



Test Manual

Version : 0.1

Date : 28 / 08 / 2018

Table of Contents

Table of Contents	2
Product overview	3
Hardware Description	4
Board Specifications	4
Block Diagram	5
Helios4 Carrier Board	5
A388 System-On-Chip	5
Interfaces List	6
Schematics	8
Operating Manual	9
Peripheral Kit	9
Software Install	12
Writing an image to a microSD Card	12
System Setup	13
Test Application	15
Network Config and Credentials	15
Connect to Helios4	15
Test Suite Usage	17
Launch Application	17
Run All tests	17
Run SATA test	18
Run MicroSD test	19
Run USB test	19
Display Status	20
Troubleshooting	21
Troubleshooting table	21
Serial Console Access	23
Under Windows	23
Under Linux (via Terminal)	25
Troubleshooting commands	26
Check all drives are detected	26
Check IP address is correct	26
Check correct System Version is running	27
Boot Screen	28
U-Boot	28
Linux	30

Product overview

Helios4 is a powerful ARM based board specially designed for Network Attached Storage. It harnesses its processing capabilities from the **ARMADA 38x-MicroSoM** from [SolidRun](#).



The **Marvell ARMADA® 388** is a robust and energy-efficient System on Chip (SoC) with a collection of high speed interfaces especially suited for headless data processing, networking and storage. This Dual-core ARM Cortex A9 CPU clocked at 1.6 Ghz, paired with 2GB of ECC memory, integrates a Cryptographic and XOR DMA engines to provide the best level of performance and reliability for NAS functionalities.

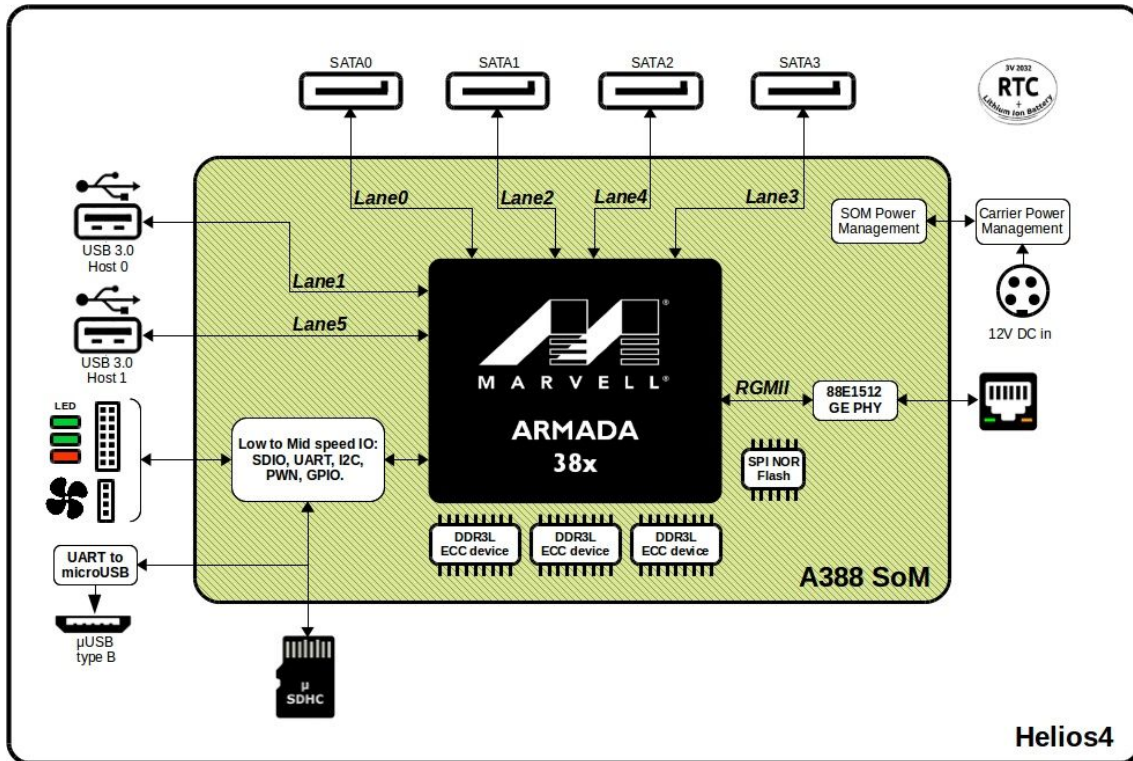
Hardware Description

Board Specifications

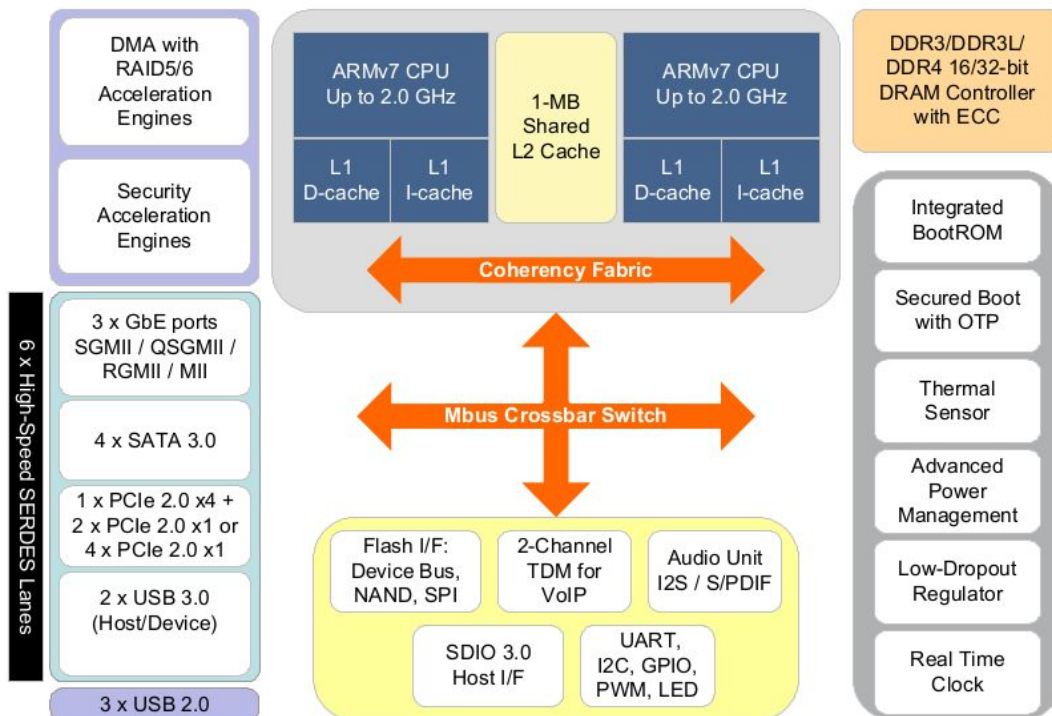
CPU Model	Marvell Armada 388 (88F6828) ARM Cortex-A9
CPU Architecture	ARMv7 32-bit
CPU Frequency	Dual Core 1.6 Ghz
Additional Features	- RAID Acceleration Engines - Security Acceleration Engines - Wake-on-LAN
System Memory	2GB DDR3L ECC
SATA 3.0 Ports	4
Max Raw Capacity	48 TB (12 TB drive x 4)
GbE LAN Port	1
USB 3.0	2
microSD (SDIO 3.0)	1
GPIO	12
I2C	1
UART	1 (via onboard Micro-USB converter)
Boot Mode Selector	- SPI - SD Card - UART - SATA
SPI NOR Flash	32Mbit onboard
PWM FAN	2
DC input	12V / 8A

