ibm-tape-drive-automatic-standalone

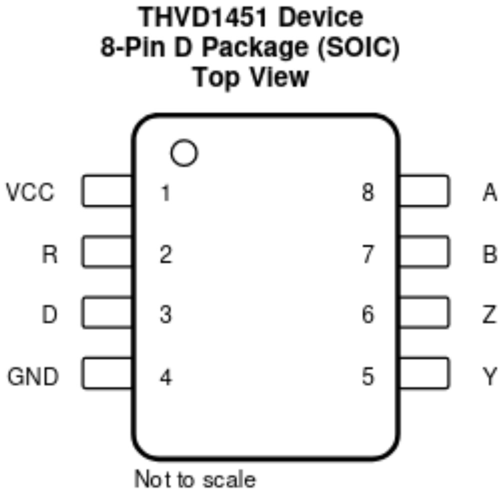
How to turn an IBM Drive into automatic standalone mode

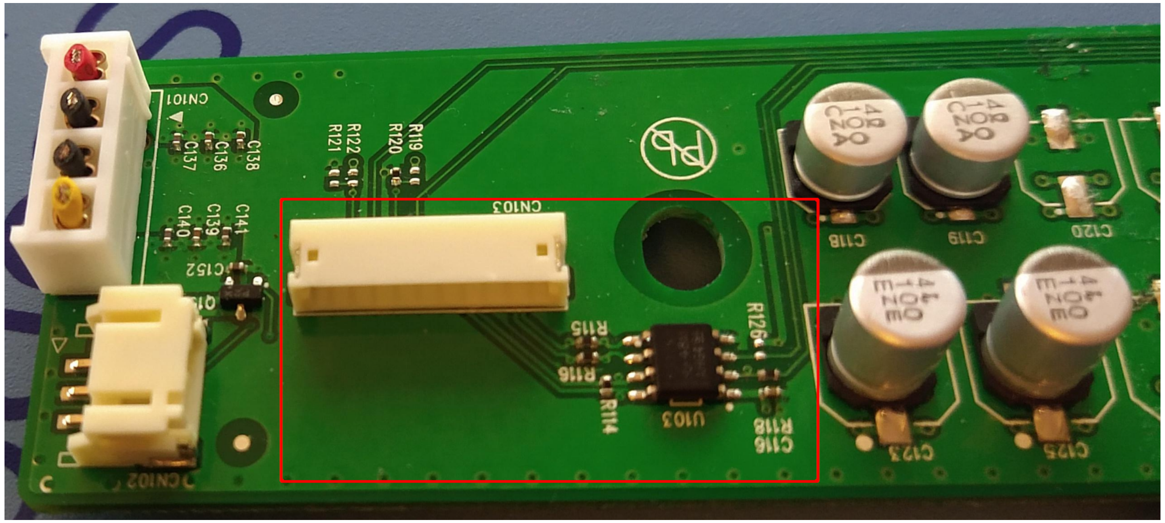
Hello all

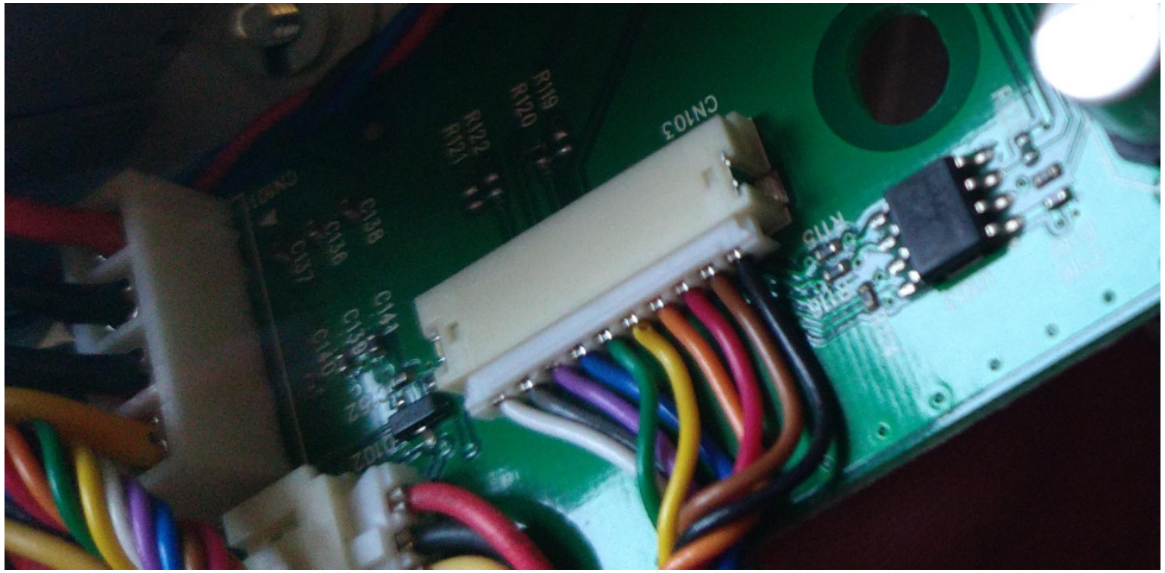
I did get an IBM LTO 6 lately and was having trouble setting it in automatic drive online Mode. After a lot of reading of the barely available documentation and lending an oscilloscope from a friend, I was able to put things together and get the drive to work without an library. Hopefully I am able to save someone on this planet some time and struggle by sharing my documentation.

In the first place I did get an RS-422-USB cable with an screw terminal, but due to the undocumented pinout of the JST-SH 10 Pin connector I wasn’t successful in establishing an communication (maybe just bad luck).

The second step was to search out the circuit board and check the components. There I found an chip from TI, which is similar to this one THVD1451. TI has an great documentation for its components! https://www.ti.com/product/THVD1451

