

Table of Contents

| | |
|--|---|
| Getting started..... | 3 |
| Assembly..... | 3 |
| Supplying power..... | 3 |
| Flashing the firmware..... | 3 |
| Configuring the interface..... | 4 |
| Formatting the disk..... | 4 |
| Using your hard drive..... | 5 |
| Disk commands..... | 5 |
| Changing directories..... | 5 |
| Adding and removing files and directories..... | 5 |
| Renaming files and directories..... | 6 |
| Misc commands..... | 6 |
| Other functions..... | 6 |
| Swapping ID..... | 6 |
| Standby mode..... | 6 |
| Resources..... | 6 |

Getting started

Assembly

I suggest that you start the build by adding the LM2596 regulator and all it's peripheral components (see the area around the LM2596 in the schematic) and verify that you can power the PCB with 12V and get 5V out from the regulator. If you make a mistake here there's a good chance that the regulator outputs 12V and that would fry other components on the board if they were installed. You should have 5V on both sides of the inductor L1 when everything is right.

Because of component shortages ,the PCB is made to support a couple of different 5V regulators – a fixed 5V LM2596 regulator or the cheaper adjustable LM2596 which requires a couple of extra peripheral components (R401 and C404) and a 3.1kΩ resistor at R402 to keep the voltage at 5V.

If you go with a fixed 5V regulator your should not mount R401 and C404 and you should install a 0Ω resistor or a little jumper wire instead of a resistor at R402.

When you have an acceptable 5V output, add the 3.3V and 1.8V regulators and capacitors C501, C502, C601, C602 and verify the 1.8V and 3.3V by measuring the center pin on each regulator.

When all voltages are verified you can add the rest of the SMD parts including the SATA connector and then the through-hole parts.

Supplying power

The controller needs to be powered by 12V DC through a barrel jack with +12V on the center pin or by soldering wires to the foot print where the barrel jack would be.

For SSD drives, a 1A power supply should be more than enough but it might be a bit on the low side for a 3.5" 7200 RPM disk so a 2A PSU might be a safer choice in that case.

The jumper JP1 can be used to install a power switch, a header and jumper or a permanent solder jumper. Opening the jumper removes all power to the board.

Flashing the firmware

The firmware to program the ATmega with is named nh4lite0.bin and can be found in the ajnnh057.zip archive under software in the NLQHD-SATA GitHub repository at <https://github.com/RetroNynjah/NLQHD-SATA> or at <http://nlq.de>

The following AVR fuses should be used.

| | |
|-------|------|
| LFUSE | \$FF |
| HFUSE | \$D9 |
| EFUSE | \$FD |

There are several ways to program an ATmega but the below example is using AVRDUDE. Setting the fuses and flashing the firmware on an Atmega1284 at the same time can be done like this.

```
avrdude -C avrdude.conf -v -p atmega1284 -c usbasp -U lfuse:w:0xff:m -U hfuse:w:0xd9:m -U efuse:w:0xfd:m -Uflash:w:nh4lite0.bin
```

You will have to adjust the command to match your programmer and change your AVR device if you are using an Atmega1284P. The fuses are the same for 1284P.

Configuring the interface

The device configuration is stored in the EEPROM of the ATmega1284 and needs to be set after programming the firmware. To configure the device, connect it to a C64 or C128 and run the configuration program.

The configuration program can be found on the nlqhd-057-lite.d64 and nlqhd-057-lite.d81 disks. The program is called C-NLQ-CAL&CON271

When asked if you are using NLQHD-lite hardware the answer is Yes.

Go through the configuration wizard and choose to configure each option. The default choices are usually fine but you may want to select a device ID that suits your environment.

When you are back at the configuration menu, select M to copy messages to the ATmega1284 EEPROM. This transfer status message texts and drive identification string to the EEPROM. Without this the drive will not report any text along with the error codes when querying the drive error channel for status and it won't present itself properly if you query the error channel directly after drive reset. You can use the @ command to query the drive error channel from JiffyDOS to make sure that it's reporting status correctly.

You don't need to run any calibration with the NLQHD-SATA because it is using an external crystal and therefore it's also not a problem to use JiffyDOS while configuring the device despite the warnings in the configuration program that speeders should be disabled until the oscillator has been calibrated.

Formatting the disk

You can use the program C-NLQ-PAR&FOR232 to partition and format your disk using FAT32 if it's not already partitioned and formatted. Using this tool for partitioning a larger disk ensures that the partition size is exactly the maximum supported size.

Using your hard drive

Disk commands

The below command examples assumes JiffyDOS kernel on the computer. To issue the commands without JiffyDOS you would have to use a command like OPEN15,8,15,"CD:GAMES" instead of just @CD:GAMES.

Changing directories

Move to directory

```
@CD:GAMES
```

Go up one directory level

```
@CD<= (PETSCII left arrow)
```

Go back to the root directory

```
@CD//
```

All commands that target file or folder names that includes spaces must be within quotes.

Examples:

Move to directory

```
@"CD:MY GAMES"
```

Adding and removing files and directories

Create new directory

```
@"MD:MY APPS"
```

Remove directory (must be empty)

```
@"RD:MY OLD DIR"
```

Delete (scratch) a file

```
@"S:MY OLD FILE"
```

Delete (scratch) all files in current directory with names beginning with MY

```
@"S:MY*"
```

Delete (scratch) all files in current directory

```
@S:*
```

Renaming files and directories

Rename a file or directory

```
@"R:NEW FILENAME=OLD FILENAME"
```

Misc commands

Turn off device (press reset switch to wake up again)

```
@AOF
```

Other functions

Swapping ID

It's possible to temporarily swap device ID with another device that has device ID 8 or 9. This makes it possible to run software that assumes that a disk drive must have ID 8 or 9.

To swap ID, press the middle button (left button if there are only two buttons) eight times. The busy light will start flashing series of eight flashes. Now press the right button to execute the device change. Pressing the middle button nine times followed by a right button press swaps ID with device 9.

Example: The NLQHD is device 30 and a 1541 disk drive is device 8. After the ID swap the 1541 is device 30 and the NLQHD is device 8.

The NLQHD can now be used as if the current folder was a disk inserted into device 8.

Standby mode

The device can be put into standby by holding the right button for 4 seconds or more. When the device is in standby it's device ID is removed from the IEC bus and can be used by other devices.

To wake the device from standby you can press the middle button on the front panel or the reset button at the back of the interface.

Resources

NLQHD-SATA GitHub repository

<https://github.com/RetroNynjah/NLQHD-SATA>

NLQ web site

<http://nlq.de>

JiffyDOS web site

<http://www.jiffydos.com>