Block Diagram

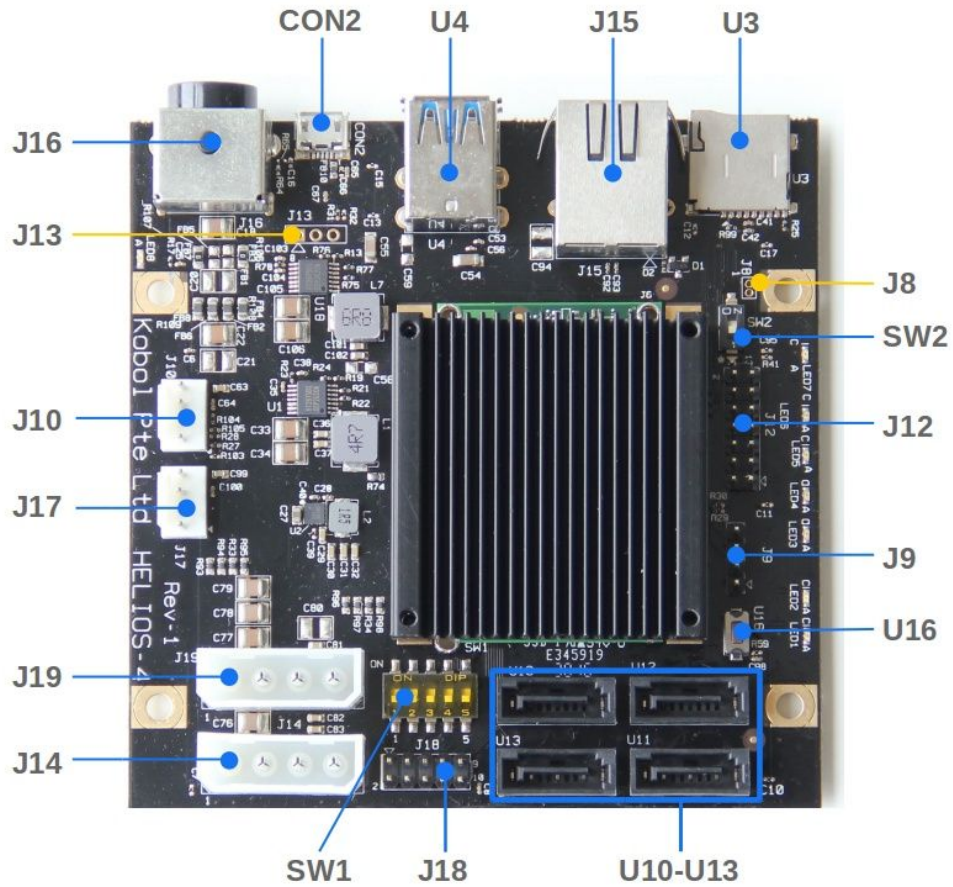
Helios4 Carrier Board



A388 System-On-Chip



Interfaces List



Name	Peripheral Type	Connector Type	Details
CON2	Serial port	Micro-USB Connector	Via onboard FTDI USB-to-UART0 bridge
J8	RC Battery	Not populated	External battery source for RTC clock
J9	I2C	4x1 Pin Male Header	I2C Channel 1
J10	Fan	4x1 Pin Male Header	PWM and RPM support

J12	GPIO	7x2 Pin Male Header	GPIO configurable as input or output Via IO Expander on I2C Channel 0
J13	<i>Serial port</i>	<i>Not populated</i>	<i>SoM UART0 interface</i>
J14	HDD Power	Molex 4-Pin Female	Rated for 2x HDD
J15	LAN	RJ45	Gigabit Ethernet
J16	DC connector	Kycon 4-Pin Mini-DIN	DC input 12V / 8A
J17	Fan	4x1 Pin Male Header	PWM and RPM support
J18	LED Panel	5x2 Pin Male Header	Expansion port of on-board LED
J19	HDD Power	Molex 4-Pin Female	Rated for 2x HDD
SW1	Boot Mode	Dip Switch	Boot mode selector : SPI,MMC,UART,SATA
SW2	LED Mode	Dip Switch	LED mode selection : board or expansion panel
U3	microSD	Push-Push card connector	Support SDHC and SDXC
U4	USB 3.0	Dual Port USB3.0	Type A
U16	Reset	Push Button	CPU Reset
U10	SATA	SATA 3.0	Port 0 (SATA1)
U11	SATA	SATA 3.0	Port 1 (SATA2)
U12	SATA	SATA 3.0	Port 2 (SATA3)
U13	SATA	SATA 3.0	Port 3 (SATA4)

Schematics

- Carrier Board Schematics :
https://wiki.kobol.io/files/carrier/Helios4_Schematics_R1-1.pdf
- System-On-Module Schematics :
<https://wiki.kobol.io/files/som/a38x-microsom-schematics-simplified-rev2.00.pdf>

Operating Manual

This section explains how to setup and operate Helios4 in order to exercise all its high speed interfaces.

Helios4 exposes the following high speed interfaces :

- SATA 3.0 (4x)
- USB 3.0 (2x)
- Ethernet
- SDIO

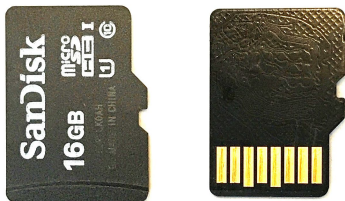
Peripheral Kit

The following peripheral kit is required to exercise all the interfaces listed above.

MicroSD Card

SanDisk microSD card UHS Class 10 - 16GB

Model : SDCSDQAD-016G



Quantity : 1

SATA 3.0 SSD Drive

WD GREEN SATA SSD - 120GB

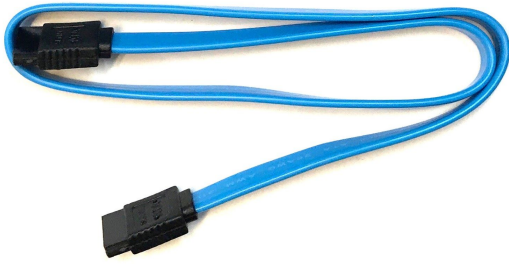
Model : WDS120G2G0A-00JH30



Quantity : 4

SATA 3.0 Data Cable (Shielded)

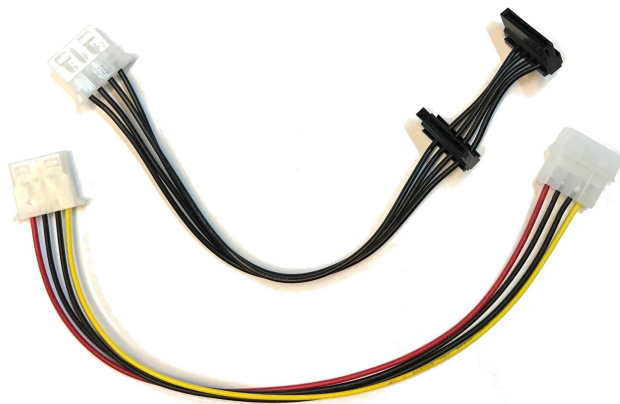
Length : 50 cm



Quantity : 4

SATA Power Cable + Extension

Length : total 50 cm



Quantity : 2

USB 3.0 Flash Drive

SanDisk Ultra Flair - 32GB

Model : SDCZ73-032G



Quantity : 2

USB 3.0 Extension Cable (Shielded)

Length : 60cm



Quantity : 2

Ethernet Cable - Cat. 6

Length : 3m



Quantity : 1

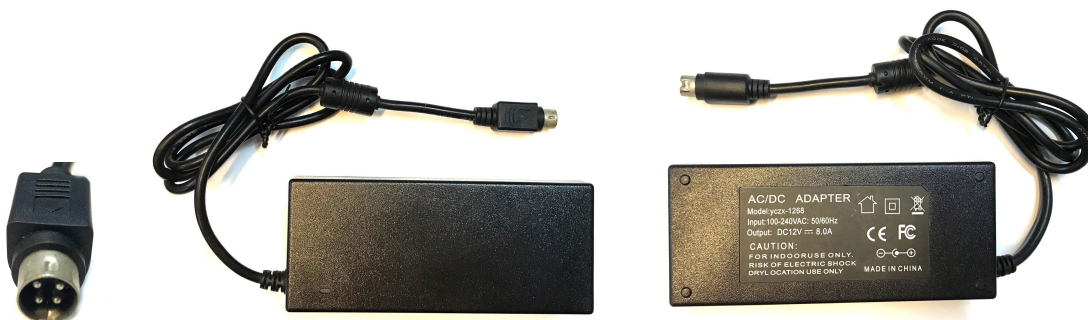
AC / DC Power Adapter

Input : 100-240VAC: 50/60Hz

Output : 12V - 8.0A

DC Connector type : Male 4-pin "Kycon"

Cable Length : 1m



Software Install

Helios4 is supported by [Armbian](#) Linux distro which provides Debian and Ubuntu based images. The latest release are Armbian Stretch (Debian) and Armbian Bionic (Ubuntu) with Linux Kernel 4.14.

A customized version of Armbian Stretch that includes the Helios4 **Test Application** is provided for the purpose of running the tests described in this manual.