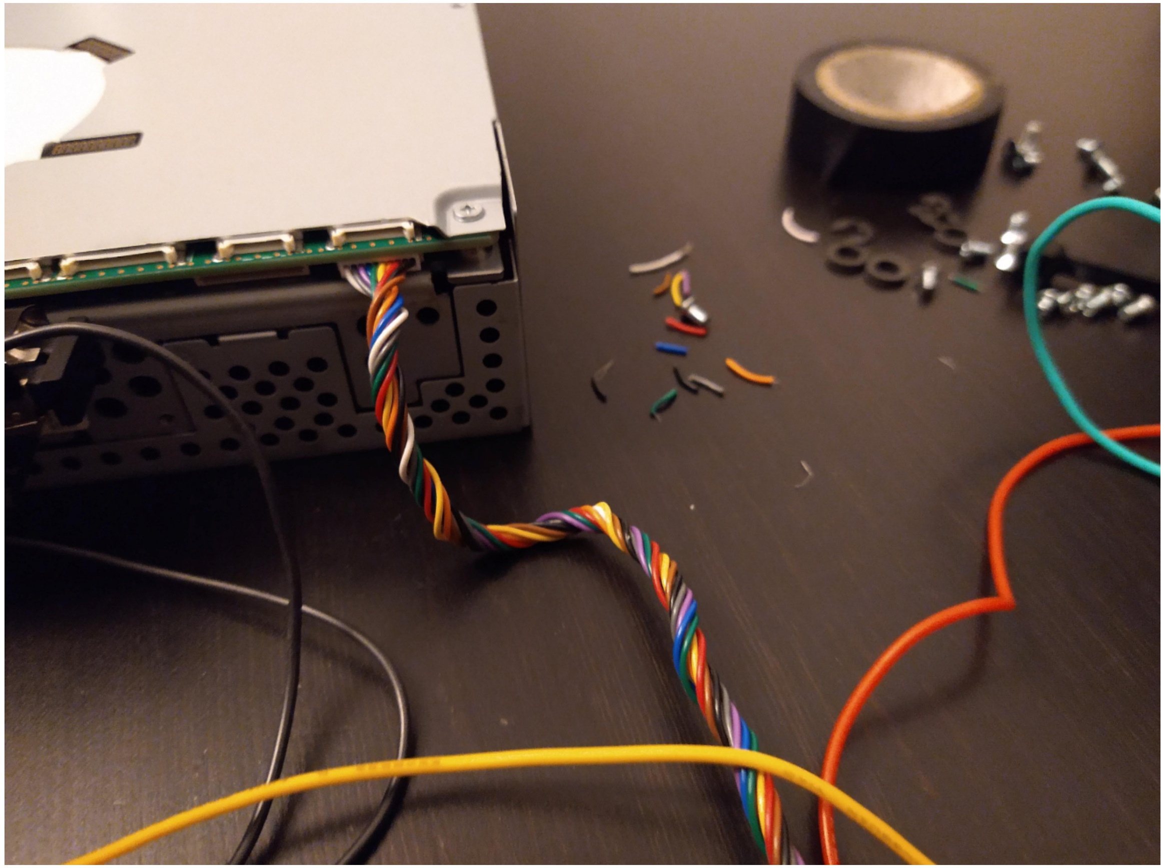

THVD1451 Device
8-Pin D Package (SOIC)
Top View
VCC
1
R
2
D
3
GND
4
8
A
7
B
6
Z
5
Y
Not to scale

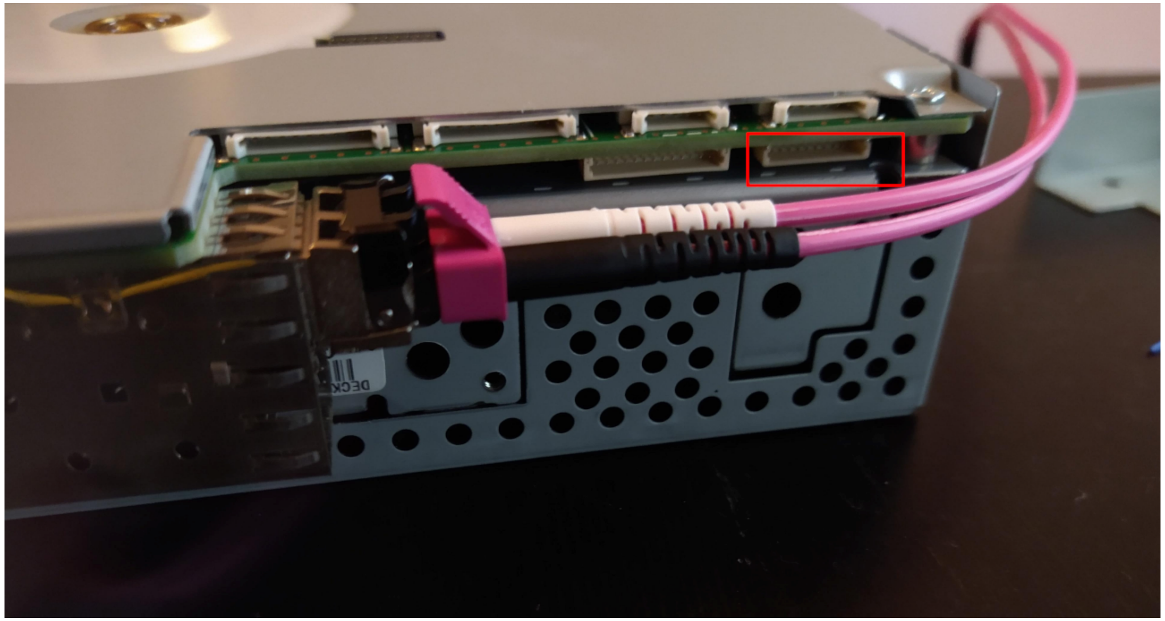

CN101
CN102
CN103
U103
R115
R116
R114
R118
R126
C116
C118
C119
C120
C123
C125
C136
C137
C138
C139
C140
C141
C152
R119
R120
R121
R122



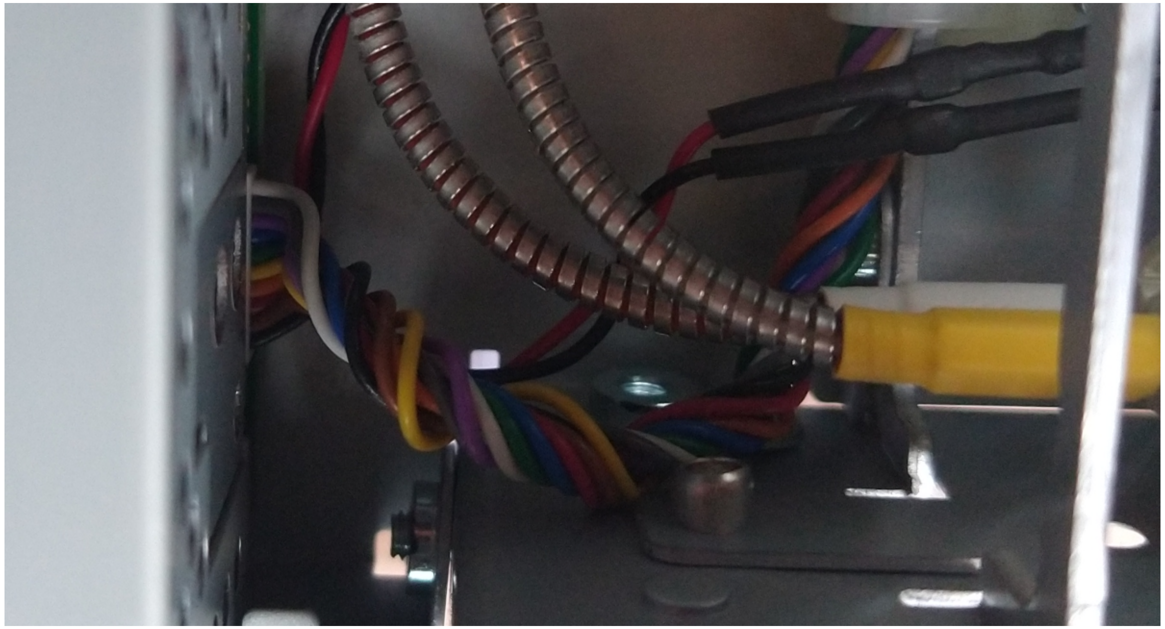
Perspective from the Drive Side (Issue #1)

(The drive is Top Down on these pictures, the Top is on the table)





(The drive is in the position bottom to table, perspective from above)



