

# ***Hauppauge MediaMVP-HD Developer FAQ***

***Version 1.2***

***Last Updated: January 10, 2011***

## ***What is the Hauppauge MediaMVP-HD?***

The Hauppauge MediaMVP-HD is a Linux based embedded system running on a Sigma 8655 Media processor. It contains 512MB of RAM, 128MB of flash, and is optimized for HD video playback.

## ***What video formats does it support?***

HD playback of pretty much everything under the sun: MPEG, MPEG-2, H.264, MKV, VC1, etc.

## ***What sources of video can it play?***

It can access video via USB-connected storage devices, or over UPnP or CIFS (e.g. Samba or Windows file shares).

## ***How is it on power consumption?***

It consumes 5 watts when doing HD playback at 1080p.

## ***So this thing runs Linux? Where can I get the source code and what can I do?***

Like most embedded Linux products, the MediaMVP-HD is a hybrid containing both open source and closed source components. The Linux kernel is open source, but it has three proprietary kernel modules (one for the NAND flash filesystem, and two for the Media playback framework). All three of these are provided by Sigma and hence it's outside of Hauppauge's control to provide the source code.

The core user interface as well is based on closed source code provided by Sigma, and hence it cannot be open sourced.

All of the open source components used in the product can be downloaded from:

<ftp://www.hauppauge.com/Support/Media%20MVP-HD/GPL%20Compliance/>

## ***But does this mean that for all practical purposes I am locked out (like the Tivo)?***

No. Hauppauge has provided a variety of mechanisms by which the product can be extended by hobbyists and commercial parties looking to do customization. For example, we have provided all the playback mechanisms as command line tools, which custom written applications (or scripts) can call to play video or audio files.

We've also avoided employing technologies such as code signing which restrict what users can run on the platform.

In the interest of fostering a community behind the device, we look forward to making available the toolchain needed to build applications against the target (it's a gcc based MIPS crosscompile toolchain), as well as providing some examples of how to write your own applications.

We're looking to do more as the product continues to mature.

***I see that the device has a telnet server? Can I use that to get a shell?***

Yes. The root password is "hauppaugetw"

***I took the box apart and saw that it has a five pin header. Is that a serial port?***

Yes. However, you need a TTL level converter to connect to it. The pinout is as follows:

Pin 1 - VCC 5V

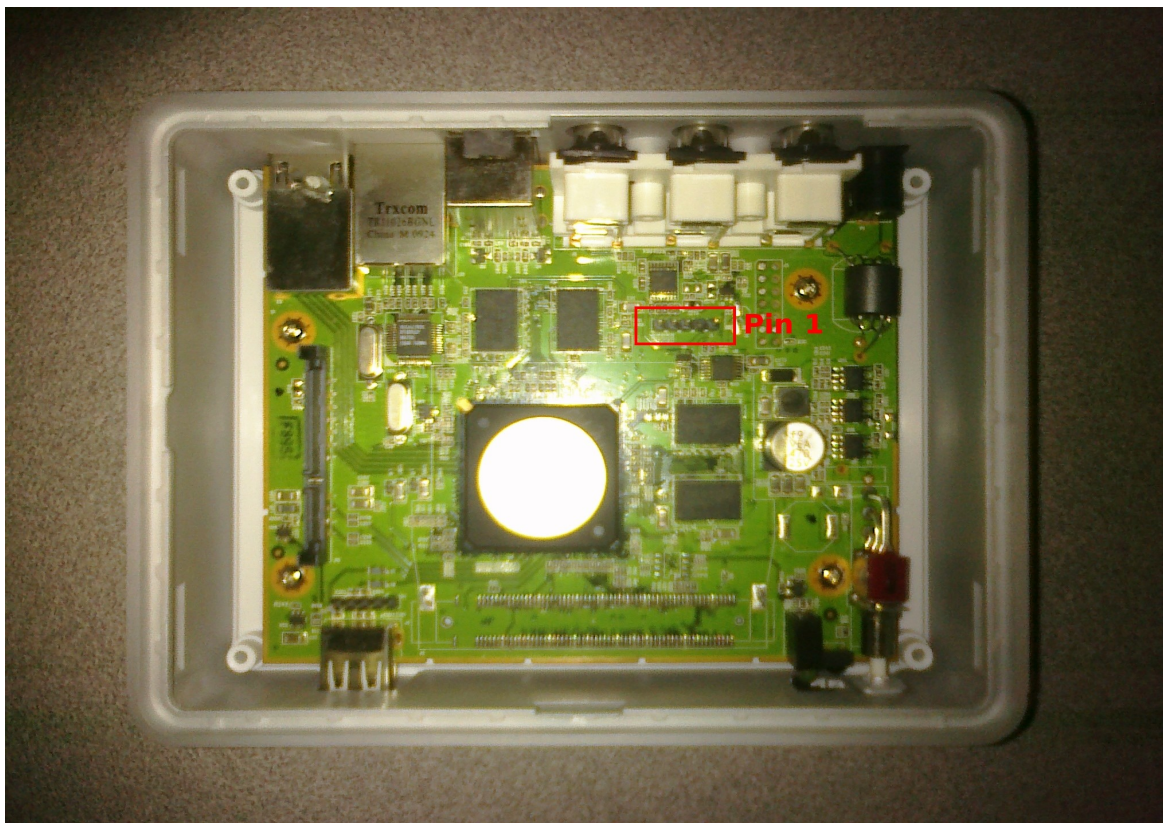
Pin 2 - key (to ensure the connector is not put on backwards)