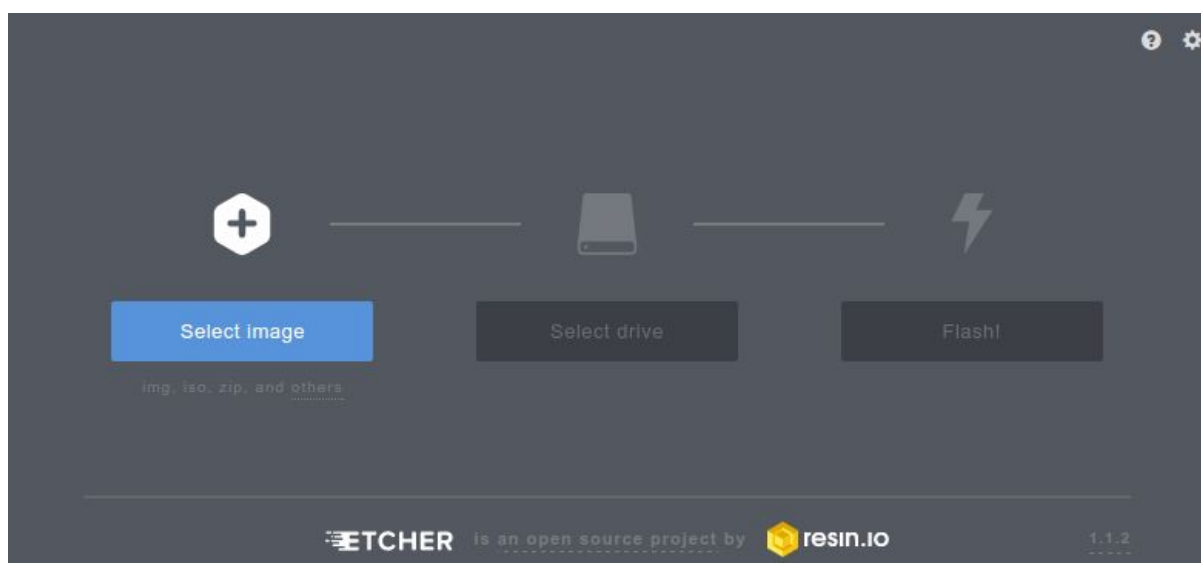
If no pre-installed microSD cards were provided you can download the image [here](#) and follow the below instruction on how to write an image on a microSD Card.

Note : Archives can be uncompressed with 7-Zip on Windows, Keka on OS X and 7z on Linux (apt-get install p7zip-full). RAW images can be written with Etcher (all OS).

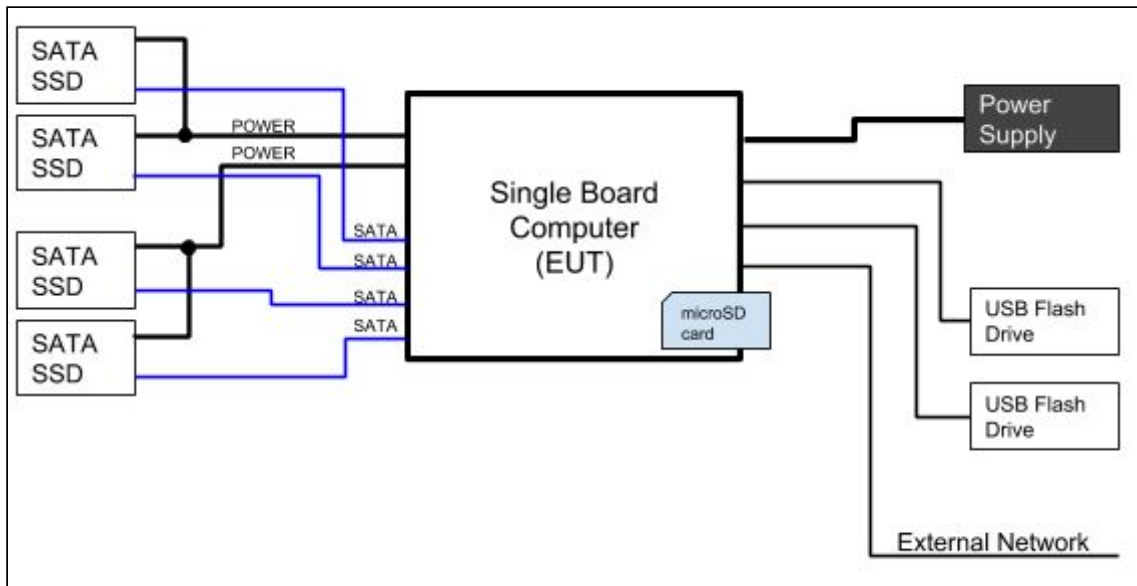
Writing an image to a microSD Card

Etcher is a graphical SD card writing tool that works on Mac OS, Linux and Windows, and is the easiest option for most users. To write your image with Etcher:

1. [Download Etcher](#) and install it on your computer.
2. Insert the microSD Card inside your SD card reader (microSD to SD adapter might be needed).
3. Open Etcher and select from your local storage the Helios4 .img.xz file you have downloaded.
4. Select the microSD Card you wish to write your image to.
5. Review your selections and click 'Flash!' to begin writing data to the microSD Card.