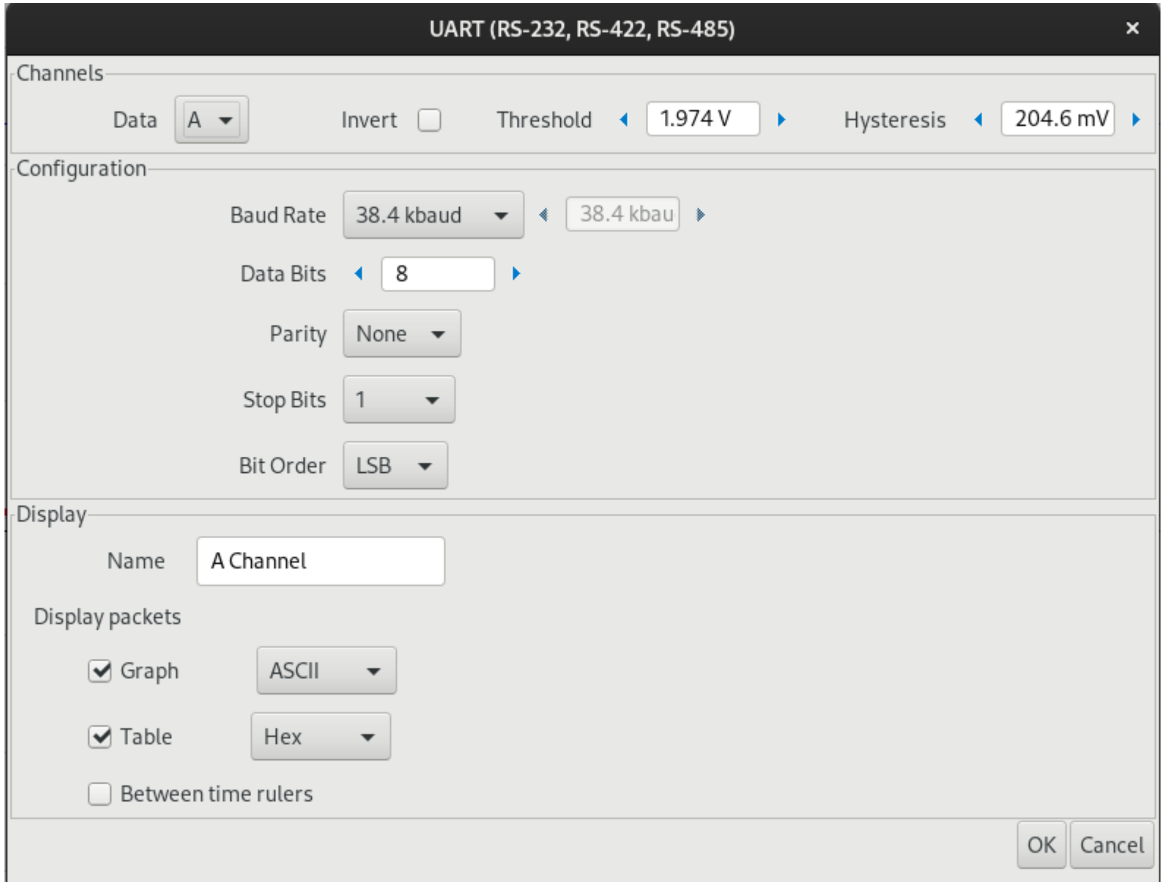
Cabeling colors are from top to bottom:

- white
- grey
- purple
- blue
- green
- yellow
- orange
- red
- brown
- black

Following the Electrical Pathways I was assuming the following cabling:

- Receive:
  - A → Yellow OR Orange
  - B → Orange OR Red
- Send:
  - Z → Red OR Brown
  - Y → Brown OR Black

With the help of the oscilloscope it was possible to evaluate, although it was the fist time for me utilizing one.



UART (RS-232, RS-422, RS-485)

Channels

Data: A

Invert: □

Threshold: 1.974 V

Hysteresis: 204.6 mV

Configuration

Baud Rate: 38.4 kbaud (38.4 kbau)