Pin 3- TX

Pin 4 - RX

Pin 5 - Ground

Warning: If you don't know what a "level converter" is, then you probably shouldn't be playing with the serial port (stick with using the Telnet service). Connecting a serial cable without a level converter will damage the board.





***What I can do with the serial port that I cannot do with the Telnet server?***

Both of them provide root access, however the serial port also allows you to see things like the kernel output at startup, and to interact with the hardware even when the Telnet service is inaccessible (such as during bootup or if there is a networking problem).

***The WD Live is also a Sigma based product. Can I use the same serial cable that some hobbyists have suggested for that device?***

Yes, but the pinout on the MediaMVP-HD is slightly different, you you will need to rewire the connector.

***What are the other connectors inside the unit?***

In addition to the UART, the device also has a 5 pin internal USB connector and a SATA interface. Note that the SATA interface provides sufficient power to drive a 2.5" internal hard drive, but connecting a 3.5" drive requires an external power source.

***Can I install programs I wrote onto the device's internal flash?***

Yes. Starting in release 1.1.19, there are now hooks that allow a developer to place applications into the device flash (and have those applications automatically started at boot). See the question "*How do I provide my own applications*" below.

***So how do I compile my own programs for the MediaMVP-HD?***

Here's a quick tutorial on building a "hello world" binary that runs on the platform. This tutorial assumes the user is running a stock Ubuntu 9.10 environment (either x86 or x86-64 will work). This can either be a standalone installation of Linux, or installed inside of a virtual machine such as VMWare Workstation or Oracle VirtualBox.

First you need to get a cross compiler (this is a compiler that is designed to build for a target other than the host). The easiest way to do this is to download the freely available GCC toolchain put together by CodeSourcery:

<http://www.codesourcery.com/sgpp/lite/mips>

It's worth noting that this is their "Lite" edition and they do have a commercial product which includes more features, although the Lite edition is enough to build working executables. Alternatively, you can build the MIPS version of GCC yourself from source, although this tends to not be a trivial exercise and there is not a significant value in doing such for most applications.

Once the compiler is installed, you need to put the toolchain binaries into your path:

```
cd mips-4.4
export PATH=`pwd`/mips-4.4/bin:$PATH
```

Now let's create a simple C program to demonstrate the compiler is working. Using a text editor, create the following file called "hello.c":

```
#include <stdio.h>

int main(void)
{
    printf("Hello World\n");
    return 0;
}
```

Now let's compile the executable:

```
mips-linux-gnu-gcc --static -muclibc -EL -Wall -o hello hello.c
```

You should end up with an executable named "hello". You can copy that file to a USB stick, insert it into the MediaMVP-HD, and run the program from a telnet session:

```
devin@devin-macbook2:~/mtest$ telnet 172.16.1.30
Trying 172.16.1.30...
Connected to 172.16.1.30.
Escape character is '^]'.

MediaMVP-HD login: root
Password:
MediaMVP-HD[~]$ /disk/USB/disk1/hello
Hello World
MediaMVP-HD[~]$
```

How do I play video to the screen?

The easiest way to do this is to make use of the executables Hauppauge provides on the system. Before you can render video though, you need to stop the main Hauppauge application:

```
MediaMVP-HD[~]$ killall curacao
```

At this point, you can run the "test\_rmfp" tool to render video:

```
/usr/local/bin/test_rmfp /disk/USB/disk1/myvideo.mpg
```

You can run "test\_rmfp -help" to see all the possible command line options.