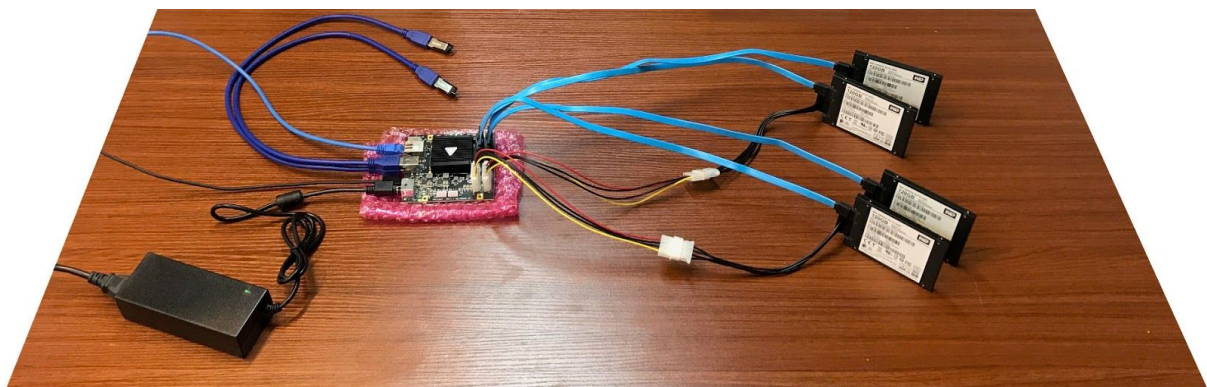
System Setup

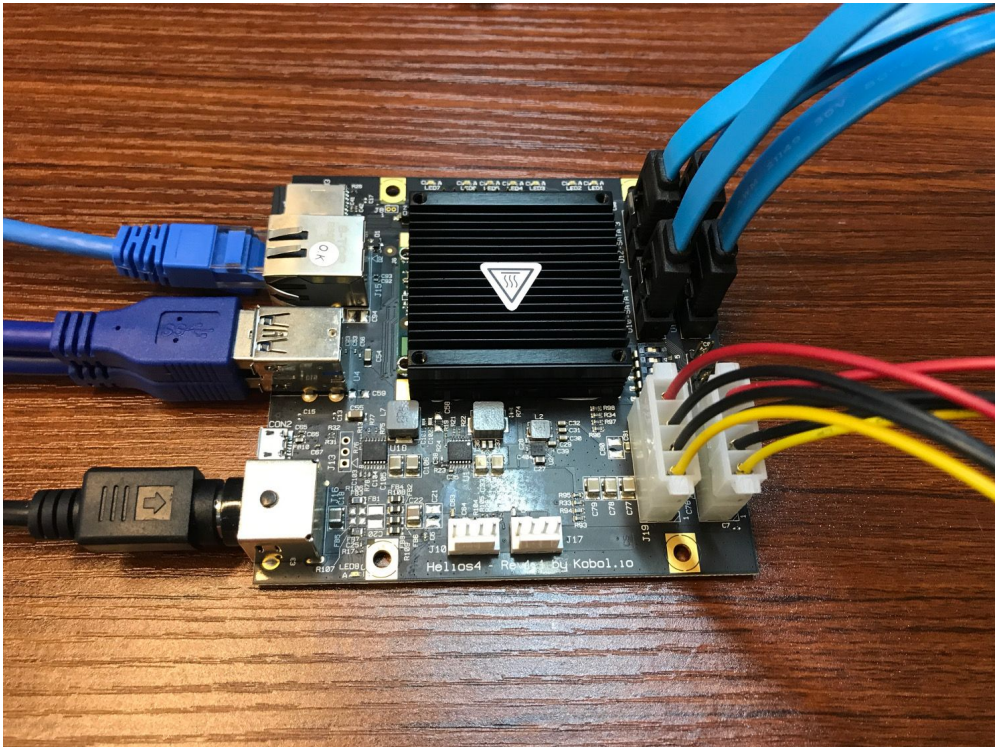


System Setup Diagram

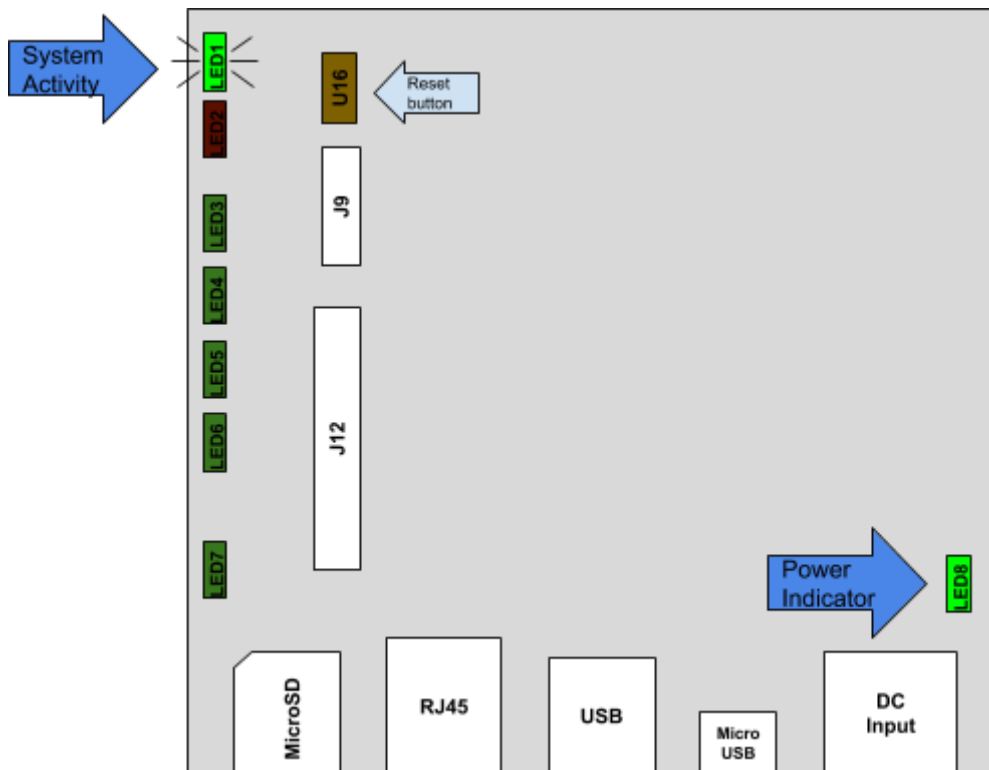
1. Insert MicroSD Card
2. Plug SATA Data, SATA Power, and USB cables
3. Plug Ethernet cable
4. Plug PSU cable (DC side)
5. Confirm all connection, peripheral does not touch each other
6. Plug PSU to AC source

Setup example :





Once the system is powered up, you will see the *LED8* lighting up. After a 15-30 seconds you should see the *LED1* blinking which indicates the Operating System is up and running.



Test Application

The Helios4 Test Suite provides an easy dialog menu where you can toggle for each interface background I/O operations that will exercise the interface.

The I/O operations that exercise the interfaces are generated by *fio* tool. The source code of the Helios4 Test Suite can be found [here](#).

To operate the Test Suite you will need to connect to Helios4 via SSH.

Network Config and Credentials

Helios4 OS has been configured with following configuration:

Network	IP Address	10.10.10.1
	Netmask	255.255.255.0
Credential	Username	root
	Password	root

Connect to Helios4

1. Set up Test PC network interface with following configuration

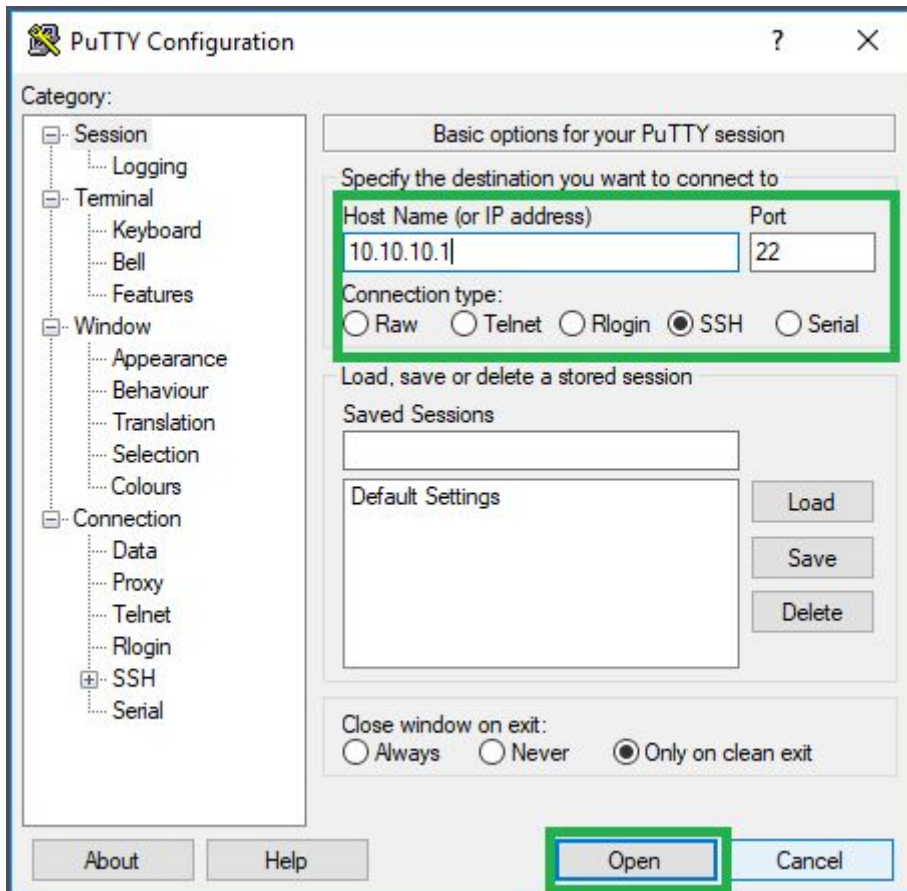
IP Address: 10.10.10.10 Netmask: 255.255.255.0

2. Connect to Helios4 via SSH.

- a. **Under Windows**

- a) [Download PuTTY](#) and install it on your computer.
- b) Run PuTTY from start menu
- c) Setup the connection and press Open

Host Name (or IP address)	10.10.10.1
Port	22
Connection type	SSH



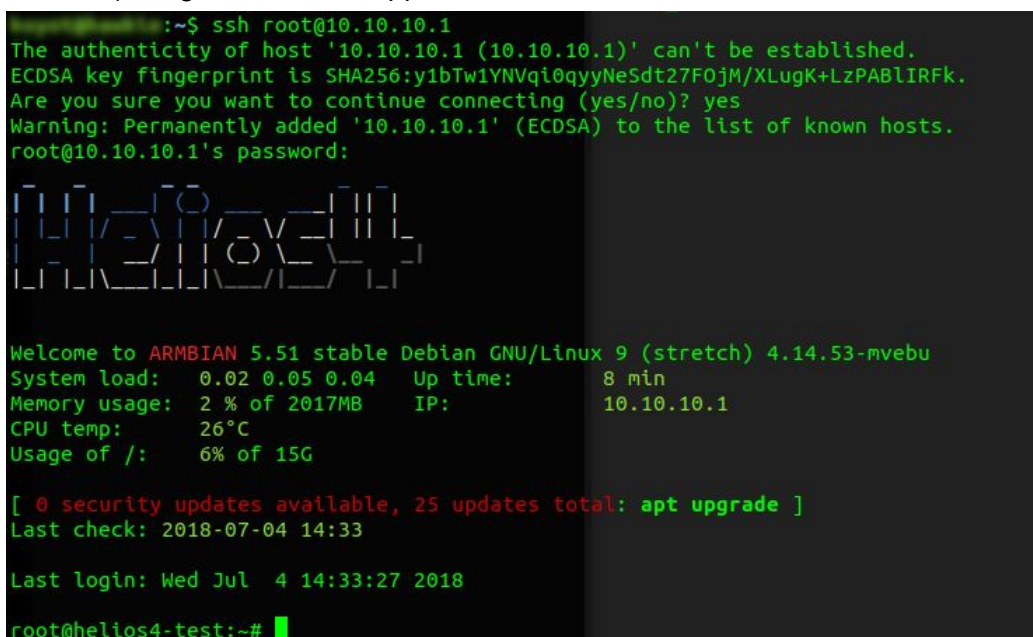
d) Login screen will appear, enter the credential.

b. Under Linux

a) Open Terminal and run

```
ssh root@10.10.10.1
```

b) Login screen will appear, enter the credential.