Data Bits: 8

Parity: None

Stop Bits: 1

Bit Order: LSB

Display

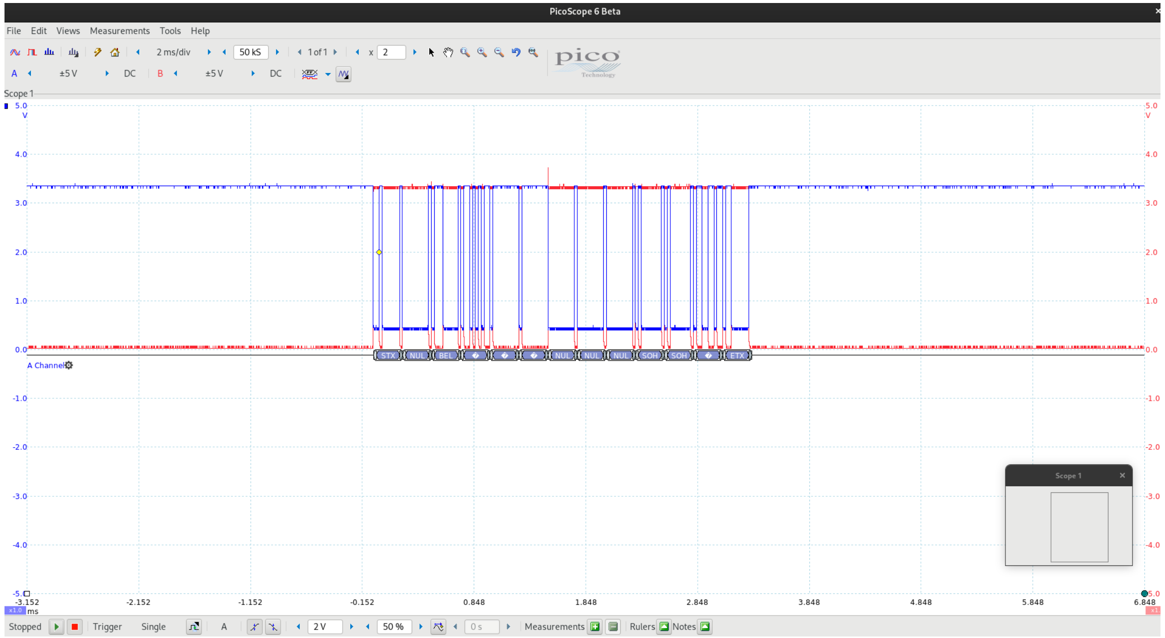
Name: A Channel

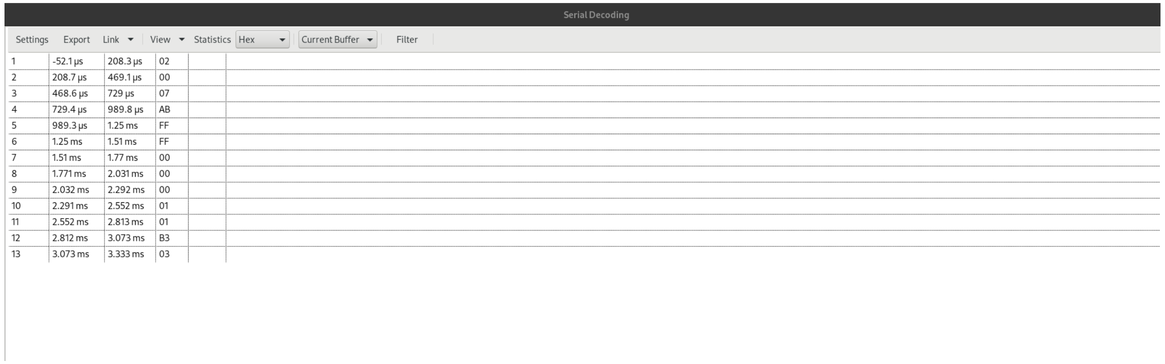
Display packets

☑ Graph: ASCII

☑ Table: Hex

□ Between time rulers





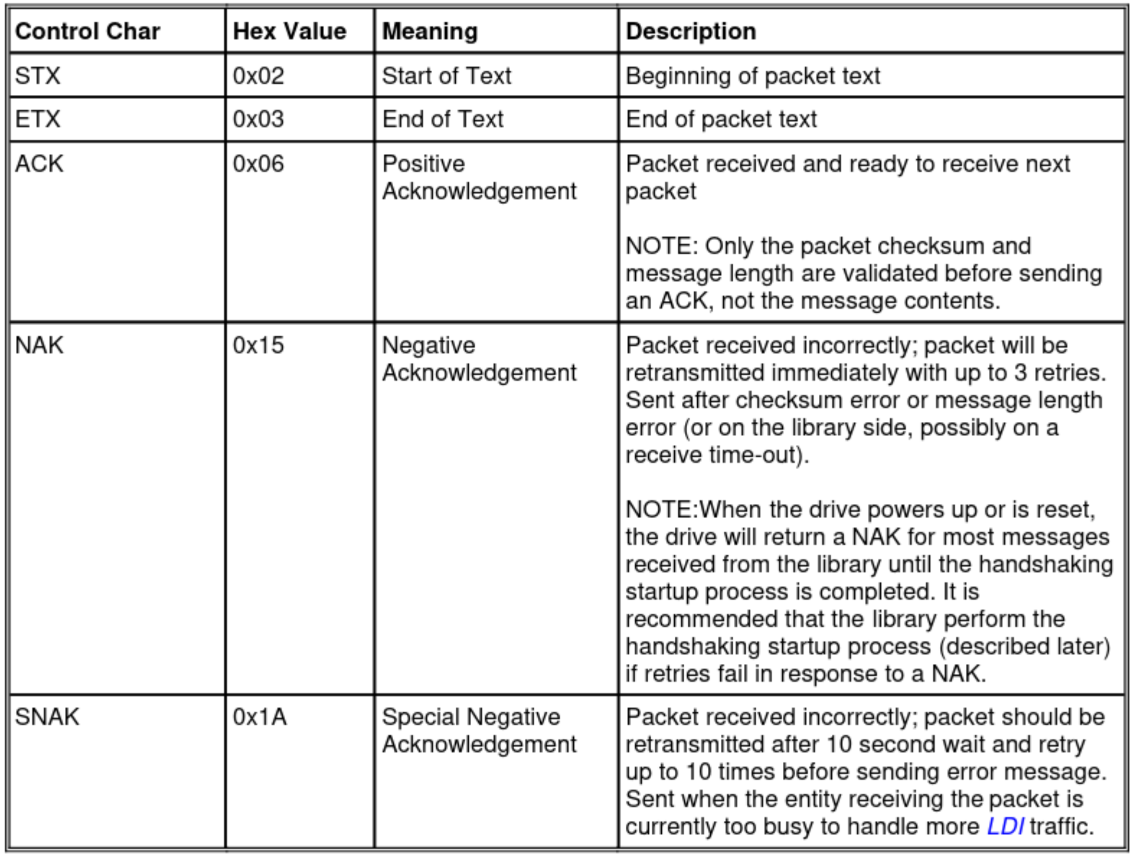
1	-52.1 µs	208.3 µs	02
2	208.7 µs	469.1 µs	00
3	468.6 µs	729 µs	07
4	729.4 µs	989.8 µs	AB
5	989.3 µs	1.25 ms	FF
6	1.25 ms	1.51 ms	FF
7	1.51 ms	1.77 ms	00
8	1.771 ms	2.031 ms	00
9	2.032 ms	2.292 ms	00
10	2.291 ms	2.552 ms	01
11	2.552 ms	2.813 ms	01
12	2.812 ms	3.073 ms	B3
13	3.073 ms	3.333 ms	03

Now I knew i was receiving signals on the YELLOW and the ORANGE Cables. The RS422 standard specifies differential signaling. In the picture above you can see the signals switching polarity which is causing an increased amplitude at the receiving side. I am not an electrical engineer and therefor can't go any deeper in detail.

But what was it I was receiving? There would come the IBM Library/Drive Interface Specification in handy. http://t10.org/ftp/t10/document.02/02-022r0.pdf

There are the Control Characters which did appear 0x02 and 0x03 for Start an End of Text.

Table 1, ASCII Control Characters



Control Char	Hex Value	Meaning	Description
STX	0x02	Start of Text	Beginning of packet text
ETX	0x03	End of Text	End of packet text
ACK	0x06	Positive Acknowledgement	Packet received and ready to receive next packet NOTE: Only the packet checksum and message length are validated before sending an ACK, not the message contents.
NAK	0x15	Negative Acknowledgement	Packet received incorrectly; packet will be retransmitted immediately with up to 3 retries. Sent after checksum error or message length error (or on the library side, possibly on a receive time-out). NOTE:When the drive powers up or is reset, the drive will return a NAK for most messages received from the library until the handshaking startup process is completed. It is recommended that the library perform the handshaking startup process (described later) if retries fail in response to a NAK.
SNAK	0x1A	Special Negative Acknowledgement	Packet received incorrectly; packet should be retransmitted after 10 second wait and retry up to 10 times before sending error message. Sent when the entity receiving the packet is currently too busy to handle more *LDI* traffic.

The middle Part seemed to be the Message for a Config_Request. To see them it is crucial to set the drive in non-polled Mode with the dip switches on the back. I have set them all to OFF.

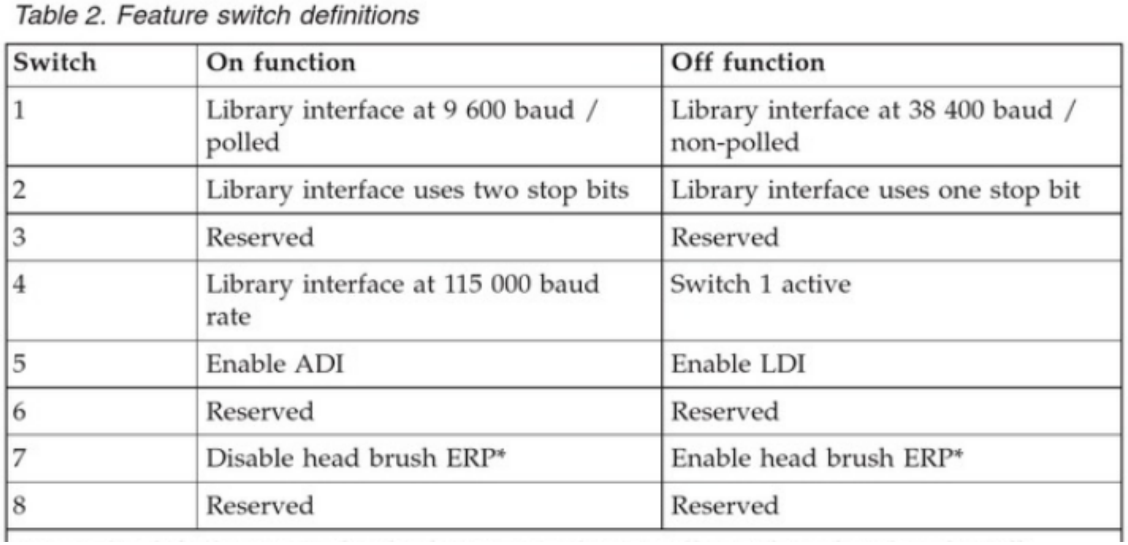
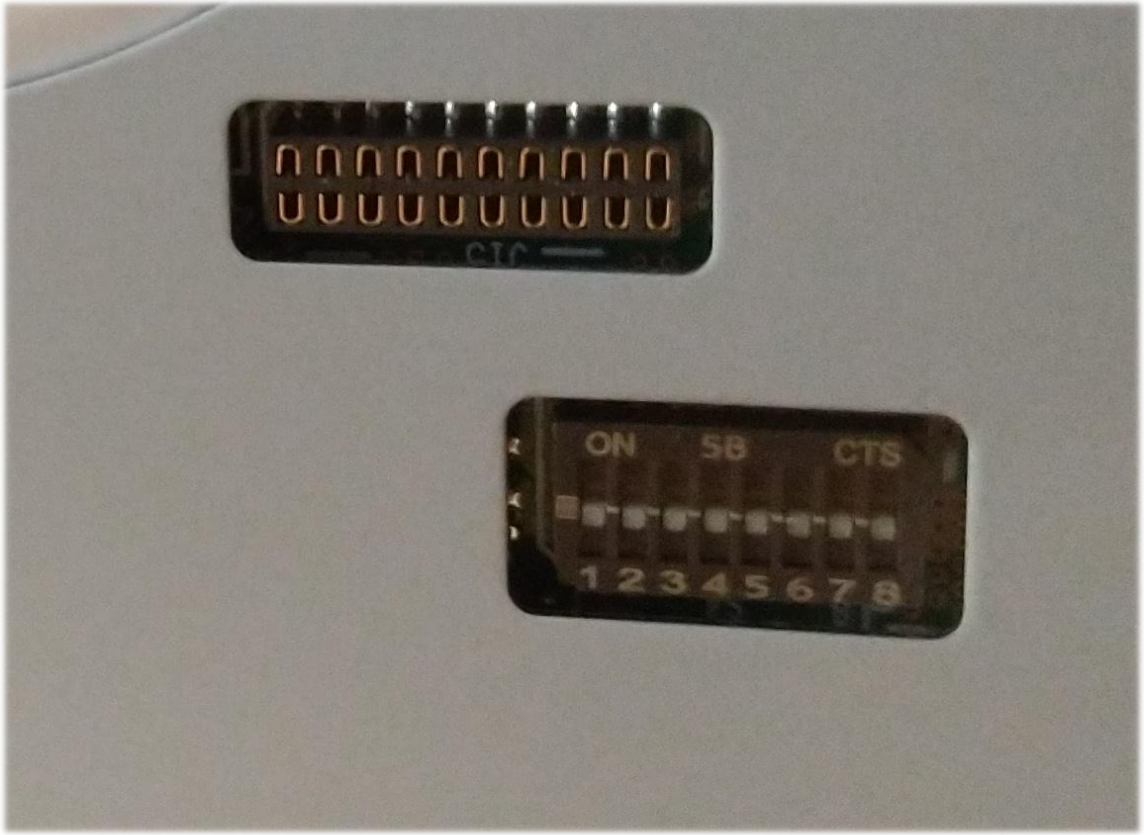


