

german: 

DMX Interface Card Hardware: this is a complete step-by-step description how to access the hardware, and how to set the working parameters required for DMX512/1990. Find memory maps, hardware addresses and other hardware-related stuff.

WG18:

Hardware Description

1512A

1512B-LC

1512B

1512C

DMXPCP

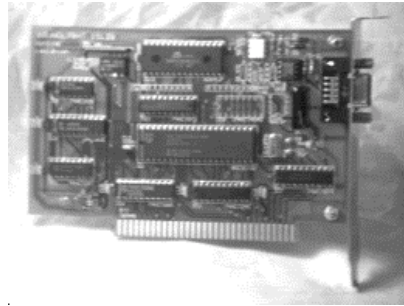
SERDMX

USB-DMX

← back

HOME PAGE

DMX-512 PC-Interface Card 1512A, 1512B-LC



SOUNDLIGHT offers a range of PC interface cards, not only readily available but affordable and easy to program. Besides we offer ready-to use [software packages](#) for free.

Programming of our interface cards is really simple. Here are the few steps to get things running.

FEATURES

1512A, 1512B-LC

*The 1512A is no longer in production. Its successor is the 1512B card, which features some improvements regarding output design and protection. When looking for support to any of these models, always please refer to: **1512A***

- fast card, Clock = 12 MHz
- easy to program, works with any PC (286 to Pentium)
- very low-cost
- **Memory Allocation of Interface Card 1512B:**
 - \$0000-\$03EF DMX operating system, max. 1000 Bytes
 - \$03F0-\$03FE Operating System Parameters dmxdef()
 - \$0400-\$05FF Send Data for Channels 1-512
 - \$0800-\$0FFF User Memory Area - for your own use

To initialize the card:

1. **Determine the card address.**

To do so, use a routine as outlined below. In all examples we will assume a card address of \$0100:

1. Write \$FF to card address \$0100
2. Write \$03 to card address \$0101 (0100 +1)
This sets the internal card RAM address to \$03FF.
3. Write \$AA to card address \$0102 (0100 +2)
4. Read card address \$0102 (0100+2)

5. If the return value is \$AA, the card is found.
6. If not, repeat steps (1) thru (5) for addresses \$0120, \$0140, \$0160
2. **Transfer the DMX Operating System into the card,**
to make the card operational. Assume you were going to find a 1512A card in your (unknown) computer.
 1. Read card address \$0100 to reset the card, then
 2. open the file SLHDMX12.BIN.
NOTE: Close this file before opening it. This ensures to never get a "file open" error. Depending on the programming language used, you must open the file as BINARY or RANDOM because all NULL bytes must be transferred exactly as read.

Then the file will be transferred into the card RAM. The card address must be known to do so.

3. Set RAM-Address = \$0000
4. Write the RAM-Address LowByte to Card-Address +0
5. Write the RAM-Address HighByte to Card-Address +1
6. Read the next byte from file and write it to Card-Address +2
7. Repeat (3) thru (5) until EOF

We have now transferred the DMX operating system and are ready to start the card. This will enable the card processor to check the card status and return a card identification byte.

8. Read Card Address +3 to start the card and wait at least 50us to read the result.
9. Write \$F9 to Card Address +0
10. Write \$03 to Card Address +1 This selects the return byte RAM address \$03F9 within the card RAM.
11. Read Card Address +2
Return value \$04: Card 1512C present
Return value \$05: Card 1512C present
Return value \$80: Card 1512A or 1512B/LC (12 MHz Card) present
Return value \$01: Card 1512B (16 MHz Card) present

If a 1512A Card is identified the initialization is okay. If another card is being identified the complete initialization procedure has to be repeated, using the proper DMX operating system. Please use the following files:

SLHDMX12.BIN for card 1512A or 1512B/LC
SLHDMX16.BIN for card 1512B
SLHDMX17.BIN for card 1512C

After file transferral the return value must be checked again. If the return value matches none of the values listed above, either the card or the DMX OS transfer process has shown defective. Return codes \$06 to \$1F are reserved for future SLH products.

3. **Transfer the Default Parameters into the card.**
The DMX operating system files allow individual setting of channel count, card

timing an other parameters. If no or invalid parameters are set the card may automatically set valid parameters instead.

Card-RAM Address	Parameter	Default or typical Value
\$03F0	Startsync Duration Minimum Value: 88us Value: $n = \text{Startsync} / 2$	dmxdef(0) = 45
\$03F1	StartByte for DMX Transmissions	dmxdef(1) = 0
\$03F2	End Idle Time (us) $n = (\text{Idle Time} - 30\text{us}) / 2$ The minimum idle time after transmission of a complete DMX telegram is 30 us	dmxdef(2) = 0
\$03F3 \$03F4	LowByte Channel Count HighByte Channel Count LO: 001 HI: 000 1 channel sent LO: 002 HI: 000 2 channels sent LO: 003 HI: 000 3 channels sent ... LO: 000 HI: 000 256 channels sent LO: 001 HI: 001 257 channels sent LO: 002 HI: 001 258 channels sent ... LO: 255 HI: 001 511 channels sent LO: 000 HI: 001 512 channels sent LO = Channel Count AND 255 HI = INT ((Channel Count -1)/256)	dmxdef(3) = 0 dmxdef(4) = 1
\$03F5	Send Repeat \$00: Send Once, then Stop (Recommended) \$80: (<>0): Send continuously	dmxdef(5) = 0
\$03F6	Inter-Digit Time $n = \text{IDT} * 16 / 24$ optimum Value: 0	dmxdef(6) = 0
\$03F7	not used	dmxdef(7) = 0
\$03F8	Card Mode \$00 = Write Mode \$FF = Read Mode <i>1512A, 1512B-LC: only Write Mode supported</i>	dmxdef(8) = 0
\$03F8	not used	dmxdef(8) = 0
\$03F9		dmxdef(9) = 0

	<i>see below</i>	
--	------------------	--

4. Now transfer the Default Parameter Values to the Card Data RAM:
 1. Set $n=0$
 2. Write $\$F0 + n$ to card address $+0$
 3. Write $\$03$ to card address $+1$
 4. Write $dmxdef(n)$ to card address $+2$
 5. $n= n+1$
 6. Repeat (2) thru (5) while n

Now the card is ready for use. Eventually you may want to erase the DMX data RAM. To do so, simply write $\$00$ to all remaining card RAM locations.

5. Start the Card

After initialization and setting of the operating system parameters the card may start its first transmission. To start the card, use the command:

1. Read Card Address $+3$

According to the card mode set above one or more DMX telegrams will be sent.

6. Transfer DMX data to the card

To transfer data to the card or read back data from the card) we highly recommend to use a timer. This will ensure the best data integrity, since access to the card during transmission may corrupt send data. Use a timer (timer repeat time >25 ms) to initiate the command sequence outlined below. Please note:

- Each write access to the card will stop the card CPU, which must then be restarted.
 - After all memory updates have been processed start the card to initiate e new DMX transmission.
 - The card will then complete the current DMX telegram an await the next data update.
4. Write LowByte of DMX Channel to CardAddress $+0$
 5. Write HighByte of DMX Channel to CardAddress $+1$
 6. Write Value of DMX Channel to CardAddress $+2$
 7. Repeat (1) thru (3) for all channels that have to be updated
 8. Start the card by reading from CardAddress $+3$

SOFTWARE ACCESS

	READ	WRITE
Base Address $+0$		Write Lo-Byte RAM-Address

Base Address +1		Write Hi-Byte RAM-Address
Base Address +2	Read Data	Write Data
Base Address +3	Start DMX Transfer	

S/W Access Table Base Address = \$100, \$120, \$140 or \$160
 Any Write Access stops the DMX-Transfer

HARDWARE SETUP

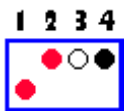
- SWITCH SETTINGS**

Do not change switch settings unless required. The cards come with address \$0100 (256 dec) set.

Switches 1 and 2 set the card's interface address



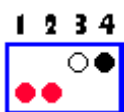
CARD ADDRESS **\$0100** (256 dec)



CARD ADDRESS **\$0120** (288 dec)

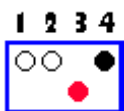


CARD ADDRESS **\$0140** (320 dec)

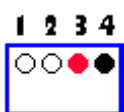


CARD ADDRESS **\$0160** (352 dec)

Switch 3 sets the card type identification. It MUST be set in accordance with the card's crystal frequency.

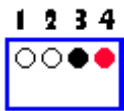


12 MHz CARD *typical setting for 1512A, 1512B-LC*



16 MHz CARD

Switch 4 is for factory use only and must never be changed.



ALWAYS ON

FREQUENTLY ASKED QUESTIONS / HOW TO .. ?

- **I can't put the card to work.**
All cards are fully checked prior to shipment. Of course any electronic component can fail, but in a computer system chances are that address, software or hardware conflicts exist. Please try one of our ready-to use [programs](#) first. Most of these programs show a clear identification of the card after initialization in a message box. Address, card type and/or frequency will be displayed. If the card cannot be found please check for an address conflict.
- **OK, your program is running fine but the card does not respond to my program. It does nothing.**
We have prepared several demos which are presented in our [programming lessons](#). Use these demos as a skeleton for your own projects. Before the card can be used you must transfer the card operating system into the card. If bytes are transferred erroneously, missing or faulty the card will not work! For Beginners it is a good idea to read out the card contents after transferral to make sure everything is okay and matches the SLHDMXnn.BIN file's contents.
- **A card was found but the wrong card type displayed.**
Please check the card switch setting. SW3 must be set according to the card's crystal frequency to allow identification..
- **I have a 1512C card and this cannot be identified when using the routines described above.**
This is a shortcoming of the current SLHDMX12.BIN and SLHDMX16.BIN driver files, which have been written prior to the release of the 1512C card. Thus the drivers do not know this card type. New drivers will be released shortly. Please use the identification scheme as outlined on the [1512C](#) page..
- **What about drivers?**
We supply DLL's which can be used with Windows 3.1/3.11 and Windows 95/98, respectively. Please check the [drivers download](#) area below to obtain the latest drivers available. A description how to access the DLL functions is contained in the 1512B [manual](#) and every source code demo of our