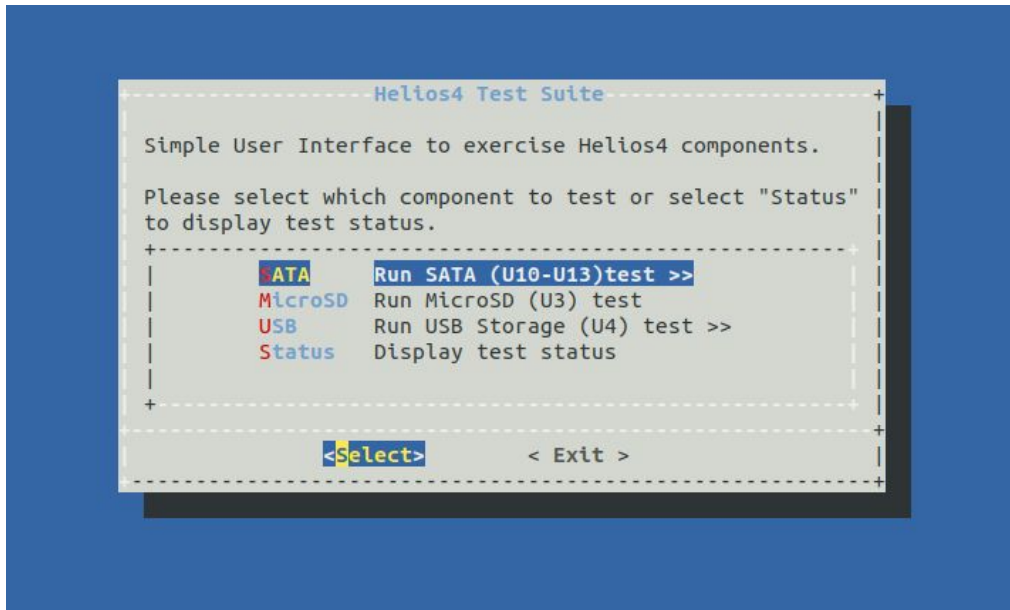


Test Suite Usage

Launch Application

To start Helios4 Test Suite use the following command:

```
root@helios4-test~# helios_test
```



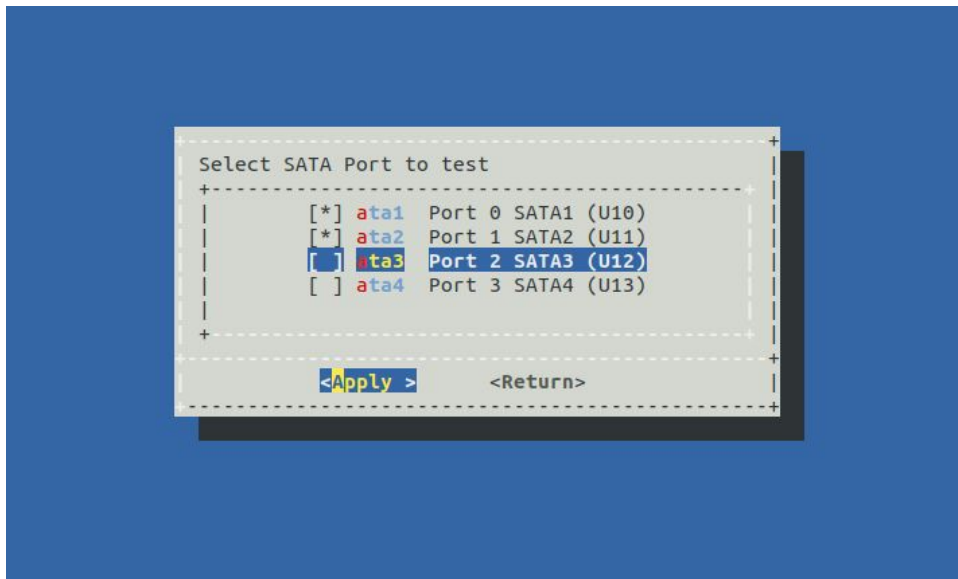
Run All tests

To automatically start all interface tests at once (SATA, MicroSD and USB), start the Test Suite with `--all` parameter:

```
root@helios4-test~# helios_test --all
```

Run SATA test

Select **SATA** submenu, then select which SATA port to test and press **Apply**.



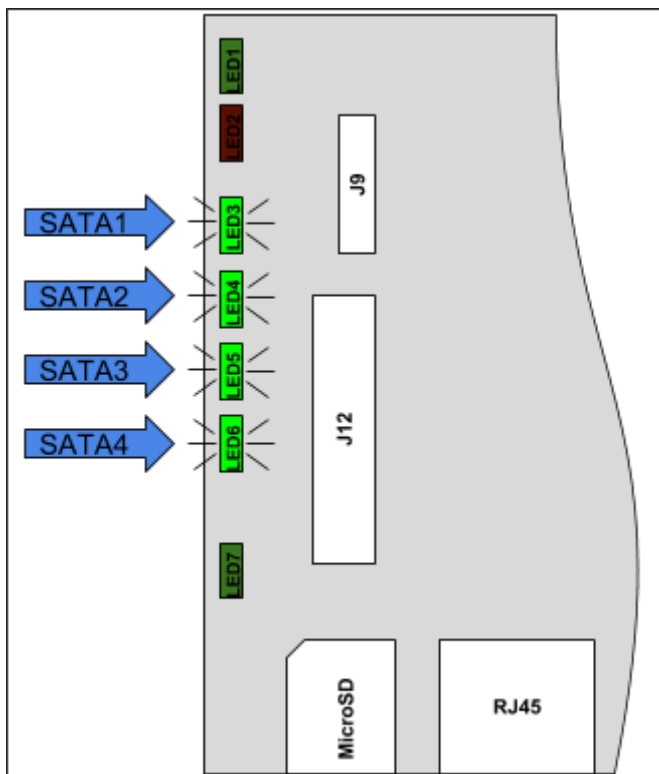
"ata1" to test SATA device connected to *U10-SATA1* header

"ata2" to test SATA device connected to *U11-SATA2* header

"ata3" to test SATA device connected to *U12-SATA3* header

"ata4" to test SATA device connected to *U13-SATA4* header

Note: Observe *LED3*, *LED4*, *LED5* and *LED6* activity that indicate respectively *SATA1*, *SATA2*, *SATA3* and *SATA4* activity. The LED will blink for the respective SATA port when the test is running.