Table 2. Feature switch definitions

Switch	On function	Off function
1	Library interface at 9 600 baud / polled	Library interface at 38 400 baud / non-polled
2	Library interface uses two stop bits	Library interface uses one stop bit
3	Reserved	Reserved
4	Library interface at 115 000 baud rate	Switch 1 active
5	Enable ADI	Enable LDI
6	Reserved	Reserved
7	Disable head brush ERP*	Enable head brush ERP*
8	Reserved	Reserved



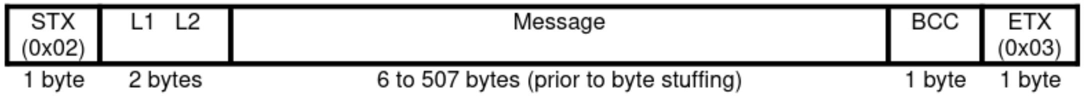
Config_Request (Drive-to-Library, Type=0xAB, Subtype=0x01)

When the drive powers up or is reset, the drive will return a NAK for any Two_Way message received from the library until the handshaking startup process is completed. If the interface is operating with a default of non-polled mode (settable via the drive feature switches), the drive will initiate the handshaking startup process by sending the Config_Request message. This message will be sent within 10 seconds after the drive powers up or is reset. *For polled mode of operation, see Scenario for Handshaking Startup Process (Library-*Initiated) *on page 30.*

Byte	Description
0	Msg Type (0xAB)
1	Tgt Addr (0xFF)
2 ... 5	Msg ID
6	Msg Subtype (0x01)

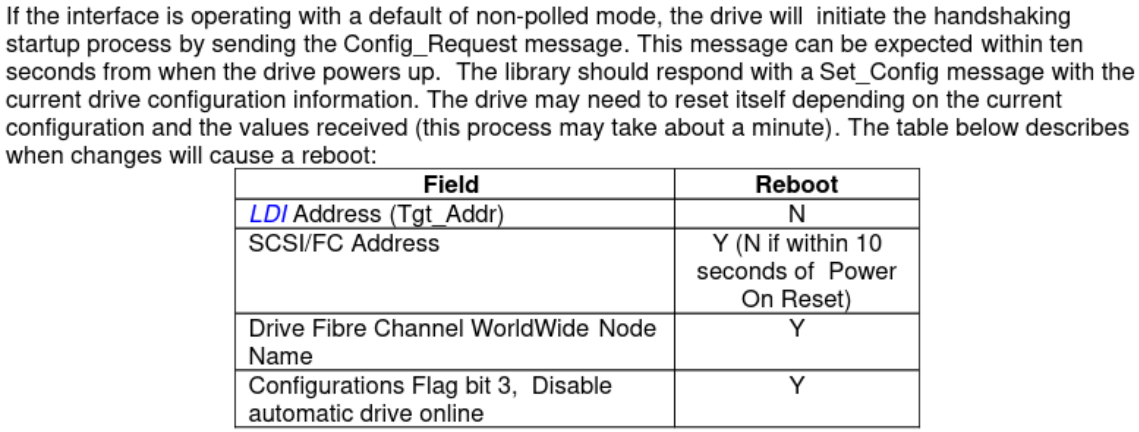
Packets does have the following format:

Figure 2, Packet Format



STX (0x02)	L1 L2	Message	BCC	ETX (0x03)
1 byte	2 bytes	6 to 507 bytes (prior to byte stuffing)	1 byte	1 byte

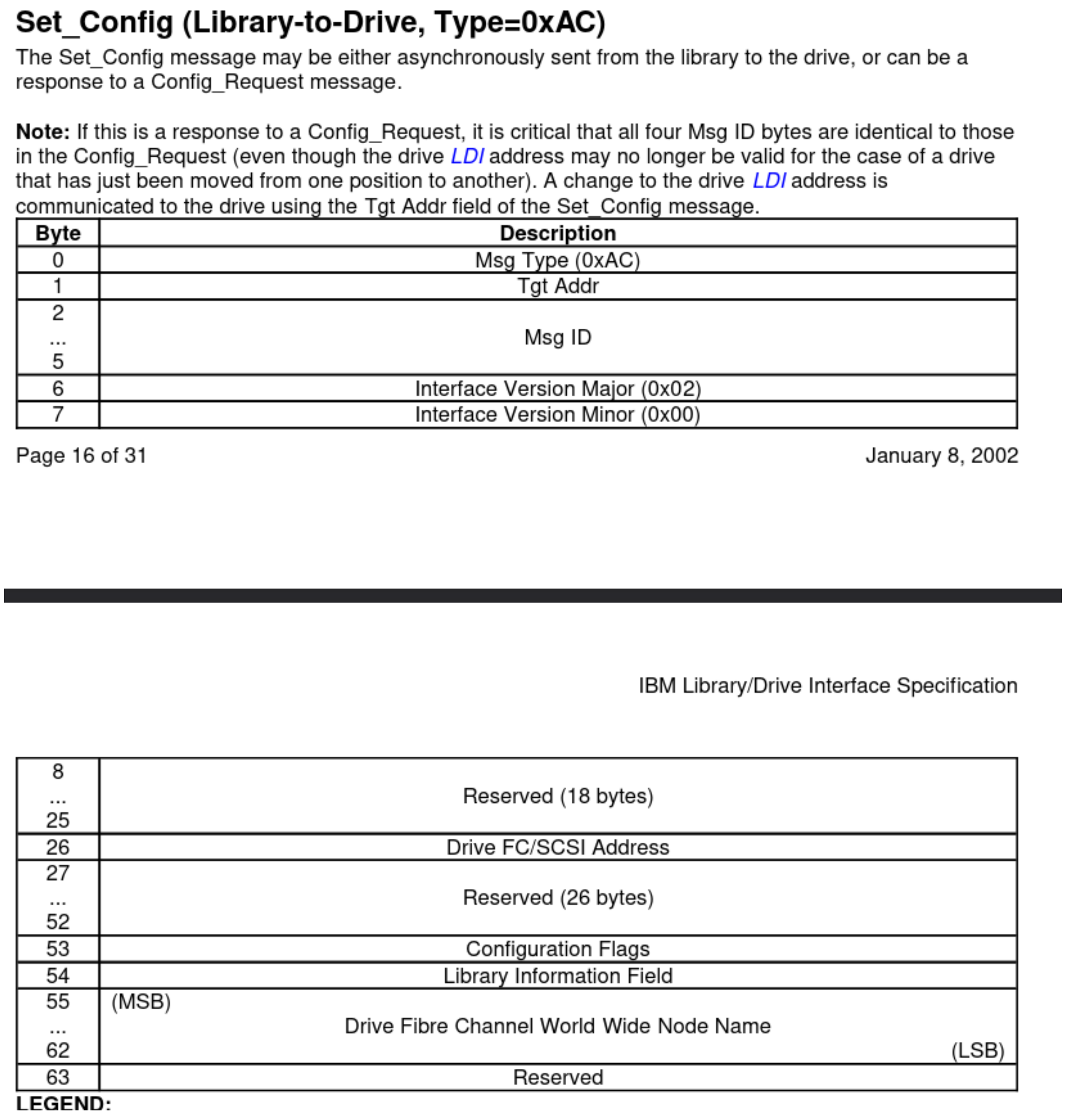
So this would make sense for an drive that was took out of an library. At page 30 is a description of the Interface Scenarios. The drive is operation in an **non-polled Mode** due to the **dip switch**. This means the drive is initiating the handshaking process and asking for a config.