To restart the Hauppauge GUI, either reboot the box or run the following from the command line:

```
/etc/init.d/S60curacao start
```

There is also a tool called "/bin/set\_outports" which allows you to change the screen resolution (you

can run the following to see all the command line options:

```
/bin/set_outports -help
```

Note that the Hauppauge application automatically sets the screen resolution at startup (and leaves it in that state even if you shutdown the application. Hence, it may be easier for users to just configure the screen resolution the first time using the MediaMVP-HD, and then just avoid using set\_outports entirely.

### ***How do I play video and make it stop if the user hits the "OK" button on the remote?***

Here's a script that can demonstrate some basic playback with response to the remote:

```
#!/bin/sh

if [ -z "$1" ]; then
    echo "Usage: $0 <filename>"
    exit 1
fi
if [ ! -f "$1" ]; then
    echo "Error: file not found"
    exit 1
fi

/usr/local/bin/test_rmfp $1 >/dev/null 2>&1 &
echo "Hit OK on your remote to terminate video"

# Run in a loop until user hits "ok"
while [ 1 ]; do
    cat /dev/ir0 |hexdump -d -n 1 > /tmp/iroutput
    killall cat 2>/dev/null
    grep 00165 /tmp/iroutput > /dev/null
    if [ $? == 0 ]; then
        echo "OK found"
        killall test_rmfp
        exit 0
    fi
    sleep 1
done
```

### ***How do I prevent the main Hauppauge application from running?***

To prevent the main Hauppauge GUI (known as Curacao) from being executed at device boot, run the following commands:

```
/etc/SETUP_set LAUNCH_CURACAO no
```

```
/upgrade/setup_save.sh
```

To re-enable the GUI, run the following:

```
/etc/SETUP_set LAUNCH_CURACAO yes  
/upgrade/setup_save.sh
```

Note: Curacao is responsible for configuring the various video outputs (defining which outputs are enabled, the video standard, and resolution). If an OEM disables the launching of Curacao, they will need to perform the equivalent functionality by running the “set\_outports” command. The full options can be seen for set\_outports by running:

```
set_outports -help
```

### ***How do I provide my own applications?***

Starting in firmware version 1.1.19, developers can now install their own applications to be run either alongside the main Hauppauge provided user interface, or replace it entirely.

To install the sample application (which shows some system information on the screen asks the user to hit a button on the remote control, and then plays a video named “sample.ts” off of a USB stick), run the following via telnet or the serial console.

```
mkdir /vendor  
mount /dev/sigmblocki /vendor  
cp /usr/share/vendor/example_vendor.sh /vendor/runme.sh  
chmod 755 /vendor/runme.sh  
/etc/SETUP_set LAUNCH_VENDORAPP yes  
/upgrade/setup_save.sh  
umount /vendor
```

Once the unit gets power cycled, the /vendor/runme.sh script will be executed before Curacao. Please see the file /usr/share/vendor/example\_vendor.sh for more development notes on how it can be modified/customized to meet the customer needs.

To disable the execution of the vendor application at boot, run:

```
/etc/SETUP_set LAUNCH_VENDORAPP no  
/upgrade/setup_save.sh
```

Other files required for the application can be copied to /vendor as well (e.g. code, images, MP3 files, etc), up to the 64MB limit of the partition.

Note: any changes/customization the OEM may wish to perform should be made to /vendor/runme.sh, and not to /usr/share/vendor/example\_vendor.sh (which is just a read-only example script).

### ***How do I perform a factory reset of the unit from the command line?***

To reset the unit to its factory state, run the following;

```
/upgrade/restore_default.sh
```

Note: this command will revert OEM customization performed on the device, including custom applications and whether Curacao is configured to launch. Factory reset will not have an effect on the boot splash screen though.

### ***How do I perform a factory reset from the remote control?***

Starting in firmware 1.1.20, the device can be factory reset from the remote control, even if the device does not boot to the main user interface. This functionality is intended for OEM developers who render their unit inoperable through a custom application, or cases where a configuration error results in the user unable to see the main GUI.

To perform a factory reset at boot:

Point the Hauppauge remote control at the device and hold down the YELLOW key while powering up. Keep holding it down through the splash screen until the splash screen disappears and then reappears. Once the user sees the splash screen for the second time, release the yellow button. The unit has been factory reset.