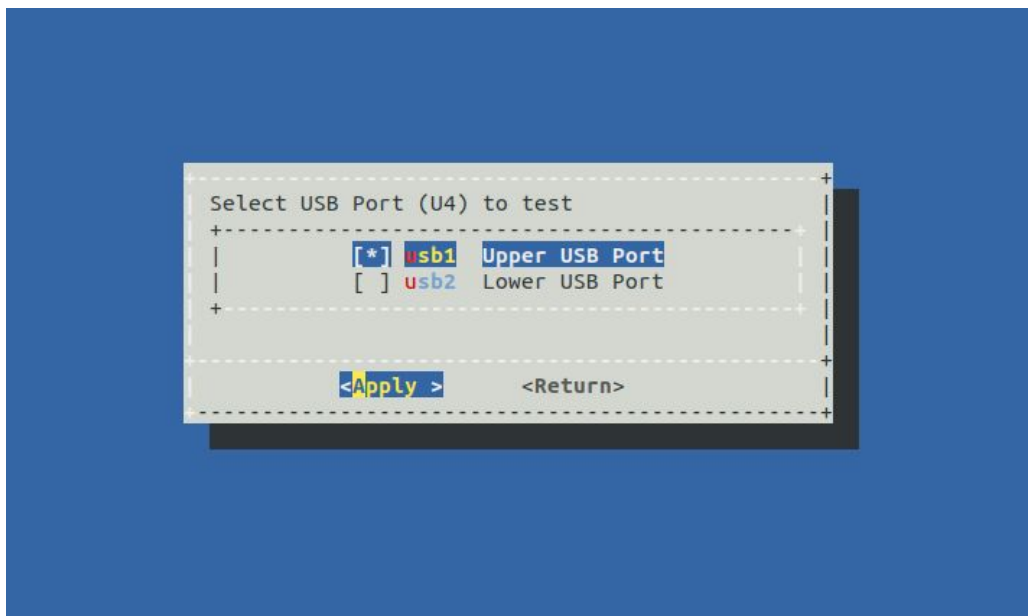
Run MicroSD test

Select **MicroSD** submenu. The test will start immediately.



Run USB test

Select **USB** submenu, then select which USB port to test and press **Apply**.

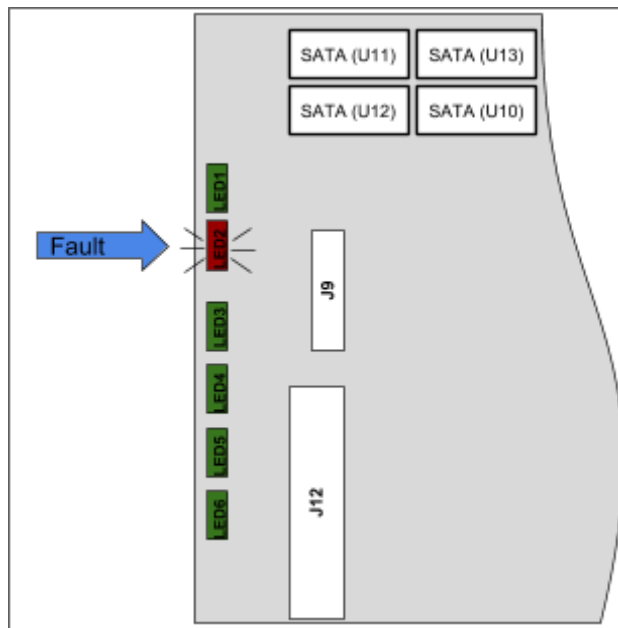


Troubleshooting

Troubleshooting table

Problem	Possible Cause	Solution
Cannot connect to Helios4	Faulty power supply	Verify whether LED8 is turned on. If it is not, tighten PSU cable. If problem still occurs, replace the PSU.
	OS failure	Verify whether LED1 is blinking. If it is not, reset or power cycle.
	Network down	Verify whether Ethernet LED turned on. If it is not, try to unplug and plug the network cable.
	Wrong network configuration	Configure Test PC network as instructed in Network Config and Credentials .
	Corrupt filesystem	Reflash the OS image as instructed in Writing an image to a microSD Card . If problem still occurs, replace the MicroSD card.
Cannot launch Helios4 Test application	Wrong software on microSD card	Check correct System Version is running . If wrong version, reflash the OS image as instructed in Writing an image to a microSD Card .
	Corrupt filesystem	Reflash the OS image as instructed in Writing an image to a microSD Card . If problem still occurs, replace the MicroSD card.

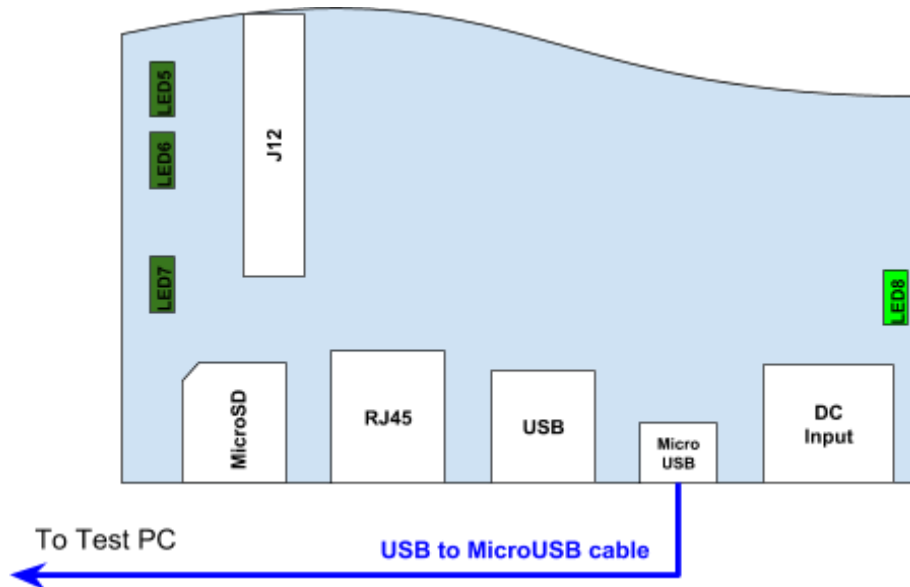
Cannot start test on one or more SATA Ports	SATA cable loose connection	Check whether all drives are detected . Power off the system, tighten the SATA cable and power on. If problem still occurs, replace the cable.
	Broken SATA drive	Check whether all drives are detected . Replace the drive.
Cannot start test on one or more USB Ports	USB extension cable	Check whether all drives are detected . Power off the system, tighten the USB cable and power on. If problem still occurs, replace the cable.
	Broken USB drive	Check whether all drives are detected . Replace the drive.
Helios4 system stops to respond	Network down	Verify whether Ethernet LED turned on. If it is not, try to unplug and plug the network cable.
	System crash/panic	Verify whether LED2 is blinking red. If yes, reset or power cycle.



Fault (red) LED, blinking when kernel panic occurred

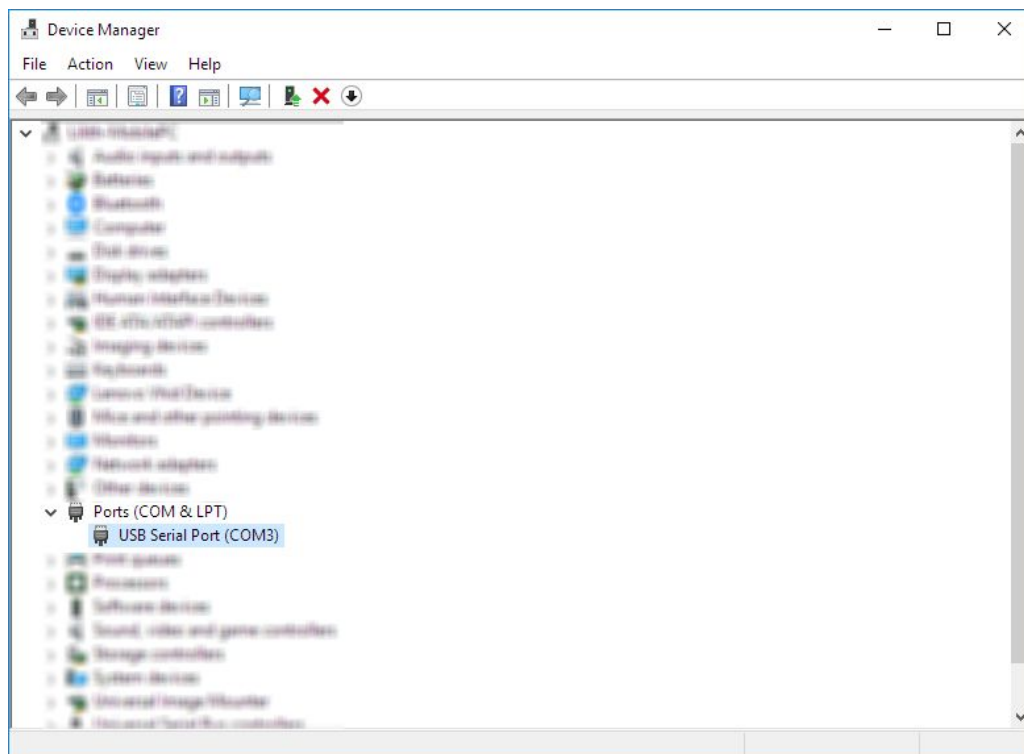
Serial Console Access

Plug in USB cable to Helios4 micro USB (CON2) connector and the other end to Test PC.

