AVRDMX

DOCUMENTATION FOR AVRDMX-DONGLE,

MK 255

Interface protocoll

AVRDMX interface manual, version 1.0 (06/05/2000)

Issued by:

Kari Viitala
email: viikari@ee.tut.fi

INTRODUCTION

The AVRDMX-dongle is simple parallel port interface for connecting DMX-512 lights to personal computer. Parallel port must be ECP-type port if dongle read-back is used. Allmost all new computers have this type parallel port. Parallel port must be set in the computer bios to the ECP-mode, in many computers it is set default to the SPP mode. The AVRDMX-dongle can simultaneously send and receive DMX-512 data. Maximum number of channels for the DMX input is 127 and for DMX output 255. Incomig data can be merged to the host computer data using highest takes presence (HTP) method. It also is possible to save default startup scene to the dongle internal memory. When power is applied to the dongle it restores scene from its memory and starts to send it. When computer is connected to the dongle it overwrites internal data and computer data is written to DMX line. DMX output from dongle is stardard USITT DMX-512 data. Break lenght is 100-200µs, mark after break is about 30µs. Typical DMX-line update rate is 30-35Hz.

INTERFACE PROTOCOL

Transmission protocol between personal computer and AVRDMX-dongle is quite simple. Protocol have four different cycles, two read and two write cycles. Interface uses following parallelport control signals:

Table 1: Parallel port signals

PORT PIN	PIN NAME	PIN FUNCTION
PIN 1	STROBE	Read/$\overline{\text{Write}}$, read (high) or write(low) data to dongle
PIN 2-9	DATA	DATA I/O
PIN 11	BUSY	WAIT signal, Dongle activate (high) this signal when it cannot receive data or data has been received succesfully. When data is read or written to/from dongleWAIT signal goes high when command has been done. PC should watch this signal
PIN 14	LINEFEED	$\overline{\text{DMX\_STROBE}}$, PC takes this low when it wants to read or write DMX-data to/from dongle
PIN 16	INIT PRINT.	MODE_SELECT, Data read or write mode control
PIN 17	SELECT PRINT.	$\overline{\text{AUX\_STROBE}}$, PC takes this low when it wants to read or write auxliary data

READ/WRITE MODES

DMX modes, Use **DMX_STROBE**

Table 2: READ and WRITE MODES

MODE_SELECT=0 R/W=0	DMX data write to dongle memory. Data bytes are written in sequence starting at channel 1 and ending to channel 255. It is also possible to write less than 255 channels.

MODE_SELECT=0 R/W=1	DMX data read from dongle memory. Data bytes are read in sequence starting at channel 1 and ending to channel 127. It is also possible to read less than 127 channels.

AUX mode, Use **AUX_STROBE**

MODE_SELECT=1 R/W=0	STATUS register write.STATUS : bit 0: DMX Receiver Enable, 1=enabled, 0=disabled, default=1. When bit is unset DMX receiver is disabled. bit 1: DMX merge Enable, 1=enable, 0=disable, default=1 When bit is set DMX input is merged to the host computer data using HTP (Highhest Takes Precense). bit 2: Reserved for future use bit 3: Reserved for future use bit 4: Reserved for future use bit 5: Reserved for future use bit 6: Default scene write command. When bit is set, dongle writes current host input data to its internal memory. When status is read this bit is allways zero. **Note!** DMX receiver and transmitter are disabled during memory write operation. Writing takes about 0.5 second. bit 7: Counter clear. When bit is set, all read and write counters are cleared and all read and write operations are terminated. When status is read this bit is allways zero.

MODE_SELECT=1 R/W=1	STATUS register read. Status register functions are described above.

PARALLEL PORT SETTINGS FOR AVRDMX-DONGLE

Parallel port should be initialized in PC bios to ECP (Enhanced Capabilities Port) mode. Then in application program, parallel port should be initialized by writing to register (**LPT base address)+(402hex**) = **20hex** (Common LPT base addresses are LPT1=378hex, LPT2=278hex). This register is ECP Extended Control Register. This command will set parallel port to bidirectional byte mode. When data is read from dongle, parallel port drivers must be disabled and then enabled when data is written. Port drivers are disabled when bit 5 is written to one in parallel port control register. Note that some parallel port signals are inverted by parallel port hardware. When there is written one actual output is zero.

