

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

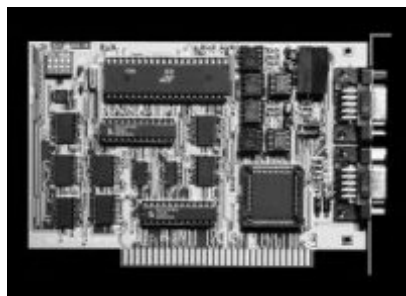
SERDMX

USB-DMX

← back

HOME PAGE

### ISA DMX-512 PC-Interface Card 1512C



We offer 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 1512C

- very fast card, Clock = 16 MHz
- Two outputs, two inputs, 512 channels each
- opto-isolated output drivers (potential-free output)
- DMX read and write simultaneously for 1024 channels
- low-cost
- **Memory Allocation of Interface Card 1512C:**

\$0000-\$03EF	DMX operating system, max. 1000 Bytes
\$03F0-\$03FE	Operating System Parameters dmxdef()
\$0400-\$05FF	Send Data for Universe1, Channels 001-512
\$0600-\$07FF	Send Data for Universe2, Channels 513-1024
\$0800-\$09FF	Receive Data for Universe1, Channels 001-512
\$0A00-\$0BFF	Receive Data for Universe1, Double Buffer
\$0C00-\$0DFF	Receive Data for Universe2, Channels 513-1024
\$0E00-\$0FFF	Receive Data for Universe2, Double Buffer

**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 are going to find a 1512C card in your (unknown) computer.

1. Read card address \$0100 to reset the card, then
2. open the file SLHDMX17.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. Write zero to 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 1512C 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 and other parameters. If no (or invalid) parameters are set the card will automatically default to valid parameters instead. This may be the best choice, especially for beginners.

All functions available from the next table are only supported by the latest OS file SLHDMX17.BIN (file dated 19.09.00 or newer), which can be [downloaded here](#). Make sure, that older versions are deleted or updated.

Card-RAM Address	Parameter	Default or typical Value
\$03F0	following parameters are valid 0 or >0	dmxdef(0) = 0
\$03F1	StartByte Universe 1 for DMX Transmissions, CH. 001-512	dmxdef(1) = 0
\$03F2	StartByte Universe 2 for DMX Transmissions, CH. 513-1024	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 <hr/> LO = Channel Count AND 255 HI = INT ((Channel Count - 1)/256)	dmxdef(3) = 0 dmxdef(4) = 1
\$03F5	Send Repeat \$80: send always \$40: Send Once, then Stop (Recommended) \$80: (<>0): Send continuously \$60: (re)start a one-time transmission	dmxdef(5) = \$80
\$03F6	not used	dmxdef(6) = 0
\$03F7	Receive Double Buffering	dmxdef(7) = 0

	<hr/> UNIVERSE 2 (513-1024) Bit5 = Double Buffering Active Bit4 = Buffer Number [0/1] <hr/> UNIVERSE 1 (001-512) Bit1 = Double Buffering Active Bit0 = Buffer Number [0/1]	
\$03F8	not used	dmxdef(8) = 0
\$03F9	Card Type Return Value \$80 = Card 1512A / 1512B-LC \$01 = Card 1512B \$04/05 = Card 1512C \$44/45 = Card 1512C ROM-based Version	dmxdef(9) = 0
\$03FA \$03FB	LowByte Receive Channel Counter HighByte Receive Channel Counter for Universe 0 (000-512)	READ ONLY
\$03FC \$03FD	LowByte Receive Channel Counter HighByte Receive Channel Counter for Universe 1 (513-1024)	READ ONLY
\$03FF	Receive Input Status Bit7 = Status Universe 2 (513-1024) Bit6 = Status Universe 1 (001-512)	READ ONLY

Now transfer the Default Parameter Values to the Card Data RAM:

*Do nothing if you want to use factory defaults*

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.

#### 4. 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. Write 0 to Card Address +3

After starting the card once no further start/stop commands should be issued. In contrary to 1512A and 1512B cards the 1512C must not be stopped to ensure DMX data reception at all times.