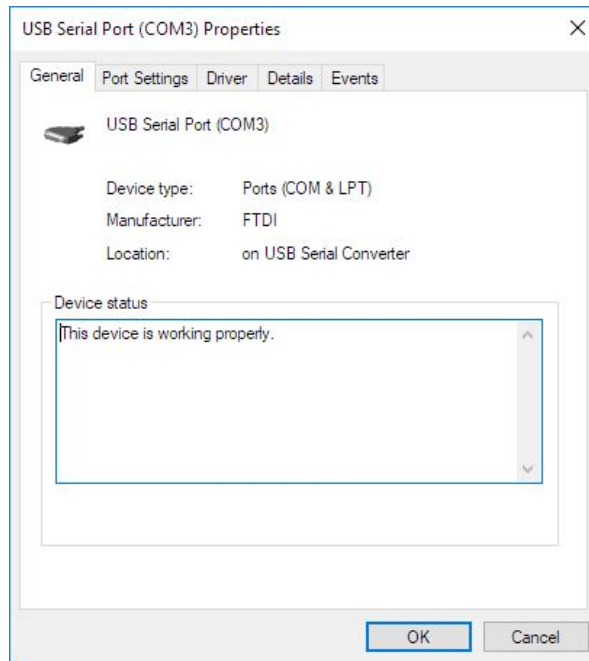


Under Windows

[Download Virtual COM Port \(VCP\) driver](#) and install. Open Device Manager, find *USB Serial Port* under “Ports (COM & LPT)”. Open *Properties* of the port.



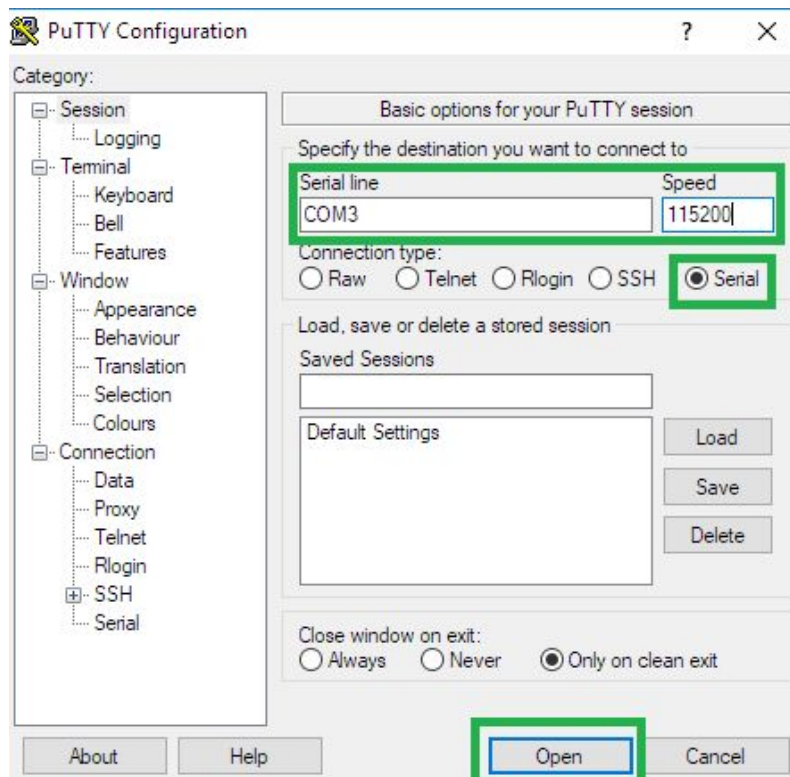
Verify that **FTDI** is the Manufacturer. Take note on the COM number.



Helios4 Serial Port assigned as COM3

Run PuTTY from Start Menu. and setup the connection and press Open

Serial line	Assigned COM number in Device Manager
Speed	115200
Connection type	Serial



Under Linux (via Terminal)

Install picocom (under Debian/Ubuntu)

```
root@helios4-test~# apt-get install picocom
```

Connect to serial (usually recognised as `/dev/ttyUSB0`) with picocom

```
root@helios4-test~# picocom -b 115200 /dev/ttyUSB0
picocom v1.7

port is          : /dev/ttyUSB0
flowcontrol     : none
baudrate is     : 115200
parity is       : none
databits are    : 8
escape is       : C-a
local echo is   : no
noinit is      : no
noreset is     : no
nolock is      : no
send_cmd is    : SZ -vv
receive_cmd is : RZ -vv
imap is        :
omap is        :
enap is        : crclrf,delbs,

Terminal ready

Debian GNU/Linux 8 helios4 ttyS0

helios4 login: █
```

Note: To exit picocom do **Ctrl-a** then **Ctrl-x**

Troubleshooting commands

Check all drives are detected

```
root@helios4-test~# lsblk
```

You should see the following result:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	1	111.8G	0	disk	
sdb	8:16	1	111.8G	0	disk	
sdc	8:32	1	111.8G	0	disk	
sdd	8:48	1	111.8G	0	disk	
sde	8:64	1	28.7G	0	disk	
sdf	8:80	1	28.7G	0	disk	
mmcblk0	179:0	0	14.9G	0	disk	
└─mmcblk0p1	179:1	0	14.7G	0	part	/
zram0	253:0	0	50M	0	disk	/var/log
zram1	253:1	0	504.4M	0	disk	[SWAP]
zram2	253:2	0	504.4M	0	disk	[SWAP]

4x SSD Drives

2x USB Drives

MicroSD Card

Check IP address is correct

```
root@helios4-test~# ifconfig eth0
```

You should see the following result:

```
root@helios4:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:43:25:fb:84
          inet addr:10.10.10.1  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::250:43ff:fe25:fb84/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:77 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:532
          RX bytes:15108 (14.7 KiB)  TX bytes:14196 (13.8 KiB)
          Interrupt:37

root@helios4:~#
```

Check correct System Version is running

- Check Kernel Build version

```
root@helios4-test~# uname -a
```

You should see the following result:

```
Linux helios4 4.14.53-mvebu #3 SMP Wed Jul 4 14:27:00 UTC 2018 armv7l GNU/Linux
```

- Check OS Build version

```
root@helios4-test~# cat /etc/armbian-release
```

You should see the following result:

```
# PLEASE DO NOT EDIT THIS FILE
BOARD=helios4
BOARD_NAME="Helios4"
BOARDFAMILY=mvebu
VERSION=5.51
LINUXFAMILY=mvebu
BRANCH=next
ARCH=arm
IMAGE_TYPE=stable
BOARD_TYPE=conf
INITRD_ARCH=arm
KERNEL_IMAGE_TYPE=zImage
```

- Check Helios4 Test Suite version

```
root@helios4-test~# helios4_test -v
```

You should see the following result:

```
Helios4 Test Suite - Version 0.2
```

Boot Screen

U-Boot

BootROM - 1.73

Booting from MMC

General initialization - Version: 1.0.0

AVS selection from EFUSE disabled (Skip reading EFUSE values)

Overriding default AVS value to: 0x23

Detected Device ID 6828

High speed PHY - Version: 2.0

Init Customer board SerDes lanes topology details:

Lane #	Speed	Type
0	3	SATA0
1	5	USB3 HOST0
2	3	SATA1
3	3	SATA3
4	3	SATA2
5	5	USB3 HOST1

High speed PHY - Ended Successfully

DDR3 Training Sequence - Ver TIP-1.46.0

DDR3 Training Sequence - Switching XBAR Window to FastPath Window

DDR Training Sequence - Start scrubbing

DDR Training Sequence - End scrubbing

DDR3 Training Sequence - Ended Successfully

BootROM: Image checksum verification PASSED



** LOADER **

U-Boot 2013.01 (Aug 28 2018 - 07:04:37) Marvell version: 2015_T1.0p16

Board: Helios4

SoC: MV88F6828 Rev A0
running 2 CPUs

CPU: ARM Cortex A9 MPCore (Rev 1) LE
CPU 0

CPU @ 1600 [MHz]

