch to turn off blinking cursor.
Cursor Display#2:	*Bxh 73h cp*	Note: see display #1
Write Char:	*Bxh 74h cc*	
Special:	*Bxh 75h nn*	
Cursor Display#3:	*Bxh 76h cp*	
Write Char:	*Bxh 77h cc*	
Special:	*Bxh 78h nn*	

Use the Cursor command to move the cursor to an initial starting position. Thereafter, it will auto increment. Note that the cursor itself may or may not be visible, depending upon reception of the special LCD command.