## 5. 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.

1. Write LowByte of DMX Channel to CardAddress +0
2. Write HighByte of DMX Channel to CardAddress +1
3. Write Value of DMX Channel to CardAddress +2
4. Repeat (1) thru (3) for all channels that have to be updated

---

## SOFTWARE ACCESS

	<b>READ</b>	<b>WRITE</b>
Base Address +0	CPU Reset (Stops DMX Transfer)	Write Lo-Byte RAM-Address
Base Address +1		Write Hi-Byte RAM-Address
Base Address +2	Read Data	Write Data
Base Address +3		Release RESET, Start DMX Transfer

*S/W Access Table Base Address = \$100, \$120, \$140 or \$160  
Write Access does not affect DMX-Transfer. RESET function is latching.*

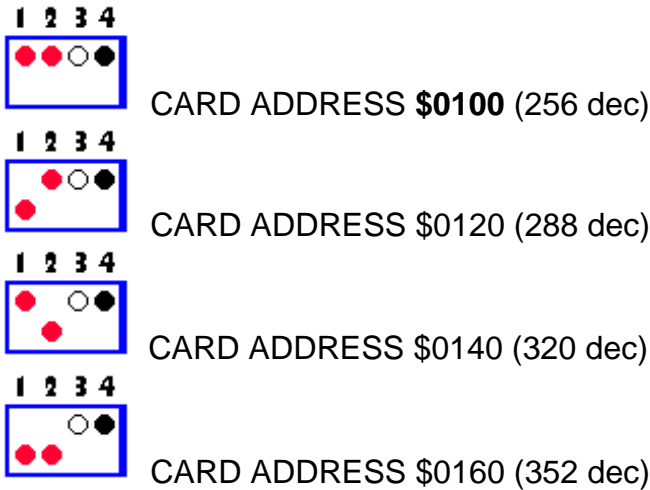
---

## HARDWARE SETUP

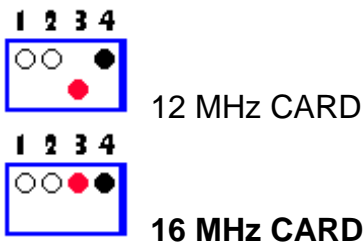
- **SWITCH SETTINGS**

*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***



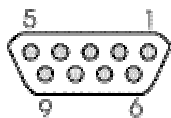
**Switch 3 sets the card type identification. It *MUST* be set in accordance with the card's crystal frequency.**



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



**DMX-512 PORT II  
9Pin female**

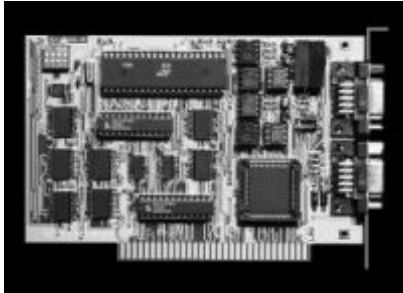


This is the DMX interface connector as used with our [Interface Card 1512C](#). Please be aware, that [DMX-512](#) requires the 5-pin XLR as standard connector. As this type of connector is too bulky to be mounted on a slot card, the 9-pin Sub D style was chosen and we recommend the use of an adaptor cable, which you can easily do by yourself. Alternatively, it is readily available from your supplier (part number 1512C-AK, two pcs. needed).

PIN	INPUT / OUTPUT	ASSIGNMENT
1	INPUT	DMX +
2	INPUT	DMX -
3	OUTPUT	DMX +
4	OUTPUT	DMX -

5		do not use
6	GND	Screen, GND
7	GND	Screen, GND ( <i>optional</i> )
8	GND	Screen, GND ( <i>optional</i> )
9	GND	Screen, GND ( <i>optional</i> )

---



upper connector:  
UNIVERSE 1 (Channels 001-512)

lower connector:  
UNIVERSE 2 (Channels 513-1024)

**Linking two Cards:** Use a straight 9-pin Sub D male/male cable to connect a 1512B card (used as transmitter) to a 1512C card (used as receiver).

---

## 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.. The 1512C card scans several card registers to ensure correct identification of the correct card type. If the card ID return register \$03F9 is read too fast after starting the card CPU (step 2.8 above) the result may not yet have been written back to the ID register. Increase the delay before accessing the ID register.
- My program works fine with 1512A und 1512B cards. I now have a 1512C card and this cannot be identified when using the standard initialization routines.**

*Please note that there is a little difference in how to handle 1512A/1512B and 1512C cards in terms of starting/stopping the card CPU. While a timer-driven stop/load/start scheme is used with 1512A/1512B cards, the 1512C card can randomly be accessed. Furthermore, it is not possible to detect a 1512C card in a 1512A/1512B installation. 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*