Parallel port registers:

BASE: Base address for data, LPT1=378h, LPT2=278h
BASE+1: Status port Read Only

Bit 7	Busy (INVERTED)
Bit 6	Ack
Bit 5	Paper Out
Bit 4	Select In
Bit 3	Error
Bit 2	IRQ (Not)
Bit 1	reserved
Bit 0	reserved

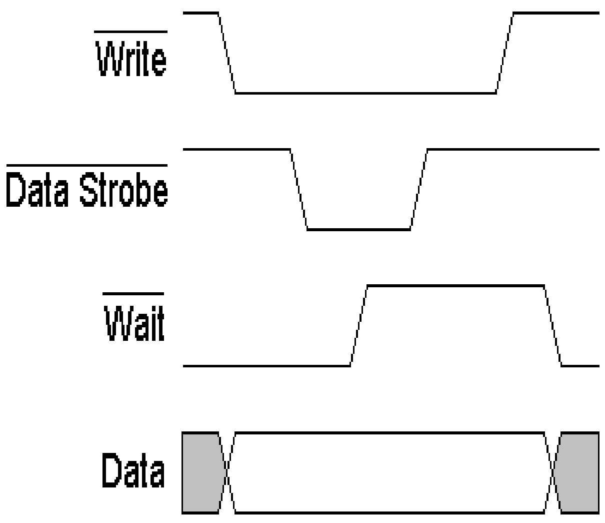
BASE+2: Control Port Read/Write

Bit 7	Unused
Bit 6	Unused
Bit 5	Enable Bi-Directional Port
Bit 4	Enable IRQ Via Ack Line
Bit 3	Select Printer (INVERTED)
Bit 2	Initialize Printer (Reset)
Bit 1	Auto Linefeed (INVERTED)
Bit 0	Strobe (INVERTED)

READ AND WRITE TIMING DIAGRAMS

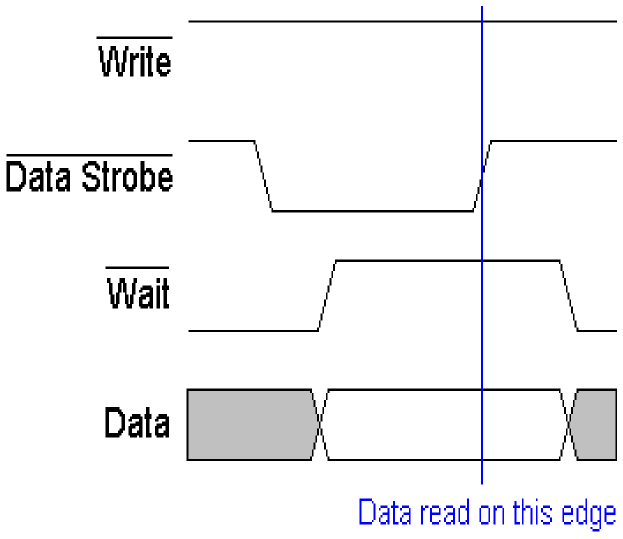
Data write to dongle:

1. Set MODE = 0
2. Set WRITE to LOW
3. Set Data to databus
4. Set DMX_STROBE to LOW
5. Wait until WAIT goes HIGH
6. DMX_STROBE back to HIGH



Data read from dongle

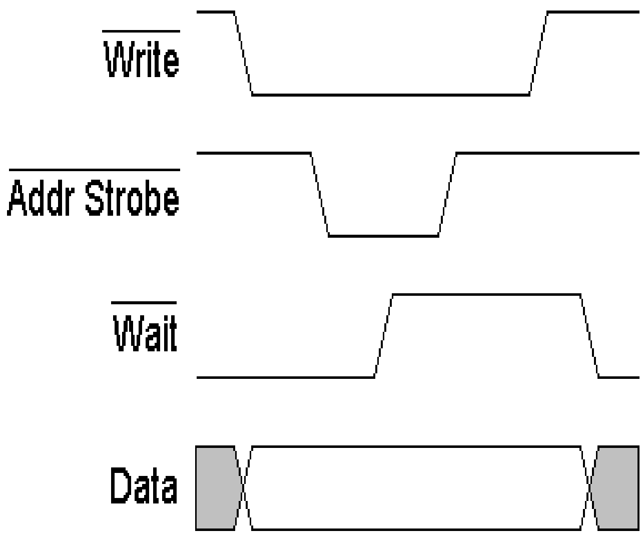
1. Set DIRECTION (=Bit 5) to INPUT (= 1) in parallel port control register.
2. Set MODE = 0
3. Set WRITE to HIGH
4. Set DMX_STROBE to LOW
5. Wait until WAIT goes HIGH
6. Read data from databus
7. DMX_STROBE back to HIGH



AUX write to dongle

Status write to dongle.

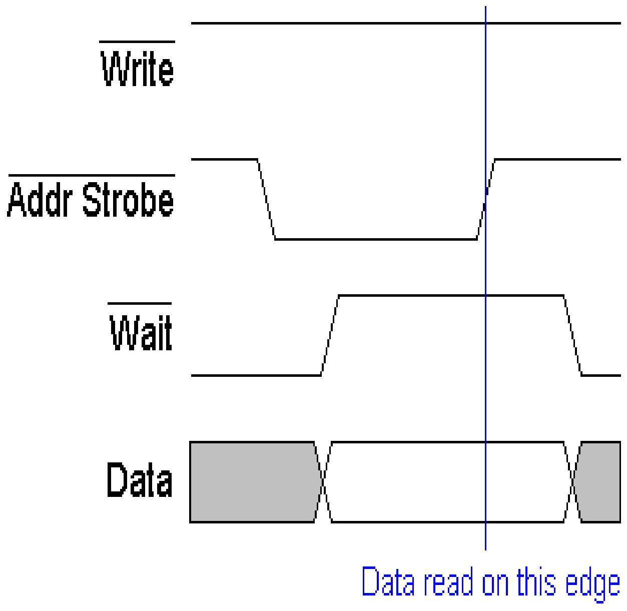
1. Set MODE = 1
2. Set WRITE to LOW
3. Set Data to databus
4. Set AUX_STROBE to LOW
5. Wait until WAIT goes HIGH
6. AUX_STROBE back to HIGH



AUX read from dongle

Status read from dongle.

1. Set DIRECTION (=Bit 5) to INPUT (= 1) in parallel port control register.
2. Set MODE = 1
3. Set WRITE to HIGH
4. Set AUX_STROBE to LOW
5. Wait until WAIT goes HIGH
6. Read data from databus
7. AUX_STROBE back to HIGH



EXAMPLES

Here is some simple examples how to use AVRDMX dongle under Visual Basic 5. Note that OUT and INP functions are not included to standard Visual Basic.

At first parallel port must be initialized. This is done by writing to ECP control register 20hex. Control register locates at address (LPT baseaddress) + 402hex, thus 378hex+402hex =77Ahex. This initializes parallel port bytewide-bidirectional mode.

Initialization routine:

```
Public Sub INITLPT()
        initialize parallel port registers to correct state
        'Remember to set parallel port in bios to ECP mode
        Out PortAddress + 2, &H1 'write to control port, write byte to low and other bytes to zero
        Out PortAddress + &H402, &H20   'ENABLE BYTE MODE
End Sub
```

Data write to dongle

Following code example shows how to write data byte dongle.

```
Public Sub DMX_BYTEOUT(Value As Integer)   'Send one dmx value
    Dim timeoutcount As Integer
    Dim a As Integer

    'set channel value to data port
    Out PortAddress, Value
    'set to data write mode, (write to low, data_strobe=high)
    Out PortAddress + 2, &H1
    'assert data strobe, (write= low, data_strobe=low)
    Out PortAddress + 2, &H3
    'wait until readyor timeout, (read busy bit)
    a = &H80
    While a = &H80
        a = Inp(PortAddress + 1)
        a = a And &H80
        timeoutcount = timeoutcount + 1
        If timeoutcount = 2000 Then
            a = &H0
            timeoutcount = 0
        End If
    Wend
    'deassert data strobe
    Out PortAddress + 2, &H1

End Sub
```

Data read from dongle:

Example shows how to read single byte from dongle.

```
Public Function DMX_BYTEIN() As Integer 'read one dmx value
    Dim timeoutcount As Integer
    Dim Value As Integer
    Dim a As Integer

     'set to data read mode (write=high, bidirectionalmode=1)
    Out PortAddress + 2, &H20
    'assert data strobe
    Out PortAddress + 2, &H22

    'wait until ready or timeout
    a = &H80
    While a = &H80
        a = Inp(PortAddress + 1)
        a = a And &H80
        timeoutcount = timeoutcount + 1
        If timeoutcount = 2000 Then
            a = &H0
            timeoutcount = 0
        End If
    Wend
     'Read byte from data port
    Value = Inp(PortAddress)
    'deassert data strobe
    Out PortAddress + 2, &H20
    DMX_BYTEIN = Value
End Function
```