If the interface is operating with a default of non-polled mode, the drive will initiate the handshaking startup process by sending the Config_Request message. This message can be expected within ten seconds from when the drive powers up. The library should respond with a Set_Config message with the current drive configuration information. The drive may need to reset itself depending on the current configuration and the values received (this process may take about a minute). The table below describes when changes will cause a reboot:

Field	Reboot
LDI Address (Tgt_Addr)	N
SCSI/FC Address	Y (N if within 10 seconds of Power On Reset)
Drive Fibre Channel WorldWide Node Name	Y
Configurations Flag bit 3, Disable automatic drive online	Y

To fulfill the handshake we need to “craft” an package. From Page 16 on is the description how an **Set_Config** message should look like.



Set_Config (Library-to-Drive, Type=0xAC)

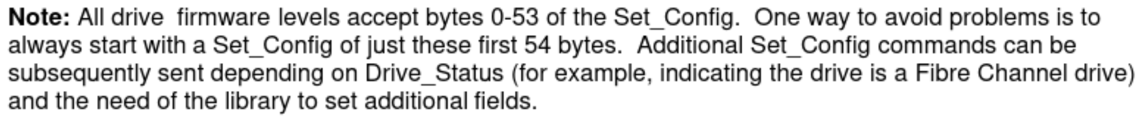
The Set_Config message may be either asynchronously sent from the library to the drive, or can be a response to a Config_Request message.

Note: If this is a response to a Config_Request, it is critical that all four Msg ID bytes are identical to those in the Config_Request (even though the drive *LDI* address may no longer be valid for the case of a drive that has just been moved from one position to another). A change to the drive *LDI* address is communicated to the drive using the Tgt Addr field of the Set_Config message.

Byte	Description
0	Msg Type (0xAC)
1	Tgt Addr
2 ... 5	Msg ID
6	Interface Version Major (0x02)
7	Interface Version Minor (0x00)
8 ... 25	Reserved (18 bytes)
26	Drive FC/SCSI Address
27 ... 52	Reserved (26 bytes)
53	Configuration Flags
54	Library Information Field
55 ... 62	(MSB) Drive Fibre Channel World Wide Node Name (LSB)
63	Reserved

LEGEND:

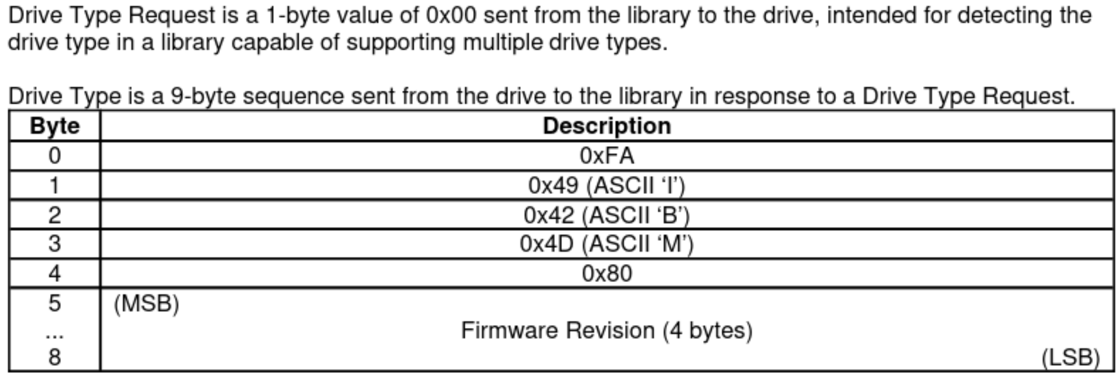
And a useful information was in the interface scenario on Page 30.



Note: All drive firmware levels accept bytes 0-53 of the Set_Config. One way to avoid problems is to always start with a Set_Config of just these first 54 bytes. Additional Set_Config commands can be subsequently sent depending on Drive_Status (for example, indicating the drive is a Fibre Channel drive) and the need of the library to set additional fields.

If we would just craft the package with the first 54 bytes, we would be firmware independent.

Still we do not know what the sending cables are. The good news is there are packets we can try to send to the library which are outside of the package protocol, meaning we don’t need start-text, length, checksum, end-text, acknowledgments, or byte stuffing. If we send just hex 00 to the drive, we should receive something with “I” “B” “M”.



Drive Type Request is a 1-byte value of 0x00 sent from the library to the drive, intended for detecting the drive type in a library capable of supporting multiple drive types.

Drive Type is a 9-byte sequence sent from the drive to the library in response to a Drive Type Request.

Byte	Description
0	0xFA
1	0x49 (ASCII ‘I’)
2	0x42 (ASCII ‘B’)
3	0x4D (ASCII ‘M’)
4	0x80
5 ... 8	(MSB) Firmware Revision (4 bytes) (LSB)

If you remember the lines I assumed as sending line have been:

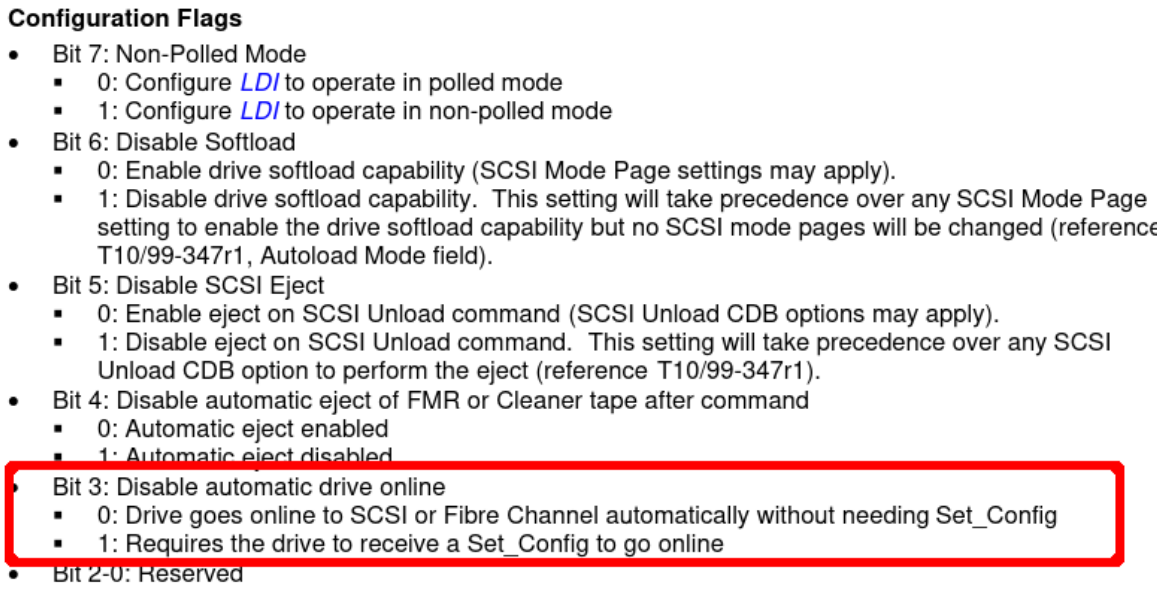
- Z → Red OR Brown
- Y → Brown or Black

It turn out to be:

- Z → Red

- Y → Brown
- Ground is Black.