L2 @ 800 [MHz]

```

    TClock @ 250 [MHz]
    DDR3 @ 800 [MHz]
    DDR3 32 Bit Width, FastPath Memory Access, DLB Enabled, ECC Enabled
DRAM: 2 GiB
MMC: mv_sdh: 0
*** Warning - bad CRC, using default environment

USB2.0 0: Host Mode
USB3.0 0: Host Mode
USB3.0 1: Host Mode

Map: Code: 0x7fee6000:0x7ff978f0
     BSS: 0x7ffef2fc
     Stack: 0x7f9e3f20
     Heap: 0x7f9e4000:0x7fee6000
     U-Boot Environment: 0x000fe000:0x00100000 (MMC)

Board configuration detected:
Net:
| port | Interface | PHY address |
|-----|-----|-----|
| egiga0 | RGMII | 0x00 |
egiga0 [PRIME]
Hit any key to stop autoboot: 0
Trying to boot from MMC
1604 bytes read in 36 ms (43 KiB/s)
## Executing script at 03000000
Boot script loaded from mmc
105 bytes read in 29 ms (2.9 KiB/s)
19252 bytes read in 65 ms (289.1 KiB/s)
4687037 bytes read in 291 ms (15.4 MiB/s)
5605552 bytes read in 350 ms (15.3 MiB/s)
## Loading init Ramdisk from Legacy Image at 02880000 ...
   Image Name: uInitrd
   Created: 2018-08-22 12:27:29 UTC
   Image Type: ARM Linux RAMDisk Image (gzip compressed)
   Data Size: 4686973 Bytes = 4.5 MiB
   Load Address: 00000000
   Entry Point: 00000000
## Flattened Device Tree blob at 02040000
   Booting using the fdt blob at 0x02040000
   Using Device Tree in place at 02040000, end 02047b33

   Skipping Device Tree update ('fdt_skip_update' = yes)

Limit DDR size at 3GB due to power of 2 requirement of Address decoding

Starting kernel ...

```

Linux

```
Uncompressing Linux... done, booting the kernel.
Loading, please wait...
starting version 232
Begin: Loading essential drivers ... done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ...
done.
Begin: Running /scripts/local-premount ... Scanning for Btrfs filesystems
done.
Begin: Will now check root file system ... fsck from util-linux 2.29.2
[/sbin/fsck.ext4 (1) -- /dev/mmcblk0p1] fsck.ext4 -a -C0 /dev/mmcblk0p1
/dev/mmcblk0p1: clean, 36372/887168 files, 283558/3779284 blocks
done.
done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.

Welcome to Debian GNU/Linux 9 (stretch)!

[ OK ] Created slice User and Session Slice.
[ OK ] Reached target Swap.
[ OK ] Reached target Remote File Systems.
[ OK ] Created slice System Slice.
        Mounting POSIX Message Queue File System...
[ OK ] Created slice system-serial\x2dgetty.slice.
[ OK ] Created slice system-getty.slice.
[ OK ] Started Forward Password Requests to Wall Directory Watch.
[ OK ] Listening on /dev/initctl Compatibility Named Pipe.
[ OK ] Reached target Slices.
[ OK ] Started Dispatch Password Requests to Console Directory Watch.
[ OK ] Reached target Paths.
[ OK ] Reached target Encrypted Volumes.
[ OK ] Listening on fsck to fsckd communication Socket.
[ OK ] Listening on Journal Audit Socket.
[ OK ] Set up automount Arbitrary Executab...rmats File System Automount
Point.
[ OK ] Listening on udev Control Socket.
[ OK ] Listening on Syslog Socket.
[ OK ] Listening on Journal Socket (/dev/log).
        Mounting Debug File System...
[ OK ] Listening on udev Kernel Socket.
[ OK ] Listening on Journal Socket.
        Starting Set the console keyboard layout...
        Starting Journal Service...
        Starting Create list of required st...ce nodes for the current
kernel...
        Starting Load Kernel Modules...
        Starting Restore / save the current clock...
        Starting Remount Root and Kernel File Systems...
        Starting Nameserver information manager...
[ OK ] Mounted Debug File System.
[ OK ] Mounted POSIX Message Queue File System.
[ OK ] Started Create list of required sta...vice nodes for the current
kernel.
```

```

[ OK ] Started Load Kernel Modules.
[ OK ] Started Remount Root and Kernel File Systems.
      Starting Load/Save Random Seed...
      Starting udev Coldplug all Devices...
      Starting Apply Kernel Variables...
      Mounting FUSE Control File System...
      Starting Create Static Device Nodes in /dev...
[ OK ] Mounted FUSE Control File System.
[ OK ] Started Journal Service.
[ OK ] Started Set the console keyboard layout.
[ OK ] Started Restore / save the current clock.
[ OK ] Started Nameserver information manager.
[ OK ] Started Load/Save Random Seed.
[ OK ] Started Apply Kernel Variables.
      Starting Flush Journal to Persistent Storage...
[ OK ] Started Flush Journal to Persistent Storage.
[ OK ] Started Create Static Device Nodes in /dev.
[ OK ] Reached target Local File Systems (Pre).
      Mounting /tmp...
      Starting udev Kernel Device Manager...
[ OK ] Mounted /tmp.
[ OK ] Started udev Coldplug all Devices.
[ OK ] Reached target Local File Systems.
      Starting Set console font and keymap...
      Starting Armbian ZRAM config...
      Starting Create Volatile Files and Directories...
      Starting Raise network interfaces...
[ OK ] Started Set console font and keymap.
[ OK ] Started Create Volatile Files and Directories.
[ OK ] Reached target System Time Synchronized.
      Starting Update UTMP about System Boot/Shutdown...
[ OK ] Started Entropy daemon using the HAVEGE algorithm.
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ OK ] Started udev Kernel Device Manager.
[ OK ] Found device /dev/ttyS0.
[ OK ] Started Armbian ZRAM config.
      Starting Armbian memory supported logging...
[ OK ] Started Armbian memory supported logging.
[ OK ] Reached target System Initialization.
      Starting Armbian hardware monitoring...
[ OK ] Started Daily Cleanup of Temporary Directories.
[ OK ] Started Daily apt download activities.
[ OK ] Started Daily apt upgrade and clean activities.
[ OK ] Reached target Timers.
      Starting Armbian hardware optimization...
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Started Armbian hardware optimization.
[ OK ] Started Raise network interfaces.
[ OK ] Started Armbian hardware monitoring.
[ OK ] Reached target Basic System.
      Starting Login Service...
      Starting System Logging Service...
      Starting LSB: Load kernel modules needed to enable cpufreq
scaling...
[ OK ] Started Regular background program processing daemon.
[ OK ] Started D-Bus System Message Bus.

```

```
Starting Network Manager...
Starting LSB: Start/stop sysstat's sadc...
[ OK ] Started System Logging Service.
[ OK ] Started LSB: Start/stop sysstat's sadc.
[ OK ] Started Login Service.
[ OK ] Started LSB: Load kernel modules needed to enable cpufreq scaling.
Starting LSB: set CPUFreq kernel parameters...
[ OK ] Started LSB: set CPUFreq kernel parameters.
Starting LSB: Set sysfs variables from /etc/sysfs.conf...
[ OK ] Started LSB: Set sysfs variables from /etc/sysfs.conf.
[ OK ] Started Network Manager.
[ OK ] Reached target Network.
Starting Permit User Sessions...
Starting OpenBSD Secure Shell server...
[ OK ] Started Unattended Upgrades Shutdown.
Starting Network Manager Wait Online...
[ OK ] Started Permit User Sessions.
Starting Network Manager Script Dispatcher Service...
[ OK ] Started Network Manager Script Dispatcher Service.
Starting Hostname Service...
[ OK ] Started Hostname Service.
[ OK ] Started OpenBSD Secure Shell server.
Starting Authorization Manager...
[ OK ] Started Network Manager Wait Online.
[ OK ] Reached target Network is Online.
Starting LSB: Advanced IEEE 802.11 management daemon...
Starting /etc/rc.local Compatibility...
Starting LSB: Start NTP daemon...
[ OK ] Started LSB: Advanced IEEE 802.11 management daemon.
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Serial Getty on ttyS0.
[ OK ] Started Getty on tty1.
[ OK ] Reached target Login Prompts.
[ OK ] Started Authorization Manager.
[ OK ] Started LSB: Start NTP daemon.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.
```

Debian GNU/Linux 9 helios4-test ttyS0

helios4-test login: