

RS232 PROTOCOLS WITH CHECKSUM

Always start by reading the protocol of the device thoroughly.

All good configuration starts this way!!!

In section 2.3 of the Philips RS232 protocol we see the structure for the messages we are about to construct for the Philips display.

Notice the way the Checksum is calculated, in this case it's done with an XOR function.

**SERIAL / ETHERNET INTERFACE
 COMMUNICATION PROTOCOL
 SPECIFICATION
 (SICP V1.88)**
**For
 PHILIPS Professional Displays**

2.3 Command Format

The serial/Ethernet command packet format is as follows:

MsgSize	Control	Group	Data[0]	Data[1]	...	Data[N]	Checksum
---------	---------	-------	---------	---------	-----	---------	----------

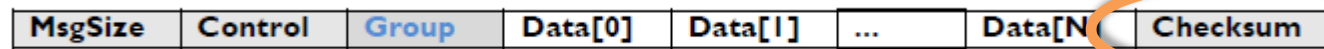
Every field of packet format consists of one byte – MsgSize = 1 byte, etc.

In detail:

Number of Field	Name of Field	Description															
Byte 1:	MsgSize	Message Size has to be calculated in the following way: $MsgSize + Control + Data(0) + \dots + Data(N) + Checksum$ Range = 3 to 40 (0x3 to 0x28).															
Byte 2:	Control	Message Control. Bit 7..0: Monitor ID Signal mode: Display Address range from 1 to 255 Broadcast mode: Display Address is 0 which indicates no ACK or Report is expected.															
Byte 3:	Group	Group ID range: Off(for old command),1-254 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Monitor ID</th> <th>Group ID</th> <th>Range</th> </tr> </thead> <tbody> <tr> <td>0-255</td> <td>0-254</td> <td>Range</td> </tr> <tr> <td>0</td> <td>0</td> <td>Broadcast</td> </tr> <tr> <td>1-255</td> <td>0</td> <td>Control by Monitor ID</td> </tr> <tr> <td>0-255</td> <td>1-254</td> <td>Control by Group ID</td> </tr> </tbody> </table>	Monitor ID	Group ID	Range	0-255	0-254	Range	0	0	Broadcast	1-255	0	Control by Monitor ID	0-255	1-254	Control by Group ID
Monitor ID	Group ID	Range															
0-255	0-254	Range															
0	0	Broadcast															
1-255	0	Control by Monitor ID															
0-255	1-254	Control by Group ID															
Byte 4 to Byte 39:	Data[0] to Data[N]	Data. This field can be also empty. If not empty then the range of Data Size, N = 0 to 36 (0x24).															
Last Byte:	Checksum	Checksum. Range = 0 to 255 (0xFF). Algorithm: The EXCLUSIVE-OR (XOR) of all bytes in the message except the checksum itself. $Checksum = [MSG-SIZE] XOR [CONTROL] XOR DATA[0] \dots XOR DATA[N]$															

Philips uses a protocol, where they ensure the integrity of the sent message by calculating the checksum!

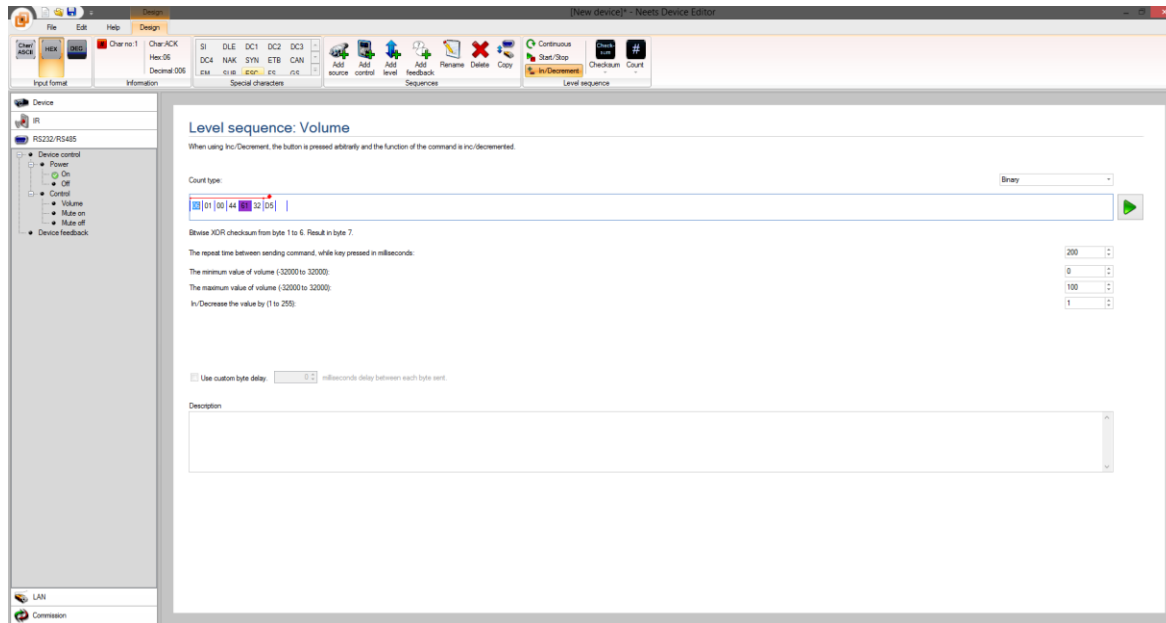
The checksum is a mathematical calculation of all of the bytes prior to the checksum!



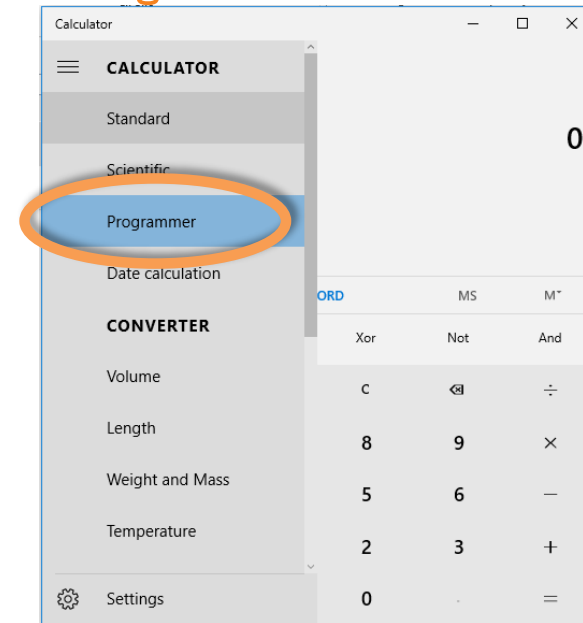
Each field is 1 Byte = 8 bits = 2 Hex digits

What do we need in order to calculate the checksum???

Neets Device Editor



Windows calculator in Programmer mode



In this case we would like to change an input, therefore we have to do the following:

- Find the right section in the protocol (Message Set)
- Choose the input we want to set (HDMI 2)
- Note the command and use the example in order to edit our command.

5.1.1 Message-Set

Bytes	Bytes Description	Bits	Description
DATA[0]	0xAC = Input Source – Set		Command requests the display to set the current input source
DATA[1]	Input Source Type/Number		0x01 = VIDEO 0x02 = S-VIDEO 0x03 = COMPONENT 0x04 = CVI 2 (not applicable) 0x05 = VIDEO 0x06 = HDMI 2 0x07 = Display Port 2 0x08 = USB 2 0x09 = Card DVI-D 0x0A = Display Port 1 0x0B = Card OPS 0x0C = USB 1 0x0D = HDMI 0x0E = DVI-D 0x0F = HDMI3 0x10 = BROWSER 0x11 = SMARTCMS 0x12 = DMS (Digital Media Server) 0x13 = INTERNAL STORAGE 0x14 = Reserved 0x15 = Reserved
DATA[2]	Reserved		Reserved
DATA[3]	OSD Style	Bit7	Reserved
		Bit6	Reserved
		Bit2.0	Source info. Display Style 0 = Reserved 1 = Source label
DATA[4]	Mute Style	Bit 7	(Reserved, value is 0)
		Bit 6	(Reserved, value is 0)
		Bit 5	(Reserved, value is 0)
		Bit 4	(Reserved, value is 0)
		Bit 3	(Reserved, value is 0)
		Bit 2	(Reserved, value is 0)
		Bit 1	(Reserved, value is 0)
		Bit 0	(Reserved, value is 0)

5.1.1 Message-Set

Rewrite the command so that it fits...

Bytes	Bytes Description	Bits	Description
DATA[0]	0xAC = Input Source – Set		Command requests the display to set the current input source
DATA[1]	Input Source Type/Number		0x01 = VIDEO 0x02 = S-VIDEO 0x03 = COMPONENT 0x04 = CVI 2 (not applicable) 0x05 = VGA 0x06 = HDMI 2 0x07 = Display Port 2 0x08 = USB 2 0x09 = Card DVI-D 0x0A = Display Port 1 0x0B = Card OPS 0x0C = USB 1 0x0D = HDMI 0x0E = DVI-D 0x0F = HDMI3 0x10 = BROWSER 0x11 = SMARTCMS 0x12 = DMS (Digital Media Server) 0x13 = INTERNAL STORAGE 0x14 = Reserved 0x15 = Reserved
DATA[2]	Reserved		Reserved
DATA[3]	OSD Style	Bit7 Reserved Bit6 Reserved Bit2:0 Source info. Display Style 0 = Reserved 1 = Source label	
DATA[4]	Mute Style	Bit 7 (Reserved, value is 0) Bit 6 (Reserved, value is 0) Bit 5 (Reserved, value is 0) Bit 4 (Reserved, value is 0) Bit 3 (Reserved, value is 0) Bit 2 (Reserved, value is 0) Bit 1 (Reserved, value is 0) Bit 0 (Reserved, value is 0)	

Example: Set on DVI-D with Source label displaying on OSD (Display address 01)

MsgSize	Control	Group	Data (0)	Data (1)	Data (2)	Data (3)	Data (4)	Checksum
0x09	0x01	0x00	0xA	0x09	0x09	0x01	0x00	0xAD

MsgSize	Control	Group	Data (0)	Data (1)	Data (2)	Data (3)	Data (4)	Checksum
09	01	00	AC	06	09	01	00	???

Now... what is the checksum???

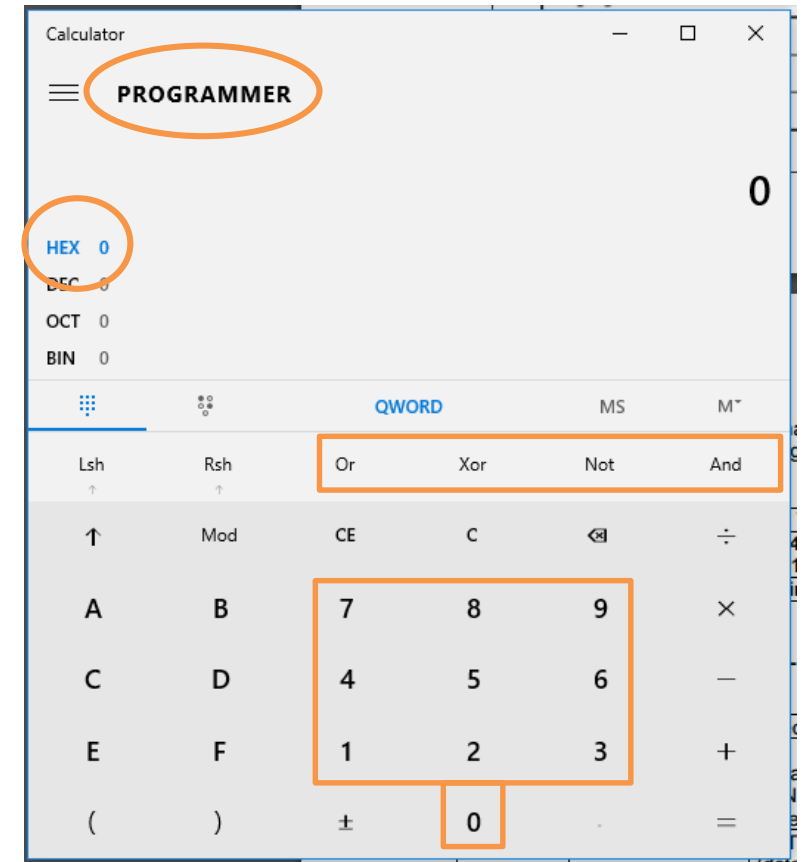
Checksum

MsgSize	Control	Group	Data (0)	Data (1)	Data (2)	Data (3)	Data (4)	Checksum
09	01	00	AC	06	09	01	00	???

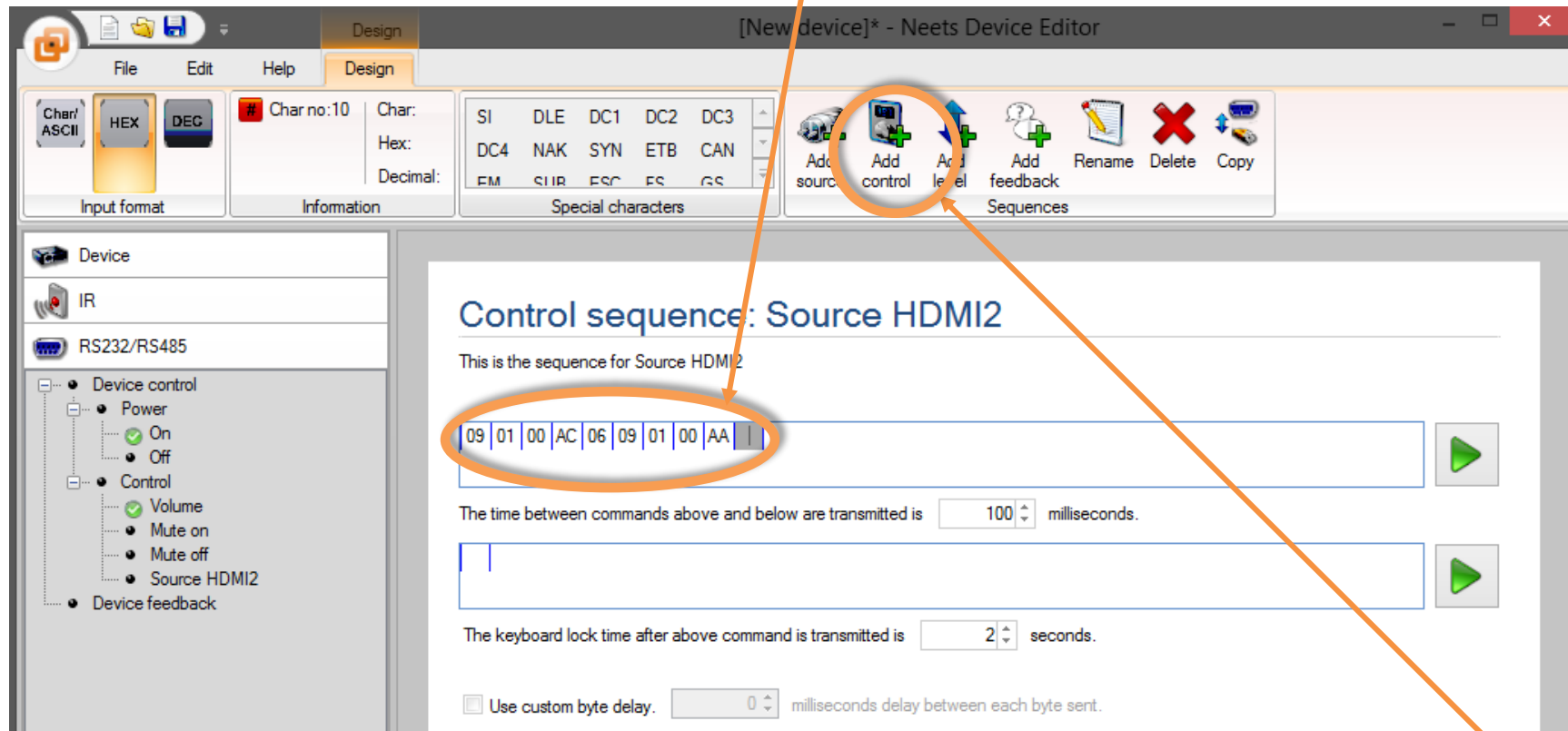
Now add all of the bytes like this:

$$09 \text{ Xor } 01 \text{ Xor } 00 \text{ Xor } AC \text{ Xor } 06 \text{ Xor } 09 \text{ Xor } 01 \text{ Xor } 00 = \text{Checksum} = AA$$

Philips uses the xor function, but normal addition is also very common in other protocols



Now insert the string into Neets Device Editor



This command is done, and the procedure can be repeated for all other controls...

The next thing is to create a step volume kontrol,
Where the checksum is calculated automatically.

Protocol

7.1.3 Message-Set exact volume level for Speaker out or Audio Out

This command can set volume level for speaker & audio out individually. Valid values range from 0x00 (lowest 0% volume) through 0xFE (highest – 100% volume). If DATA [1] or [2] value supplied is "0xFF" no action will be taken in the display and current volume level will be maintained without any effect.

Bytes	Bytes Description	Bits	Description
DATA[0]	0x44 = Volume – Set		
DATA[1]	Speaker Out Volume level		0 to 100 (%) of the user selectable range of the display.
DATA[2]	Audio Out Volume level		0 to 100 (%) of the user selectable range of the display.

Example: Set the Display Volume to 97% for Speaker out and 50% for Audio out (Display address 01)

MsgSize	Control	Group	Data (0)	Data (1)	Data (2)	Checksum
0x06	0x01	0x00	0x44	0x61	0x32	0xD5

Find the correct values!

Device Editor

Level sequence: Volume

When using Inc/Decrement, the button is pressed arbitrarily and the function of the command is inc/decremented.
No count byte(s) selected or no text entered

Count type: Binary

06 01 00 44 61 32 D5

There is no checksum defined.

The repeat time between sending command, while key pressed in milliseconds: 200

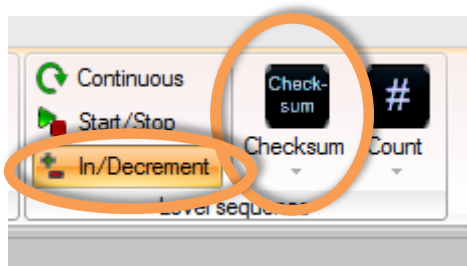
The minimum value of volume (-32000 to 32000): 0

The maximum value of volume (-32000 to 32000): 100

In/Decrease the value by (1 to 255): 1

Use custom byte delay. 0 milliseconds delay between each byte sent.

Insert them in Device Editor!



Now learn your device driver to calculate the checksum automatically.

Count type:

06 | 01 | 00 | 44 | 61 | 32 | D5 | |

Select the fields you want to perform the checksum calculation for.

1 6

Next >

Select the bytes you wish to calculate.

Count type:

06 | 01 | 00 | 44 | 61 | 32 | D5 | |

Select the target byte of where to place the calculation.

7

< Back Next >

Select the target byte, where to put the checksum calculation.

Count type:

06 | 01 | 00 | 44 | 61 | 32 | D5 | |

Finally choose what checksum calculation to use.

Bitwise XOR

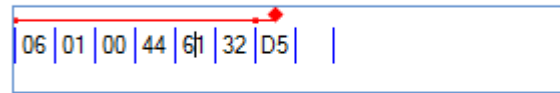
< Back Done

Choose the method of calculation.

Define the field where the count is being performed

Place your cursor in the field.

Count type:

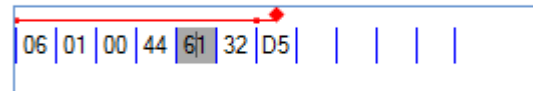


Bitwise XOR checksum from byte 1 to 6. Result in byte 7.



Field is marked

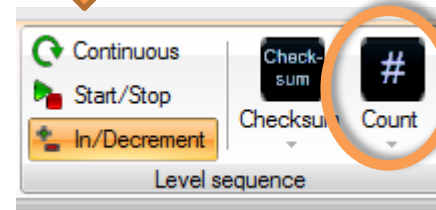
Count type:



Unknown checksum from byte 1 to 6. Result in byte 7.

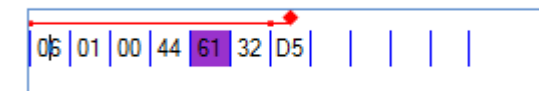


Now press "Count" and "Define count".



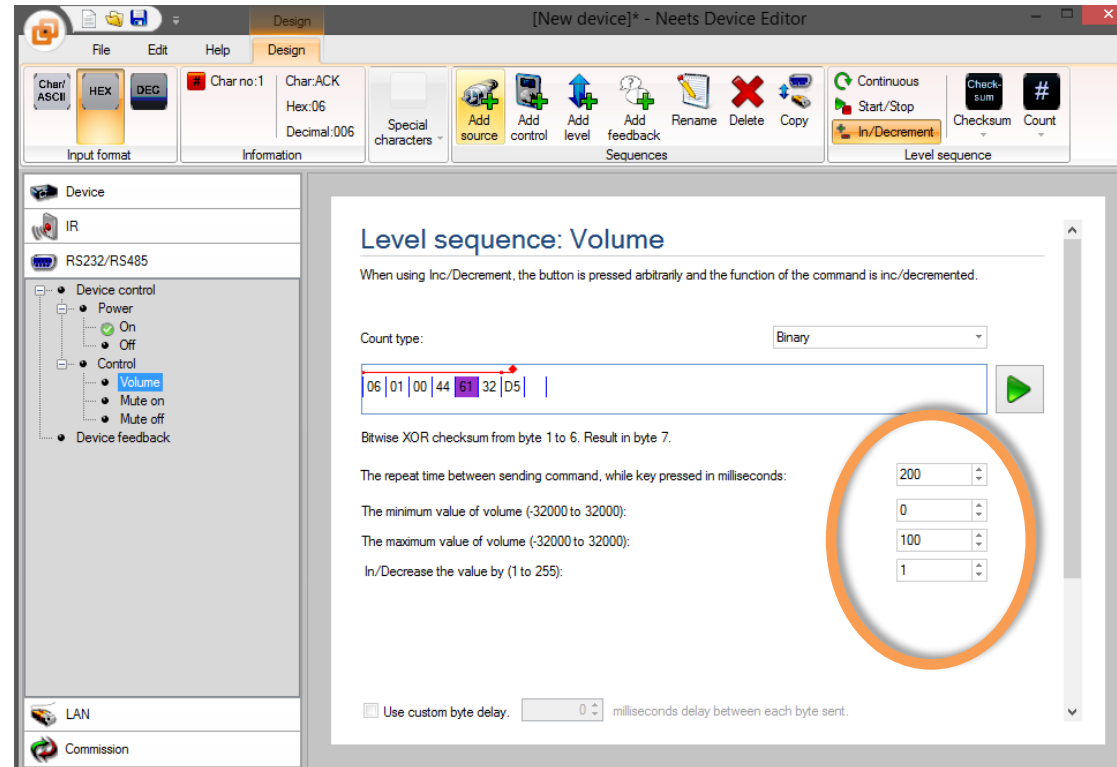
The field now turns purple.

Count type:



Unknown checksum from byte 1 to 6. Result in byte 7.

Now choose the in/decrease value, repeat time and step value



You're done 😊