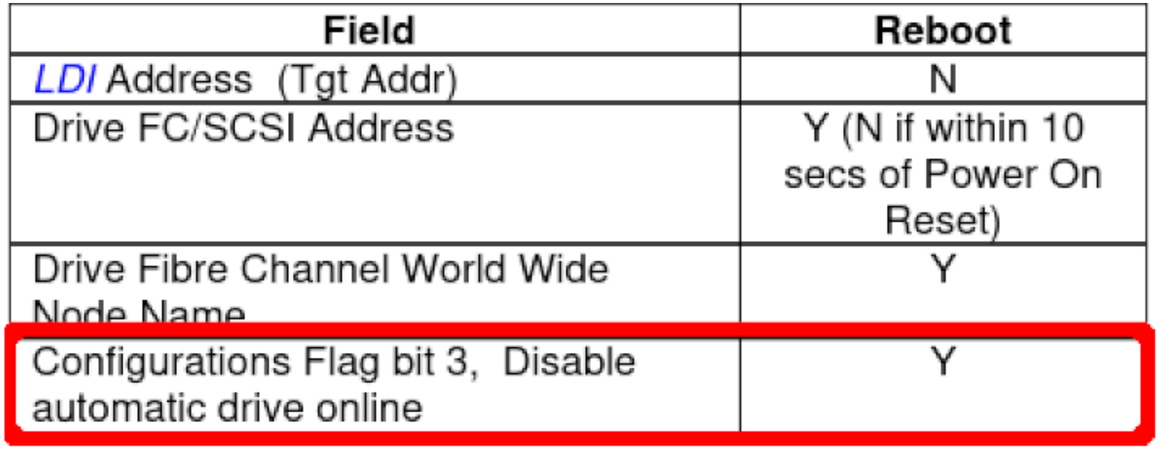
After connecting Red to TxD+, Brown to TxD- and Black to GND on our RS-422-USB Adapter, we are ready to send the Set_Config package to the drive. The setting we try to accomplish is to set **Bit 3 in Byte 53** to 0. This would allow the drive to raise the SCSI or Fiber Channel Ports without the need of receiving an Set_Config Package every time it powers up.



Configuration Flags

- Bit 7: Non-Polled Mode
  - 0: Configure *LDI* to operate in polled mode
  - 1: Configure *LDI* to operate in non-polled mode
- Bit 6: Disable Softload
  - 0: Enable drive softload capability (SCSI Mode Page settings may apply).
  - 1: Disable drive softload capability. This setting will take precedence over any SCSI Mode Page setting to enable the drive softload capability but no SCSI mode pages will be changed (reference T10/99-347r1, Autoload Mode field).
- Bit 5: Disable SCSI Eject
  - 0: Enable eject on SCSI Unload command (SCSI Unload CDB options may apply).
  - 1: Disable eject on SCSI Unload command. This setting will take precedence over any SCSI Unload CDB option to perform the eject (reference T10/99-347r1).
- Bit 4: Disable automatic eject of FMR or Cleaner tape after command
  - 0: Automatic eject enabled
  - 1: Automatic eject disabled
- Bit 3: Disable automatic drive online
  - 0: Drive goes online to SCSI or Fibre Channel automatically without needing Set_Config
  - 1: Requires the drive to receive a Set_Config to go online
- Bit 2-0: Reserved

If we where successful, the drive would send and Acknowledge **0x06** an reboot like stated in the Scenario for Startup Process.



Field	Reboot
LDI Address (Tgt Addr)	N
Drive FC/SCSI Address	Y (N if within 10 secs of Power On Reset)
Drive Fibre Channel World Wide Node Name	Y
Configurations Flag bit 3, Disable automatic drive online	Y

How to build the Package is described on the Pages 11 and 12 in the LDI Specification.

For transmitting the package I did use the following script:

```
#!/bin/bash
stty -F /dev/ttyUSB0 speed 38400 -cstopb -parenb -echo

while read -r line
do
        echo -en “$line” > /dev/ttyUSB0
done < “$1”
```

Save this as echo_hex and change the mode to execute

```
chmod +x ./echo_hex
```

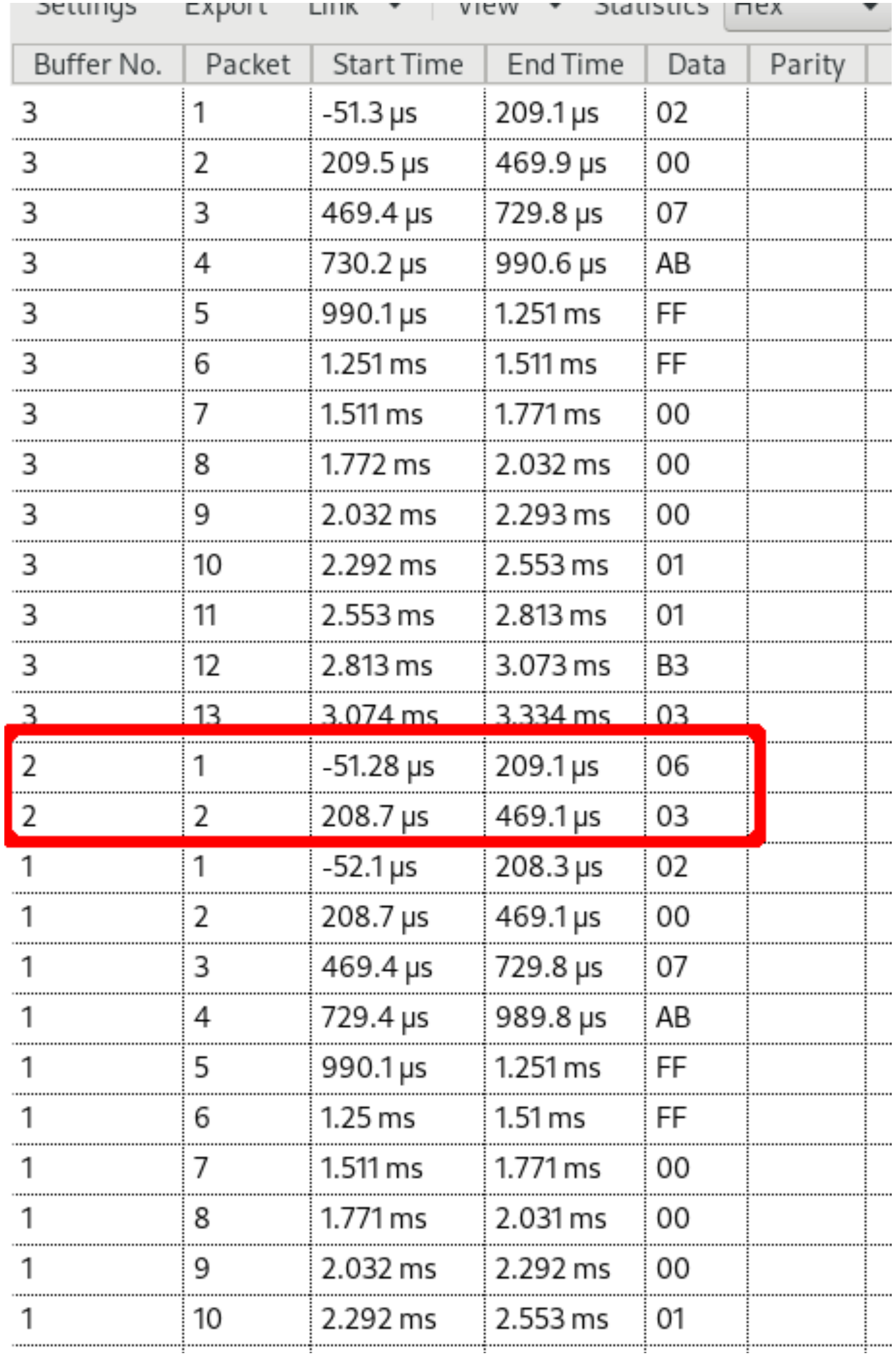
Then we need an file with the hex code. Just name it code for example.

```
\x02\x00\x36\xAC\x01\x00\x00\x00\x01\xFF\xF2\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\xE6\x03
```

Then we can issue the command

```
./echo_hex code
```

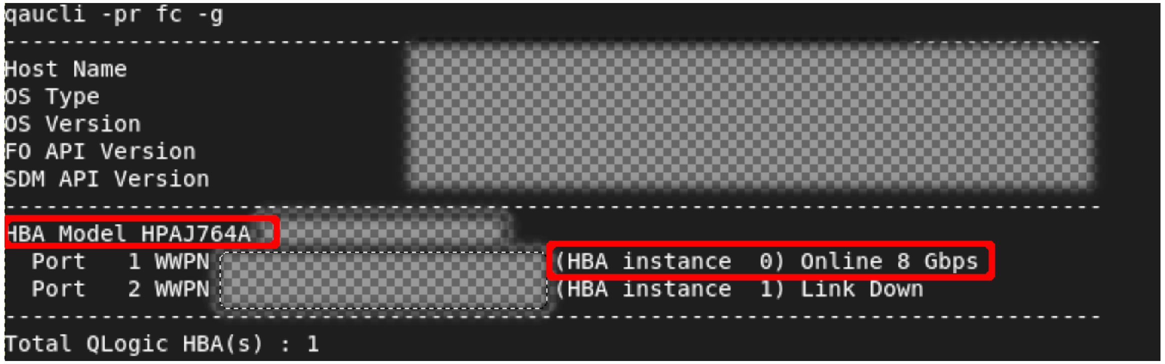
and see the Packages 0x06 0x03, for ACK and ETX on the receiving side.



Buffer No.	Packet	Start Time	End Time	Data	Parity
3	1	-51.3 µs	209.1 µs	02	
3	2	209.5 µs	469.9 µs	00	
3	3	469.4 µs	729.8 µs	07	
3	4	730.2 µs	990.6 µs	AB	
3	5	990.1 µs	1.251 ms	FF	
3	6	1.251 ms	1.511 ms	FF	
3	7	1.511 ms	1.771 ms	00	
3	8	1.772 ms	2.032 ms	00	
3	9	2.032 ms	2.293 ms	00	
3	10	2.292 ms	2.553 ms	01	
3	11	2.553 ms	2.813 ms	01	
3	12	2.813 ms	3.073 ms	B3	
3	13	3.074 ms	3.334 ms	03	
2	1	-51.28 µs	209.1 µs	06	
2	2	208.7 µs	469.1 µs	03	
1	1	-52.1 µs	208.3 µs	02	
1	2	208.7 µs	469.1 µs	00	
1	3	469.4 µs	729.8 µs	07	
1	4	729.4 µs	989.8 µs	AB	
1	5	990.1 µs	1.251 ms	FF	
1	6	1.25 ms	1.51 ms	FF	
1	7	1.511 ms	1.771 ms	00	
1	8	1.771 ms	2.031 ms	00	
1	9	2.032 ms	2.292 ms	00	
1	10	2.292 ms	2.553 ms	01	

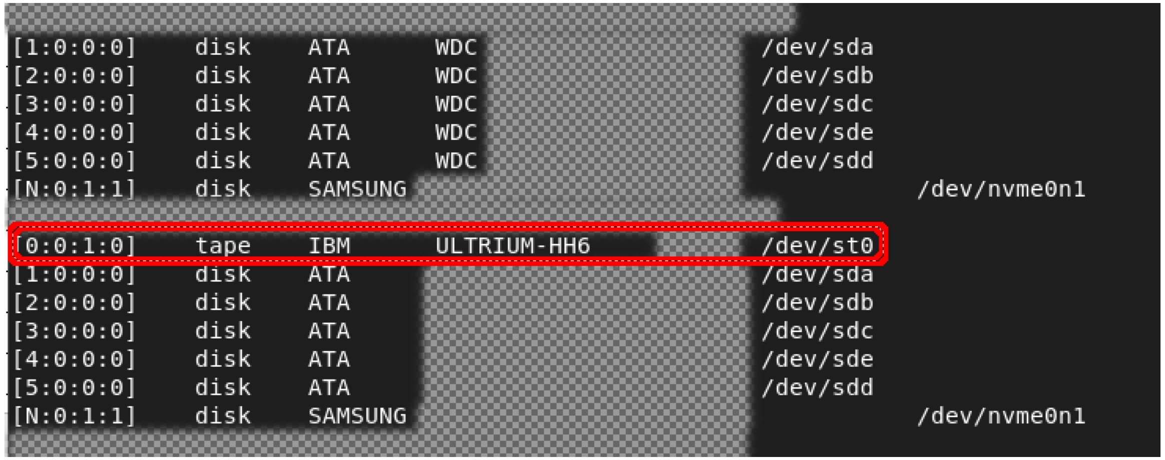
1	11	2.552 ms	2.813 ms	01		
1	12	2.813 ms	3.073 ms	B3		
1	13	3.073 ms	3.333 ms	03		

Now the drive is rebooting and we can see the Link online. This qaucli is a command from qlogic (MARVELL) which in my case produced the HP Branded FC Card.



```
qaucli -pr fc -g
-----------------------------[redacted]
Host Name
OS Type
OS Version
FO API Version
SDM API Version
--------------------------------------------------------------------------------
HBA Model HPAJ764A [redacted]
  Port   1 WWPN [redacted] (HBA instance  0) Online 8 Gbps
  Port   2 WWPN            (HBA instance  1) Link Down
--------------------------------------------------------------------------------
Total QLogic HBA(s) : 1
```

Also lsscsi is outputting the Tape now.



```
[redacted]
[1:0:0:0]    disk    ATA      WDC  [redacted]         /dev/sda
[2:0:0:0]    disk    ATA      WDC                     /dev/sdb
[3:0:0:0]    disk    ATA      WDC                     /dev/sdc
[4:0:0:0]    disk    ATA      WDC                     /dev/sde
[5:0:0:0]    disk    ATA      WDC                     /dev/sdd
[N:0:1:1]    disk    SAMSUNG                                     /dev/nvme0n1
[redacted]
[0:0:1:0]    tape    IBM      ULTRIUM-HH6  [redacted] /dev/st0
[1:0:0:0]    disk    ATA      [redacted]              /dev/sda
[2:0:0:0]    disk    ATA                              /dev/sdb
[3:0:0:0]    disk    ATA                              /dev/sdc
[4:0:0:0]    disk    ATA                              /dev/sde
[5:0:0:0]    disk    ATA                              /dev/sdd
[N:0:1:1]    disk    SAMSUNG                                     /dev/nvme0n1
```

SUMMARY

- Set all DIP Switches on the drive to off.
- Connect the RS-422-USB Adapter as follows:
  - TxD+ → Red
  - TxD- → Brown
  - RxD+ → Yellow
  - RxD- → Orange
  - GND → Black
- Send the Package to the drive: `\x02\x00\x36\xAC\x01\x00\x00\x00\x01\xFF\xF2\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xE6\x03`
- Drive restarts and goes online to SCSI or Fibre Channel automatically.